University of Reading

Gender and computer programming:

Teaching and learning strategies designed to increase the engagement of girls

---

Thesis submitted for the Degree of Educational Doctorate

By

Terence McAdams

Institute of Education

Oct 2018

Author's Declaration

'I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.'

A proof-reader has not assisted me with this thesis!

Signed:

____

Terence McAdams, Oct 2018

Abstract

The purpose of this research was to examine why so many girls decided to stop studying computer programming when they transition from middle school to senior school. This thesis examined ability and gender attitudes towards computer programming in middle school students at an International school in South Korea. In this study, 194 students in Year 8 and Year 9 in single-sex classes were taught Python and HTML5/CSS using a variety of teaching and learning strategies including tutorials, problem-based learning, tasks that included visual design, game-based learning, and storytelling. At the year-end, participants were given a computer programming assessment, with girls, relative to boys, demonstrating significantly greater computer programming ability. There was no difference between genders in the most-able programmers.

Student opinions were gathered from questionnaires and group interviews. Findings showed that there was a gender difference in preferred learning strategies, with girls enjoying computer programming incorporating visual design, storytelling, and problem-based projects more than the boys. Further, there was no significant gender difference in enjoyment, confidence, or anxiety after a year of programming using the various teaching and learning strategies.

Boys and girls did not differ in their reasons for choosing to study a subject from the following list (parents' opinions; friends' opinions; teachers' opinions; useful life skills; lesson enjoyment; career/university skills; role models). The biggest influencing factor for both genders was lesson enjoyment and the opinion of friends was the least influential factor.

The findings indicated that if computer programming is taught using the preferred teaching and learning strategies more girls are likely to choose to continue studying computer programming. In this study, the number of Year 9 girls choosing to continue studying computer programming increased from 5 girls in the first year (13% of the total) to 17 girls (38% of the total).

Acknowledgements

Sincere appreciation goes to the teaching staff at the University of Reading, Institute of Education; particularly my thesis supervisors, Dr. Natthapoj Vincent Trakulphadetkrai and Dr. Daisy Powell, and the programme director, Dr. Carol Fuller. Their professionalism, dedication and knowledge have made this process a thoroughly enjoyable experience on so many different levels.

Having never expected to continue in education beyond the age of 16, I would like to thank my friend and former colleague, Denise, for persuading me to enrol on this programme. I have enjoyed studying education with you immensely.

This research would not have been possible with the support of the students and staff at the school where I conducted this research.

Lastly, and most importantly, I wish to acknowledge the support of my wife, Chryssy, who has demonstrated unshakable faith in this undertaking. She has put up with me sitting at the computer for innumerable hours, while only half-listening to her.

Table of Contents

Author's Declaration 2

Abstract 3

Acknowledgements 4

Chapter 1 Introduction

1.1 Introduction 13

1.2 History of computing programming in English-curriculum schools 17

1.3 Context 20

1.4 Structure of the thesis 22

Chapter 2 Literature Review

2.1 Introduction 24

2.2 Role of computer programming in modern society 24

2.3 The effects of gender, social class and ethnicity on students' career aspirations 26

2.4 Nature of computer programming 28

  2.4.1 Suitable programming languages for novices 29

2.4 Intrinsic factors that may influence the number of female computer programmers 34

  2.4.1 Sex and gender differences in computer programming 35

  2.4.2 Gender differences in engagement in computer programming 39

  2.4.3 Gender differences in computer programming confidence 41

  2.4.4 Perception of computer scientists 43

2.5 Extrinsic factors that may influence the number of female computer programmers 45

  2.5.1 A lack of female role models 45

  2.5.2 Intimidating environment for females 48

  2.5.3 Influence of single sex classes in choosing to study Computer Science 51

2.6 Strategies for learning to program 52

  2.6.1 Self-instructional tutorials 53

  2.6.2 Problem-based learning (PBL) 55

  2.6.3 Storytelling 57

  2.6.4 Visual design 58

  2.6.5 Game-based learning 59

  2.6.6 Diagrammatic learning 61

2.6.7 Collaborative learning and cooperative learning 64
2.7 Summary and research questions 65

Chapter 3 Methodology and Methods

3.1 Introduction 67
3.2 Research Approach 67
3.2.1 Ontology and epistemology 68
3.2.2 Theoretical perspective 70
3.2.3 Methodology 71
3.2.4 Mixed Methods 72
3.2.5 Concerns with gender research 74
3.3 Participants 75
3.3.1 Participants prior experience of computer programming 76
3.3.2 Sampling 77
3.3.3 Timeline for delivery of the teaching strategies and data collection 79
3.3.4 Curriculum content 81
3.3.5 Teaching and learning pedagogies 82
3.3.5.1 Strategy 1 – Self-instructional tutorials 83
3.3.5.2 Strategy 2 - Storytelling 84
3.3.5.3 Strategy 3 – Visual design 85
3.4 Data collection 86
3.4.1 Data collection - Research Question 1 ('Is there a significant gender difference in computer programming ability?') 87
3.4.1.1 HTML/CSS assessment 88
3.4.1.2 Python assessment 88
3.4.1.3 Marking the programming assessment 89
3.4.2 Data collection - Research Question 2 ('Are there any intrinsic factors that affect girls choosing to study computer programming in Year 10?') 90
3.4.2.1 Computer Programming Anxiety and Confidence (CPCA) Questionnaire 91
3.4.2.2 Student Perceptions of Computer Science (SPCS) Questionnaire 92
3.4.3 Data collection - Research Question 3 ('Are there any extrinsic factors that affect girls choosing to study computer programming in Year 10?') 94
3.4.3.1 Influences on IGCSE Choices (IIC) Questionnaire 94

3.4.4 Data collection - Research Question 4 ('Is there a significant gender difference in the preferred learning styles') 96
3.4.4.1 Preferred Computer Programming Learning Style (PCPLS) Questionnaire 96
3.4.4.2 Group interviews 97
3.5 Data analysis 98
3.5.1 Data analysis - Research Question 1 ('Is there a significant gender difference in computer programming ability?') 98
3.5.2 Data analysis - Research Question 2 ('Are there any intrinsic factors that affect girls choosing to study computer programming in Year 10?') 100
3.5.3 Data analysis - Research Question 3 ('Are there any extrinsic factors that affect girls choosing to study computer programming in Year 10?') 103
3.5.3.1 Qualitative data 103
3.5.4 Data analysis - Research Question 4 ('Is there a significant gender difference in the preferred learning styles') 105
3.6 Ethical considerations 105
3.7 Summary 107

Chapter 4 Quantitative Findings

4.1 Introduction 108
4.2 Quantitative Results - Research Question 1 ('Is there a significant gender difference in computer programming ability?') 108
4.2.1 Gender difference in overall programming ability 109
4.2.2 Gender differences in programming ability in the most able programmers 110
4.2.3 Gender differences in programming ability in the students that successfully complete the extension task, 'The Rainfall Problem' 111
4.3 Quantitative Results - Research Question 2 ('Are there any intrinsic factors that affect girls choosing to study computer programming in Year 10?') 111
4.3.1 Computer programming confidence and anxiety 112
4.3.1.1 One-way ANOVA on programming confidence and anxiety 113
4.3.1.2 Confidence to complete tasks and study computer science at university 115
4.3.2 Computer programming enjoyment 116
4.3.2.1 Computer gaming enjoyment 118
4.3.3 Computer programming difficulty 120
4.3.4 Future plans 121

4.3.4 Perceptions of computer programming..121
4.4 Quantitative Results – Research Question 3 ('Are there any extrinsic factors that affect girls choosing to study computer programming in Year 10?')..123
4.4.1 Results from Influences on IGCSE Choices (IIC) Questionnaire 123
4.5 Quantitative Results – Research Question 4 ('Is there a significant gender difference in the preferred learning styles for computer programming')..125
4.51 Results from the Preferred Computer Programming Learning Style (PCPLS) Questionnaire 125
4.6 Summary 128

Chapter 5 Qualitative Findings

5.1 Introduction..129
5.2 Qualitative Results - Research Question 1 ('Is there a significant gender difference in computer programming ability?') 131
5.3 Qualitative Results - Research Question 2 ('Are there any intrinsic that affect girls choosing to study computer programming in Year 10?') 132
5.3.1 Computer programming confidence and anxiety...133
5.3.2 Computer programming enjoyment...133
5.3.3 Computer programming difficulty..135
5.3.4 Future plans...135
5.3.5 Perceptions of computer programming..136
5.4 Qualitative Results - Research Question 3 ('Are there any extrinsic factors that affect girls choosing to study computer programming in Year 10?') 138
5.4.1 Influence of teachers 138
5.4.2 Influence of parents 139
5.4.3 Influence of role models 140
5.4.4 Influence of friends 141
5.4.5 Influence of future careers 141
5.4.6 Other influencing factors 141
5.5 Qualitative Results – Research Question 4 ('Is there a significant gender difference in the preferred learning styles for computer programming?')..142
5.5.1 Self-instructional tutorials...144
5.5.2 Problem-based learning / Project-based learning 144
5.5.3 Computer programming with visual design..145

5.5.4 Working independently or in groups 145
5.5.5 Computer programming with storytelling 146
5.5.6 Game-based learning 146
5.5.7 Combination of learning strategies 147
5.6 Summary 147

Chapter 6 Discussion

6.1 Introduction 148
6.2 Discussion - Research Question 1 (**‘**Is there a significant gender difference in computer programming ability?’**)** 148
6.2.1 Potential confounding factors that might influence the test results 150
6.2.1.1 Pre-existing programming knowledge 150
6.2.1.2 External support through Academies and Tutors 151
6.2.1.3 More than one teacher delivering the curriculum 151
6.2.1.4 Deliberately trying to increase the number of Computer Science girls 152
6.2.1.5 Class size 152
6.3 Discussion - Research Question 2 (‘Are there any intrinsic that affect girls choosing to study computer programming in Year 10?’) 152
6.3.1 Computer programming confidence and anxiety 152
6.3.2 Computer programming enjoyment 154
6.3.2.1 Computer gaming enjoyment 155
6.3.3 Computer programming difficulty 155
6.3.4 Future plans 156
6.3.5 Perceptions of computer programmers 156
6.4 Discussion - Research Question 3 (‘Are there any extrinsic factors that affect girls choosing to study computer programming in Year 10?’) 157
6.4.1 Influence of teachers 158
6.4.2 Influence of parents 158
6.4.3 Influence of role models 159
6.4.4 Influence of friends 160
6.5 Discussion – Research Question 4 (**‘**Is there a significant gender difference in the preferred learning styles for computer programming?’**)** 160
6.5.1 Self-instructional tutorials 161
6.5.2 Problem-based learning / Project-based learning 161

6.5.3 Computer programming with visual design....162
6.5.4 Working independently or in groups 162
6.5.5 Computer programming with storytelling 162
6.5.6 Game-based learning 163
6.5.7 Combination of learning strategies....163

Chapter 7 Conclusions

7.1 Introduction....164
7.2 Key Findings....164
7.2.1 Key findings concerning Research Question 1 ('Is there a significant gender difference in computer programming ability?')....164
7.2.2 Key findings concerning Research Question 2 ('Are there any intrinsic factors that affect girls choosing to study computer programming in Year 10?')....165
7.2.3 Key findings concerning Research Question 3 ('Are there any extrinsic factors that affect girls choosing to study computer programming in Year 10?')....166
7.2.4 Key findings concerning Research Question 4 ('Is there a significant gender difference in the preferred learning styles for computer programming?') 166
7.3 Potential limitations of the current research 167
7.4 Implications for schools....168
7.5 Implications for further research....169
7.6 Summary....170

References....171

Appendices

Appendix A: Programming task (Microsoft Visual Basic 2010)....235
Appendix B: Ethics documentation 237
Appendix C: Tutorial for learning JavaScript programming through short tasks 266
Appendix D: Tutorial for learning Python through short tasks....268
Appendix E: Task for learning computer programming through flowcharts 270
Appendix F: Tutorial for learning Python programming through storytelling....272
Appendix G: Tutorial for learning JavaScript programming through storytelling 274
Appendix H: Tutorial for learning Python Turtle programming with visual design 279
Appendix I: Tutorial for learning HTML5 / JavaScript programming with visual design 281

Appendix J: Using Tkinter to add a Graphical User Interface (GUI)..284
Appendix K: Programming test (HTML5/CSS and Python)..286
Appendix L: Computer Programming Confidence and Anxiety (CPCA) Questionnaire 299
Appendix M: Student Perceptions of Computer Science (SPCS) Questionnaire........................303
Appendix N: Influences on IGCSE Choices (IIC) Questionnaire 306
Appendix O: Preferred Computer Programming Learning Style (PCPLS) Questionnaire.........310
Appendix P: Semi-structured questions for the group interviews..313
Appendix Q: Lichtman's Stage 1 - Initial comments..315
Appendix R: Group interviews and emerging themes..316
Appendix S: Emerging Themes...342
Appendix T: Word Tree for the word 'parent' 343

List of Figures

Figure 2.1 Example of the 'Rainfall Problem' algorithm shown in Python................................54
Figure 2.2 Example of error diagnostics in Python 62
Figure 2.3 Example of an algorithm shown as a flowchart diagram and in Java 62
Figure 3.1 Spreadsheet validation for marking the programming assessment 90
Figure 4.1 Chart showing Girls' and Boys' Test Scores, relative to a normal distribution.......109
Figure 4.2 Bar chart showing Girls' and Boys' Test Scores segmented into groups of 10%....111
Figure 4.3 Chart showing mean scores of confidence by gender (scale of 1 – 7) and the higher the number the less the confidence 113
Figure 4.4 Chart showing mean scores of anxiety by gender (scale of 1 – 7) and the higher the number the less the anxiety 115
Figure 4.5 Participants' computer gaming minutes per week 119
Figure 4.6 Participants' responses to the preferred game genre..120
Figure 4.7 Factors that influence participants' IGCSE choices..123

List of Tables

Table 2.1 Types of high-level computer programming languages...29
Table 3.1 Timeline of teaching pedagogies...80
Table 3.2 Data collection timeline...87

Table 4.1 Girls' and Boys' Intelligent Quotient (IQ) Scores and Test Scores 109
Table 4.2 Responses to the Computer Programming Confidence and Anxiety Questionnaire 112
Table 4.3 Responses to the Influences on IGCSE Choices Questionnaire 124
Table 4.4 Preferred programming learning style responses showing the mean, standard deviation, skewness *z* value, kurtosis *z* value, Levene's test of homogeneity of variances, and results from a one-way ANOVA and Mann-Whitney U-test 126
Table 5.1 Interviewed students by pseudonym and what subjects they took for IGCSE 129
Table 5.2 Narrative interviews analysed using Lichtman's thematic coding 130
Table 5.3 Frequency of common themes from Question 4 on the Student Perceptions of Computer Science Questionnaire 136
Table 5.4 Frequency of common themes from Question 5 on the Student Perceptions of Computer Science Questionnaire 137
Table 5.5 Learning strategies mentioned in question 11 on the Influences on IGCSE Choices Questionnaire 143
Table 5.6 Strategies for improving learning from question 11 on the Influences on IGCSE Choices Questionnaire 143

Chapter 1 Introduction

1.1 Introduction

Computer programming is widely considered to be an important skill in modern society (Ensmenger, 2012; García-Peñalvo, 2016; Garner, 2002; Major, 2010; Robins, Rountree, & Rountree, 2003), especially because it provides numerous employment opportunities (Hill, 2016). Recently, this has been recognised by the governments of numerous countries (e.g. Finland, Japan, South Korea, United Kingdom (UK), United States (US)), with public primary and secondary schools now providing compulsory programming classes (Guo, 2017). However, for the purpose of this thesis, the focus will be on the UK, the US, and South Korea. The rationale for this decision is that the participants in this study are South Korean middle school students who study at an English-curriculum school designed to prepare them for UK and US universities.

The social issue of increasing the number of females in Computer Science has been researched a great deal over the past 35 years. Misa (2011) in his extensive historical review of women in computing identified five 'explanatory factors' that underpin most interventions and experiments on why there is a gender imbalance in Computer Science. They are: 1) an intimidating computing classroom or workplace; 2) the computing curriculum content focusing on male-preferred topics; 3) a lack of positive role models and mentoring in the classroom; 4) a lack of peer support; 5) the culture of computing is masculine, and throughout its history practices have created an 'outsider' position for women and girls. Yet despite a range of research approaches, the problem remains unresolved.

There are numerous employment opportunities in computer programming (Nager & Atkinson, 2016), yet few women are in a position to take advantage of these opportunities and are the largest underrepresented group in Computer Science (Mitchell, 2013). Specifically, only around one in ten (13%) UK computer programmers and software developers (the largest group of IT specialists) are women (BCS, 2015). This is unlikely to change in the near future, with the Joint Council for Qualifications (JCQ) reporting that 90.2% of A-level Computing students in 2016 were male, with females only representing 9.8% of the 6,242 candidates (JCQ, 2016a). This statistic suggests that there has not been enough done to increase the number of girls choosing to study A-level Computing.

Over the past few years, researchers have continued to investigate the reasons for the gender disparity in computer programming and looked at ways to overcome potential barriers (e.g. Ashcraft, 2014; Beaubouef & Zhang, 2011; Bock, Taylor, Phillips, & Sun, 2013; Cheryan, Drury, & Vichayapai, 2012; Lewis, Anderson, & Yasuhara, 2016). However, strategies for increasing the number of females in Computer Science has met with little success and according to the Harvey Mudd University president, Maria Klawe, the number of women applying to study Computer Science in the US is actually declining (Klawe, 2016).

The gender disparity in US universities is troubling, with women representing just 29% of the 2016 Computer Science bachelor's degree graduates (National Science Foundation, 2017). In the UK, the percentage of women graduating in 2017 with Computer Science degrees was just 15% (WISE Campaign, n.d.). In South Korea, a country characterised by high researcher densities and technological sophistication, as few as 18% of researchers are women (UNESCO, 2017). Although the focus of this research is on the above three countries it is worth mentioning that this is a global issue. In a worldwide survey, the Programme for International Student Assessment (PISA) reported that women only represent 20% of computing graduates in the Organisation for Economic Cooperation and Development (OECD) member countries (OECD, 2015). Nor does it look like this might change, with PISA reporting that less than 5% of girls contemplated a career in engineering and computing (OECD, 2015). These statistics are extremely concerning, not least because it is the moral responsibility of educators to ensure that all demographic groups are given equal opportunities.

The problem of encouraging girls to consider careers in STEM subjects is even more pronounced in South Korea because only 52.7% of working-age South Korean women (aged 15 and older) are in paid employment, compared to 74.7% of working-age South Korean men (Draudt, 2016). Since 2007, when the South Korean government introduced an elective Computer Science curriculum called 'Informatics', the number of school children studying computer programming has remained low, with the percentage of students of both genders who took these classes dwindled to just 5% (Choi, An, & Lee, 2015).

Research on the underrepresentation of women in Science, Technology, Engineering and Mathematics (STEM) disciplines has been controversial with some variation with the STEM disciplines. For example, female graduates are well represented in biological studies and in 2016, represented 57% of the graduates from U.S. universities but only 29% in Computer Science (National Science Foundation, 2017). There are several competing theories of sociocultural and neurobiological causation for the underrepresentation of women in certain STEM fields (e.g. Garcia-Retamero, & López-Zafra, 2006; Legewie & DiPrete, 2012; Reilly, Neumann, & Andrews, 2016; Xu, 2008). One of the challenges for researchers is to separate the socialisation influences of parents from biological preferences (Leaper, 2013) but that is not always possible or desirable (Eagly & Wood, 2013). Moreover, it is beyond the scope of this research to assess neurobiological aspects of programming.

One sociocultural predictor is the type of societal culture, and given that Korea is a Confucian-influenced society, where women are considered subordinate to men (Lim & Meier, 2011), there is potential gender-bias in the computing industry. Teo (2006) found that students in Singapore, another Confucian-influenced society that scores highly in the PISA rankings (OECD, 2015), reported no statistically significant sex differences in computer attitudes, although male students revealed a more positive attitude towards computers than their female counterparts. Certainly, sociocultural norms in South Korea require consideration in this case study. By encouraging women to train in Engineering and Computer Science should help improve their position in the labour market (Silim & Crosse, 2014). This gender balance in the workplace is fundamental to economic advancement (Kabeer & Natali, 2013). Economic policies in South Korea have been developed to lower discrimination in the labour market and in a computer simulation, it was predicted that the planned changes would contribute positively to the participation of females in the workplace and provide per capita income growth (Kim, Lee, & Shin, 2014).

Many people are familiar with the idiom of “closing the stable door after the horse has bolted” meaning that sometimes it is too late to fix something after the problem has occurred. Over 15 years ago, it was argued that the battle to encourage females to apply for Computer Science undergraduate courses was lost before they reached the age to apply for university (Carter & Jenkins, 2001) and little has changed since. Many girls decide against continuing with STEM subjects when they are still in high school, thereby reducing the talent pool well before they have considered a particular career (Silim & Crosse, 2014). It seems reasonable to suggest that many of the unsuccessful attempts in addressing the under-representation of females in Computer Science occur too late and that a positive exposure to programming at a much younger age may help close the technological gender gap (Metz, 2007). DeClue, Kimball, Lu, and Cain (2011) concur with this perspective and recommend that interest in computing must be fostered early if the number of Computer Science graduates is to increase. Moreover, Lee (2015) in a study with 4,680 grade 12 participants in the United States reported that students who took more units in Computer Science courses in secondary institutions were more likely to pursue STEM majors at a university level.

There are a number of proponents (e.g. Hubwieser, 2012; Rushkoff & Purvis, 2011) who suggested that computer programming should be made a compulsory subject. Rushkoff and Purvis (2011) outline three key reasons. Firstly, a young person will develop confidence in using technology that is ubiquitous in work, social life, and the home. Secondly, there are significant cognitive benefits of learning to program (e.g. Park, Hyun, & Heuilan, 2015), which led Liao and Bright (1991), in a meta-analysis on sixty-five existing studies, to conclude that students with computer programming experiences scored about sixteen percentile points higher on a variety of cognitive-ability tests than students who did not have prior programming experiences. Writing a programming solution requires the programmer to solve a problem, often using abstract thinking and thus promotes cognition (Wing, 2006). Thirdly, Computer Science has enormous growth potential and brings numerous job opportunities (Nager & Atkinson, 2016).

In 2013, the Computing curriculum (Gov.UK, 2013a) was introduced and it is now a compulsory subject in the England, Wales, and Northern Ireland public school system in both primary and secondary schools. From 2018, computer programming will be a compulsory subject in South Korean state schools (Cho & Huh, 2017). However, in many English-curriculum International schools students can choose not to study Computer Science. In Year 9, secondary school students are required to make a decision of what to study in their International General Certificate of Secondary Education (IGCSE) qualifications and few girls, relative to boys, choose to study Computer Science (JCQ, 2016b). If a student decides not to study a subject when transitioning from Year 9 to Year 10, they reduce the likelihood of pursuing it at a later point in their life (Cleaves, 2005). Consequently, ceasing to study Computer Science in Year 9 increases the likelihood of never developing important career and life skills.

Female students studying at an English-curriculum International school should be given the same opportunities to study Computer Science as their male equivalents. Thus, the purpose of the current study is to investigate the Computer Science abilities and perceptions of Years 8 and 9 school students (12 – 14 years old) and explain what factors might influence their choosing IGCSE Computer Science in a selective, English-curriculum, independent school located in South Korea. Integral to this research is the desire to shed light on the reasons why girls, relative to boys, do not pursue Computer Science. From this, practical strategies will be formulated to encourage the continued participation of girls, which may, ultimately, help to address the gender imbalance in the Computer Science industry. This has the potential to improve the technology-based systems created by diversifying the views of the designers and by helping to fill the projected shortfall of computer-related jobs (Babes-Vroman et al., 2017; Kelleher, 2006). This gender diversity leads to a broader representation of preferences (Azmat, 2014) and evidence strongly suggests that collaborating teams, irrespective of gender, are increasingly producing the majority of recent scientific innovations (Wuchy, Jones, & Uzzi, 2007) and that group collaboration is greatly improved by the presence of women (Woolley, Chabris, Pentland, Hashmi, & Malone, 2010).

1.2 History of computer programming in English-curriculum schools

Prior to discussing the history of computer programming in English-curriculum schools, the meaning of several terms, specifically *computer science*, *computing*, *computational thinking*, *programming*, and *coding* should be explained. These terms are not synonymous and defining them can help to clarify these concepts in the context of this research.

The two terms of *computing* and *computer science* are sometimes used interchangeably and this may cause confusion. The UK organisation, Computing at School, who provides computing resources for UK schools, suggests that *computing* is "concerned with how computers and computer systems work and how they are designed and programmed" (Berry, 2013, p. 4). However, this definition is not entirely helpful because it could be argued that it describes *computer science* rather than *computing*.

The computing programmes, developed by the National Curriculum in England, clarify these two terms more effectively. They use the terminology, *computing,* to describe the curriculum and states that the key stage 3 (Years 7, 8, and 9) and key stage 4 (Years 10 and 11) curriculum has computer science at its core (Gov.UK, 2013b). This indicates that *computer* *science* is a subset of *computing* and this can be seen in the four aims of the curriculum. The first aim is to ensure pupils "can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation" (Gov.UK, 2013b, p. 1). The second aim also relates to computer science and equips pupils to use *computational thinking*. The third aim is not computer science and prepares pupils to evaluate and apply information communication technology (ICT). The last of the four aims is to ensure that pupils are responsible, competent, confident and creative users of ICT. To summarise, the term *computing* also includes the use of ICT and *computer science* is a subset of *computing*.

The expression *computational thinking* fits under the umbrella term *computer science* and is defined as "aspects of designing systems, solving problems, and understanding human behaviours" (Wing, 2006, p. 6). Wing argued that understanding the world from a computational perspective develops a particular lens to problem comprehension and assists in finding appropriate solutions. Therefore, computational thinking is a way of solving a problem by thinking of it in terms of a computer programming language. *Computing programming* is more than just *coding* because it requires the use of computational thinking (Lye & Koh, 2014). Thus, *coding* can be considered the act of writing programming codes that may, or may not, utilise computational thinking. Now that a clear distinction between the terms *computing, computer science, computational thinking, programming,* and *coding* has been established, we can look at the history of computing programming in English-curriculum schools.

In the 1980s, learning computer programming was considered to be a key educational requirement and an important part of the English-curriculum. It was postulated that by teaching the computer to think, a self-realisation process would occur and children would better understand how they themselves think (Papert, 1980). Essentially, children would develop computational thinking skills and be able to visualise a problem and solve it using computer programming tools.

After the initial excitement, the significance of computer programming diminished in the 1990s and instead of becoming the core subject that many academics were advocating (Mayer, Dyck, & Vilberg, 1986), Information Technology (IT) and Information Communication Technology (ICT) gradually replaced it in the English Curriculum (Esteves, Fonseca, Morgado & Martins, 2010; Jenkins, 1998). There are many possible explanations for this change. At the time, there was limited evidence of transferable skills from the LOGO programming language (Pea & Kurland, 1984; Khayrallah & Van Den Meiraker, 1987), although more recent research refutes this and evidences increased creativity and problem-solving skills (e.g. Jang & Lew, 2011; Pardamean, Evelin, & Honni, 2011). Moreover, there was the need to learn the seemingly endless and useful software applications that could be used to enhance learning in most, if not all, of national curriculum subjects (Cox et. al, 2015).

Computer programming has become a key skill that is as important as reading and writing (Prensky, 2008) and so a new national computing curriculum was introduced in England in September 2013. This curriculum has made is computing a compulsory subject in all public primary and secondary schools (Gov.UK, 2013a). This change in policy has increased girls' exposure to computer programming, a discipline that was not previously available.

This initiative has become global and the 2016 Digital Citizenship Standards for students, created by International Society for Technology in Education (ISTE) now include computational thinking, or the ability to use algorithmic thinking to develop a sequence of steps to create and test automated solutions (International Society for Technology in Education, 2016). Further, the Council of International Schools (CIS), a non-profit organisation committed to high-quality international education, now requires evidence of how students' digital citizenship is attained as one of their accreditation standards (CIS, 2016). This means that an International school, the setting of this study, cannot gain accreditation to this member-community organisation without embedding digital citizenship standards, which include the requirement to teach computer programming. Not all International schools apply for accreditation, and even in an accredited school, students may choose not to continue studying this subject after receiving a basic introduction.

Implementing a new computing curriculum is a challenging prospect and to do so effectively requires suitably qualified computing teachers. One of the government-funded initiatives was to develop the knowledge and expertise of in-service teachers through the Computing At School Master Teacher programme (Sentence, Humphreys, & Dorling, 2014). Master Teachers are experienced teachers who work with around 40 other teachers in their local area, supporting their teaching of the computing curriculum (Smith et al., 2015). Despite evidence of positive outcomes from this programme (Boylan & Willis, 2015), a Royal Society review reported that the majority of computing teachers did not have adequate training and were unprepared (Royal Society, 2017). Specifically, the Royal Society survey reported that only 44% of secondary school teachers felt confident only teaching the parts of the curriculum where there was less of a computer science focus.

The state of the computing curriculum in England is fragile and the recommendation of the Royal Society review is that its sustainability is dependent upon swift and coordinated action by governments, industry, and non-profit organisations (Royal Society, 2017). The review went on to report that the government met only 68% of the recruitment target for computing teachers and that this has been an influential factor in many of the difficulties experienced by schools. Since this current study examines a range of learning strategies for teaching computer programming to novices, it will provide an additional insight into ways that the current computing curriculum can be delivered.

1.3 Context

I have taught Computing at Key Stage 5 (16-18 years of age) since 1996 and one of the greatest challenges has been to introduce computer programming to complete novices who were not given the opportunity to study computing programming in secondary school. In sixteen years of teaching post-16 students, every A-level computing class that I have taught was male-dominated with an average of two or three girls in each group of fifteen to twenty students. Since September 2013, I have been employed by The International School of London (pseudonym), an independent, English-speaking, English-curriculum international school on a South Korean island off the coast of the South Korean mainland. I was employed with the directive to introduce Computer Science at Key Stage 2 (ages 7-11) and Key Stage 3 (ages 11-14) and develop the subject so that IGCSE Computer Science and International Baccalaureate (IB) Computer Science can be introduced to the school curriculum.

The School was opened in 2011 to provide an English-immersed education to South Korean students. Classes are mixed-gender in the Junior School (Year 1 to Year 6, inclusively); single-gender in Year 7 through to Year 11, inclusively; and mixed-gender in Year 12 and Year 13. The overwhelming majority of the students will graduate and move to the USA and UK to study at University. This is common practice in South Korea with 63,710 South Koreans being educated abroad in 2014/2015 (Institute of International Education, 2015).

When the School first opened in 2011, it ran classes of Information Communication Technology (ICT) in the Middle School and Junior School, and students learned how to use software applications, including Excel, Adobe Dreamweaver, and Adobe Photoshop. They did not learn any computer programming until I was appointed to create a new curriculum. Moreover, the students that had studied in South Korean state schools had not learned computer programming since 'Informatics', as Computer Science is called in the South Korean state schooling system, is rarely taught in Middle School (Choi, An, & Lee, 2015).

One of the challenges faced in teaching South Korean students, who have come from the state schooling system, is that they have become accustomed to learning by rote (UNESCO Bangkok, 2017). This could be described as a knowledge-based curriculum as opposed to a competency-based curriculum and has had little room for collaboration. In fact, students in state schools compete with each other and receive a ranking in their class and year group. Therefore, getting students to work with their peers and share knowledge was a challenge in the first year of teaching Computer Science.

In that first year, the subject was introduced to Years 5 through 9, inclusively. Each group was given one lesson of 40 minutes per week, so it was decided to incorporate computer programming only and leave the other computing science topics until Year 10. For the first 8 weeks, students were introduced to computer programming through Scratch, the Massachusetts Institute of Technology (MIT) developed visual programming environment designed to facilitate concepts of computational thinking in a fun and simplified approach (Meerbaum-Salant, Armoni, & Ben-Ari, 2013). The expression 'computational thinking' refers to the thought processes involved in formulating a solution to a problem in a way that utilises the rapid processing capabilities of computers (Wing, 2006). After this introductory period, Microsoft Visual Basic 2010 Express was taught to transition the students from an introductory learning language to a commercial application. Microsoft Visual Basic is a computer programming development tool for building Microsoft Windows and Web applications (Petroutsos, 2008), but it supports the novice programmer by using a graphical user interface (GUI) and has a simplistic programming syntax.

In each lesson, students were required to implement a small sample program by typing the sample programming code into the editor, then test and debug the program until it worked. Once the students had a working program they were then required to adapt it to solve several tasks of increasing difficulty (see Appendix A for an example). Additionally, students were asked to complete 40 minutes of homework each week by working through the JavaScript problems available on the online website, 'Code Academy'. It was impractical to give the students homework in Microsoft Visual Basic since it only runs on Windows personal computers (PCs) and the boarding houses only had iMacs. This grounding in JavaScript not only reinforced key concepts such as selection, sequence, and iteration, it also prepared the students for learning JavaScript in the following school year. Thus, the students had some exposure to computer programming, if only through simple problem-solving exercises.

At the end of the first year, three key points stood out. Firstly, there seemed to be no difference in the programming test scores between boys and girls; secondly, Year 7 girls' average score was better than Year 7 boys', Year 8 boys' and Year 9 girls', which indicated that the age of the students had little bearing on programming ability; thirdly, only 5 girls (13% of the Year 9 cohort) compared to 34 boys (52% of the Year 9 cohort) chose to continue with Computer Science at IGCSE. This gender imbalance is typical of UK schools also, with 48,219 boys (80%), compared to 12,302 girls (20%) completing their GCSE Computing in 2016 (JCQ, 2016b).

A combination of the factors discussed above has provided a unique opportunity to research novice programming at a secondary school and design a curriculum that will encourage more female students to study Computer Science.

1.4 Structure of the thesis

Following on from the introduction, **Chapter 2 (Literature Review)** looks at the position of Computer Science in English-curriculum schools, establishes what is a programming language, and outlines why all students should learn to program a computer, taking into consideration the current career opportunities in Computer Science. Further, this chapter includes an overview of the current literature in order to understand the influence of gender on learning to program a computer and considers why so few girls, relative to boys, choose not to continue with their formal education in Computer Science. The relevant theories are divided into two discrete categories: intrinsic motivators and extrinsic motivators. The intrinsic motivators are further sub-divided into neurobiological sex differences, computer programming ability, computer programming engagement, and computer programming confidence. The extrinsic motivators are further sub-divided into a consideration of role models, an intimidating environment for females, and the influence of single sex-classes. Finally, this chapter examines the literature on the strategies for learning to program a computer and considers problem-based learning, diagrammatic learning, collaborative and cooperative learning, programming with a storytelling and/or art perspective before finally considering game-based learning.

Chapter 3 (Methodology and Methods) focuses on the research design and outlines the ontology, epistemology, methodology, and methods used by the author. Following this, participants' characteristics and sampling strategies are outlined, the collection and analysis of data explained, and the ethical considerations relating to this case study discussed.

Chapter 4 (Quantitative findings) presents the quantitative empirical data and focuses on the following areas relating to the research questions: i) gender difference in computer programming ability; ii) intrinsic influencing factors affecting optional subject selection; iii) extrinsic influencing factors affecting optional subject selection; iv) preferred method of learning to program.

Chapter 5 (Qualitative Findings) presents the qualitative data and focuses on the following areas relating to the research questions: i) gender difference in computer programming ability; ii) intrinsic influencing factors affecting optional subject selection; iii) extrinsic influencing factors affecting optional subject selection; iv) preferred method of learning to program.

Chapter 6 (Discussion) presents a discussion of the main findings that have emerged. These points are viewed in relation to the key theoretical ideas that relate to gender and education.

Finally, **Chapter 7 (Conclusion)** provides a summary of the findings, outlines what conclusions can be drawn from them and makes recommendations for future research, as well as implications for policy and practice.

Chapter 2 Literature Review

2.1 Introduction

Fundamental to the purpose of this study is to consider why so few middle school girls, relative to middle school boys, choose to study computer programming in senior school. Knowing how children of both genders develop their skills while at school and what factors, including intangibles such as behaviour and self-confidence, influence their choices about their future education and career pathways is crucial (OECD, 2015). This chapter reviews the available literature, focusing predominately on the past 12 years. The structure of the chapter will begin with establishing the role of computer programming in modern society, then considers the effects of gender, social class, and ethnicity on students' career aspirations, before exploring the nature of computer programming, and reviewing of the history of computer programming in the UK Curriculum. Following on from this is a review of the existing research on both intrinsic and extrinsic gender factors that may affect learning to program a computer. Further, a critical discussion of some of the key learning strategies designed to increase female participation in computer programming is presented, and finally, the research questions are outlined.

2.2 Role of computer programming in modern society

Computer Science is a discipline that combines and blends science, engineering, mathematics, and art (Denning, 2005) and although computer programming is an integral part, Computer Science is much broader than just this field (Denning, 2009). The Association of Computing Machinery (ACM) describes Computer Science as a discipline that covers three overarching categories: 1) designing and developing software; 2) developing effective ways to solve problems that incorporate computers; and 3) providing more effective ways of using computers in areas such as robotics, computer vision, or digital forensics (ACM, n.d.). Arguably, the most appropriate strategy for accessing the different dimensions of Computer Science is through programming, because not only is it central to the subject, it is also a window into the related areas outlined above. In agreement with Tucker et al. (2003), programming is as necessary to the study of Computer Science, as literacy is to the study of literature.

To develop computational thinking, a student must learn the principles of computer programming, and there are a number of fundamental reasons why all students should be given the opportunity to learn this discipline: firstly, it is beneficial to the learner in terms of their educational development; secondly, it enables them to live within a world suffused with technology; thirdly, it provides future career opportunities; and fourthly, it is beneficial to society and in particular the technology industry. The last point is emphasised by job market analytics that claims roughly half of U.S. jobs in the top income quartile are in occupations that require applicants to have some computer coding knowledge (Hill, 2016). Opportunities in the U.S. job market has worldwide ramifications since computer programmers can work from anywhere in the world and it is common practice to outsource internationally (U.S. Bureau of Labor Statistics, n.d.).

It is commonly reported that teaching programming to children has positive impacts on their higher-level thinking skills and problem-solving abilities. From a historical perspective, a number of the studies in the 1980s reported an increase in cognition (Mayer & Fay, 1987; Mayer et al., 1986; Lehrer, Randle, & Sancilio, 1989; Papert, 1980). However, empirical evidence linking improved cognition in computer programmers was slow to emerge (Denner, Werner, & Ortiz, 2012), although the self-perceptions of students studying computer programming reveal that they believe their thinking skills have developed (Kalelioğlu, 2015).

If all boys and girls are given the opportunity to learn to program a computer then it should be a question of how to maximise aspects of teaching programming so that students, particularly girls, are less likely to be discouraged when learning computer programming (Kaplan, 2010). Perhaps just as relevant to this study is which programming language will most likely engage novice programmers, especially girls.

There is an increasing shortage of graduates in Science, Technology, Engineering, and Mathematics or STEM as it is commonly called (Olson & Riordan, 2012) and this is concerning, especially as this is an essential part of the economies of many countries (Meeker & Wu, 2013; Robnett & Leaper, 2013; Rothwell, 2013; Schäfer et al., 2013). Of these four disciplines, it would appear that technology and in particular computer programming is being given extreme levels of importance by the U.S. (Obama, 2016), U.K. (Cameron, 2014), and South Korean (Korea Herald, 2014) governments. Since software development projects often employ overseas programmers (U.S. Bureau of Labor Statistics, n.d.) these steps to increase the number of programmers are internationally relevant.

In an effort to embed computer programming into the school education system, the then President Barrack Obama pledged four billion US dollars into increasing the number of students with access to Computer Science classes. He explained that Computer Science in the new economy is not an optional skill and should sit alongside the three 'R's,' of reading, writing, and arithmetic (Obama, 2016). From a similar standpoint, the then UK Prime Minister, David Cameron, stated that mathematics, science, and computer coding should be a priority for schools since this would impact on the country's economic future (Cameron, 2014). A similar strategy in trying to educate children in the art of computer programming is being employed in the UK with all Year 7 students being given a BBC Micro Bit, a stripped down computer similar to a Raspberry Pi, alongside tutorials on how to program this device (Rocks, 2015). In 2018, a new software education curriculum was introduced to South Korean state schools with the requirement that computer programming is taught to students for a minimum of 17 hours per year (Cho & Huh, 2017).

Educating students on how to program a computer may provide them with future opportunities in a highly lucrative field. In a survey conducted by Looksharp, a marketplace for internships and entry-level jobs, the results from 50,000 U.S. college students and recent graduates identified Computer Science followed by engineering as the majors with the highest starting salaries (Parcells, 2015). The rationale given for the high starting salary was the disparity between actual graduates and available jobs.

Encouraging more girls to choose Computer Science in high school is likely to increase the numbers choosing this option at university. If a student opts out of Computer Science in high school they are unlikely to opt back in at university, since they will lack many of the necessary core skills. In turn, if the number of female Computer Science graduates increases, then this is likely to help address the Computer Science/engineering job shortage.

2.3 The effects of gender, social class, and ethnicity on students' career aspirations

Definitions of social class have evolved over the past 40 years. Structural approaches to measure social class are typically conducted through indicators of social economic status such as income, occupation, and education (Wyatt-Nichol, Brown, & Haynes, 2011). Findings from the BBC's 2011 Great British Class Survey, the largest survey on social class conducted in the U.K., identified seven classes that they differentiated by wealth (Savage et al., 2013). For the purposes of this research, the terms *social class* and *social economic status* will be used interchangeably. Since social class is the strongest predictor of educational achievement in the U.K. (Perry & Francis, 2010), any investigations into educational gender differences should be viewed through a lens that includes this factor.

Social class, ethnicity, and gender, and their interactions on the career aspirations of young people has received some attention from academics. In an effort untangle the influences of race/ethnicity, social economic status, and gender on career ambitions of over 22,000 U.S. Grade 8 and Grade 10 children, Howard et al. (2010) analysed data from an online career information system database. Their analysis revealed significant effects for gender and gender-ethnicity interaction for occupations that had been classified as prestigious and for those occupations that had extensive educational requirements. They discovered a clear gender difference in preferences for 16 of the top 20 occupational choices. Boys considered computer programmer, computer engineer, and mechanical engineer in their selections whereas girls did not include any career that related to Computer Science or engineering. Instead, girls selected the professions of acting, fashion design, cosmetology, veterinarian, photography, dancing, psychology, and elementary school teacher. Most importantly, there were no interactions between 'social economic status and gender' or 'ethnicity and gender' when analysing Computer Science-related career aspirations. Thus, the gender difference in computer-related professions was not influenced by social economic status or ethnicity.

Of the OECD member countries, South Korea has the highest gender pay gap, with women earning 63% of what men earn (OECD, 2017). Further, only 56.2% of women in South Korea are in gainful employment (Son, 2017). What makes this gender disparity so surprising is that South Korea is one of the top performing countries in terms of girls' academic performance (OECD, 2017). Although it appears that there is a lack of opportunity for young women in South Korea, the employment rate of Korean women in their 20s exceeds men (Statistic Korea, 2013). However, this figure declines abruptly once women turn 30 years of age (Statistic Korea, 2013). Since 30 years is the mean age of marriage in South Korea many newly married women leave their jobs to bring up their child and their opportunities to return to their professions are limited (Kim, O'Brien, & Kim, 2016). Thus, young Korean women face societal pressure to either pursue a professional career or to get married and dedicate their life to their family (Kim & Lowry, 2005). This cultural influence complicates any gender-based research in Korea.

The South Korean gender disparity also applies to degrees awarded in Computer Science and mathematics, despite the fact that girls outperformed boys in mathematics in the PISA 2015 assessments (OECD, 2017). In an effort to ascertain whether gender differences in science-related career aspirations were influenced by social economic status and other factors, Shin et al. (2015) analysed the PISA data of 15-year old Korean students. They found that gender differences in social economic status did exist. Although they found that the boys' parents had higher a higher social economic status they attributed this to the Confucian tradition of wealthier parents reproducing until they had a son. Shin et al. (2015) also found that gender differences in science-related career aspirations were influenced by girls lower self-belief and that their parents' perceived science to be of lower importance for their daughters.

It can be seen by these extensive studies in the U.S. and South Korea that there is a gender disparity in science-based career aspirations. In the U.S., there is no evidence that social economic status influences the gender disparity. Conversely, in South Korea, social economic status is one of the factors that influence science career aspirations but it is not the only factor.

2.4 Nature of computer programming

Computer programming is a set of coded instructions, essential to perform a series of operations on computers (Hoare, 1969). These coded instructions can be categorised into three overarching types of programming languages: *Machine Languages, Assembly Languages,* and *High-level Languages* (Deitel & Deitel, 2014). A machine language consists of binary numbers only and is the only set of programmable instructions that a computer can directly understand (Dale & Lewis, 2016). Assembly languages are more understandable to programmers than machines languages because they use letters, rather than binary numbers, for their operation codes (Opcodes). However, assembly languages require a translator called an assembler to convert the short mnemonic letter codes to each machine language instruction (Dale & Lewis, 2016). For example, LDA #5 would be an example of an assembly instruction that loads, or transfers, the number 5 from one memory location to a temporary memory location called an accumulator. This is far more meaningful to programmers than a machine code equivalent, which would resemble "1011100100001001". Assembly languages and machine languages are classified as low-level languages and require a detailed understanding of computer architecture (Brookshear & Brylow, 2015).

Relative to assembly languages, high-level programming languages are much closer to natural language and contain instructions that are easier to understand. Computer programming in a high-level language can be described as the act of assembling abstract symbols in a particular sequence to control the computational actions of the computer (Kelleher & Pausch, 2005). Research studies on computer programming with novices, usually, if not always, incorporate a high-level language rather than machine languages or assembly languages. This research focuses on the pedagogical strategies to teach high-level languages to middle school children.

The strengths of high-level languages lie in their abstraction over complex hardware operations, which lead to greater security, higher reliability, and reduced development costs (Frampton et al., 2009). Essentially, this means that the novice programmer does not need to be concerned with details of how hardware operations function because the translator (the software program that converts a high-level language to machine language [Dale & Lewis, 2016]) will take care of the complex hardware operations. This translator takes the form of an interpreter or a compiler with the former providing superior error diagnostics and the latter producing an executable file, one that is in binary and does not require the original program to run (Dale & Lewis, 2016).

2.4.1 Suitable programming languages for novices

There are thousands of computer programming languages (Kaplan, 2010) and to illustrate the range available, table 2.1 categorises a few of the more popular ones by their use. These categories include visual and block-based languages, simplistic syntax languages, general-purpose languages, web development languages, mobile device languages, and database development languages.

Table 2.1

Types of high-level computer programming languages

Language	Description
Visual and block-based languages	
Scratch	Scratch and other block-based languages (e.g. App Inventor and Alice) incorporate colourful visual blocks to avoid the distraction of syntax while supporting the implementation of complex projects (Rizvi, Humphries,

	Major, Jones, & Lauzun, 2011).
LOGO	Logic Oriented Graphic Oriented (LOGO) was created to teach geometric theory by controlling a robotic turtle's movements (An & Park, 2011). Today, it is incorporated into the Python programming language as Turtle.
<u>Simplistic-syntax languages</u>	
BASIC	Beginner's All Purpose Symbolic Instruction Code (BASIC) was developed in 1963 by John Kemeny and Thomas Kurtz (Kelleher & Pausch, 2005). More recently BASIC has evolved into the Microsoft Visual Basic.NET framework, which includes a Graphic User Interface (GUI), drag and drop objects, and detailed error diagnostics.
Ruby	Released in 1995, Ruby not only uses a simple syntax but is also a general-purpose interpreted, object-oriented programming language (Thomas, Fowler, & Hunt, 2013).
Python	Released in 1990, Python uses a simple syntax but is a general-purpose, interpreted, programming language created by Guido Van Rossum (Paulson, 2007). It was reported to be the most commonly used computer programming language in 2015, with 31.2% of the coding population (Devsaran, 2015).
<u>General-purpose languages</u>	
Java	Java is a general-purpose, compiled, object-oriented programming language (Kelleher & Pausch, 2005) developed in 1995 by Sun Microsystems.
C	C is also a general-purpose, compiled programming language with its later versions, C# and C++ supporting object-oriented programming (OOP). As a compiled language the translator checks the typing in the compilation stage before the program starts executing (Paulson, 2007).
<u>Web development languages</u>	
JavaScript	JavaScript is interpreted by the web browser and is used in conjunction with HTML and CSS. The latest versions allowing developers to create dynamic content without using third-party plug-ins (i.e. programs written in other languages) and importing them (Anthes, 2012).
PHP	Hypertext Pre-Processor (PHP) was developed in the mid-90s to create dynamic web pages that are interpreted by a server before the web page is transmitted to the client's computer (Tatroe, MacIntyre, & Lerdorf, 2013).
<u>Mobile device languages</u>	
Java for Android	Java for Android is a variation of the Java programming language, but for mobile apps to run on the Android platform.

Objective C	To develop apps on the iPhone and iPad, Apple developed an object-oriented variation of the C programming language called Objective-C (Yan, Becker, & Hecker, 2011).
Swift	Swift was introduced in 2014 as a superior language to Objective C (Williams, 2014). Since then, Swift made the top 10 in the monthly TIOBE index ranking of popular programming languages (Hein, 2017).
Database development languages	
SQL	The standard language for communication with a database management system is Structure Query Language, commonly referred to as SQL (Tale, 2016). This language is used in conjunction with other programming languages so is incorporated with mobile apps and web development applications. Most applications that stores data in a database format will incorporate SQL

As can be seen in table 2.1, there are a variety of different uses for programming languages with the majority of studies on novice programming utilising block-based languages or simplistic syntax languages. The former is designed to reduce the need to remember syntax by dragging blocks and the latter is designed to reduce the number of instruction sets and unnecessary syntax (Kemeny & Kurtz, 1980). Simplistic syntax languages still require users to memorise programming commands and to type these commands in a particular order.

Programming robots and circuit boards is another strategy that has been employed to teach programming to novices. Whilst robotic programming has been very effective in promoting programming to children, it does not have broad appeal (Kelleher & Pausch, 2005) and is an expensive solution not available to all schools.

As outlined above, there are numerous programming options available. Thus, the decision as to what is the best introduction to computer programming for novices needs to be considered carefully. Further, there are be features of programming languages that girls may find particularly engaging and this should, therefore, be an additional consideration.

The National Curriculum in England (Gov.UK, 2013c) does not recommend a particular programming language to teach to novices. However, a popular off-the-shelf curriculum, called 'Switched on Computing' embeds Scratch, App Inventor, and LOGO in Years 3 through to 6 (RisingStars-uk, n.d.). The UK organisation that developed this curriculum, Rising Stars, claim that their programme is taught in 6,000 UK and International schools, so it seems reasonable that most primary schools will use highly visual programming languages.

There are several studies (e.g. Denner et al., 2012; Fowler, Fristoe, & MacLaurin, 2012; Stolee & Fristoe, 2011) that have considered the use of highly visual children's programming languages to motivate children to learn computer programming concepts. However, only a few studies have considered whether these introductory languages are effective at preparing beginners for the rigours of learning a text-based computer programming language (Dorling & White, 2015). The concept of *mediated transfer* (Perkins & Salomon, 1992), where the learner applies knowledge learned in one context to another, applies in this case. The introductory language should have suitable features to teach the foundational concepts of computer programming in order to support the transition to commercially used programming languages (Kaplan, 2010). Scratch was designed with this purpose in mind and research carried out by Maloney, Peppler, Kafai, Resnick, and Rusk (2008) noted that children (ages 8 – 18 years) were able to learn some computer programming concepts, even when the learning was self-driven and unstructured.

A study conducted in Israel by Meerbaum-Salant, Armoni, and Ben-Ari (2013) assessed how effectively Grade 9 (age 14 – 15 years) boys and girls could systematically learn computer programming concepts using Scratch, also reported that many students could successfully learn important concepts of Computer Science. They stated that there were some difficulties with certain programming concepts, including initialisation, variables, and concurrency, but suggested that these problems would be overcome by modifying the pedagogical process. This study used a constructionist approach that advocates student-centric discovery learning where students use some information they already know to acquire more knowledge (Alesandrini & Larson, 2002). This meant that the students' learning was structured through making computer program projects that covered particular Computer Science concepts.

Although originally intended for an audience younger than university students, Scratch has been used as an introduction to computer programming concepts at university (e.g. Malan & Leitner, 2007). In a study designed to assess the usefulness of Scratch at transitioning students to the powerful, text-based programming languages, Java and C, but with undergraduate students instead of middle school children, Wolz, Leitner, Malan, and Maloney (2009) found that starting with Scratch helped students learn computer programming theories quickly. They reported that, within a week, students could utilise concepts, such as event handling, concurrency, threads, repetition, selection, variables, logic and data types. They rapidly proceeded to comprehend classic data structures and algorithms in Java or C.

To further support the benefits of learning a visual programming language prior to learning a text-based one, Armoni, Meerbaum-Salant and Ben-Ari (2015) found that those Year 11 (ages 15 – 16 years) students who had learned Scratch previously spent less time learning new topics, had fewer learning difficulties, and achieved higher cognitive levels of understanding of most concepts in C# and Java. However, they did temper this support by noting that at the end of the teaching process, there were no significant differences in ability compared to students who had not studied Scratch.

The benefits of teaching highly visual programming languages to children do not extend to simply accelerating the learning of text-based commercial programming languages. Early exposure is likely to combat stereotypes (Carter, 2006) and improve motivation to learn to program (Armoni et al., 2015).

The type of text-based programming language to teach novice programmers also requires consideration. One approach is to teach an object-oriented language, such as Java or C++, and teach the principles of object-oriented programming (OOP) immediately. An alternative is to use a simplistic syntax programming language to teach procedural concepts (e.g. iteration, selection, library importing) using a language like Python or Perl, and then move on to OOP, which is generally more complex (McCane, 2009). In an experimental, quantitative study, focusing these two approaches of teaching introductory computer programming (OOP from the beginning versus procedural following by OOP), Jayal, Lauria, Tucker, and Swift (2011) reported that the second approach was more effective. They suggested that the scaffolding approach of using Python enabled students to initially focus on the crucial basic issues, without being distracted by the additional complexity of object-oriented programming.

Learning to program a computer can be a daunting challenge given the complexity and abstraction of this skill. Some programming languages are designed to support the novice programmer, while others offer a greater range of features. Thus, the selection of the programming language is an important consideration if students are to be sufficiently motivated to make adequate progress.

There has been a plethora of research in Computer Science education, often focusing on students' success in undergraduate Computer Science modules (CS1). Kinnunen and Simon (2012) note that this focus relates to four areas: course content; students' ability to learn (psychological factors, cognitive skills); the learning environment; and pedagogical approaches. This study considers the same areas but has categorised them into intrinsic influencing aspects and extrinsic influencing aspects. The following section explores what intrinsic factors may affect the number of females choosing to study computer programming.

2.4 Intrinsic factors that may influence the number of female computer programmers.

In Chapter 1, evidence of the lack of females in Computer Science, and particularly computer programming, was provided. Over the past forty years or so, Computer Science has been considered a career best suited to men. Women, who were perfectly competent, have been prevented from pursuing careers in Computer Science by an invisible barrier referred to as the 'glass ceiling' (Larkin & Quinn, 2010). This metaphor was originally used to describe women who were unable to advance beyond a certain management level but it has been subsequently used to describe gender or racial discrimination in other fields (Cotter, Hermsen, Ovadia, & Vanneman, 2001). A qualitative survey of women's career narrative in Science was conducted to address the limited research on women's perspectives and strategies for advancement in male-stereotyped disciplines (Amon, 2017). This survey reported that in terms of barriers to achievement, women were not recognized as authority figures and had to work to build legitimacy. This theme was echoed in a survey of women in engineering with participants' reporting that they were consistently underestimated and had to deal with issues of bias and sexism (Smith & Gayles, 2018). Unquestionably, the glass ceiling is still relevant today and may discourage girls from pursuing a career in computing or engineering if they perceived that there is little opportunity for advancement (Kelly, Dampier, & Carr, 2013).

This section of chapter 2 explored the literature to determine why these barriers have developed and identified several intrinsic factors that have emerged. The following areas will be considered: sex and gender differences in computer programming ability; gender engagement in computing; sex differences in computer programming confidence levels, and perception of computer programmers.

2.4.1 Sex and gender differences in computer programming

Prior to discussing sex and gender differences in computer programming, it is important to clarify the terminological difference between sex and gender. According to Eliot (2011), 'sex' is a biological attribute, defined by chromosomes, whereas 'gender' is a social construct that is the sum of all the attributes associated with one sex. Therefore in this section, the term sex has been used when referring to biological differences only and gender when there are other influences including social, cultural, or psychological.

People, in general, have common-sense ideas about the nature of males and females and their place in society, with these beliefs being handed down through the generations (Archer & Lloyd, 2002). Partly because of this, many psychologists are opposed to comparisons between males and females on the grounds that these differences can be interpreted in such a way as to support a misogynist agenda (Halpern, 2012). However, opinions about sex differences in intelligence and cognitive ability permeate society and academic discourses. Baron-Cohen (2004) argued that men and women are suited to different careers because of the difference in cognitive function. Recent developments in technology, most notably the functional magnetic resonance imaging (fMRI) scanner, have led to an increase in neurobiological research on the behavioural and cognitive changes that occur during adolescence (e.g. Kaushanskaya, Gross, & Buac, 2013; Pera, 2014).

In a meta-analysis on sex differences in the structure of the human brain, it was found that areas of the limbic system, including the amygdala and hippocampus, display sex differences (Ruigrok et al., 2014). Since the limbic system regulates hormones and influences emotional states, it is somewhat unsurprising that numerous studies on emotional responses have identified differences (e.g. Montagne, Kessels, Frigerio, de Haan, & Perrett, 2005) and found that women have better memory for emotional events than men (Canli, Desmond, Zhao, & Gabrieli, 2002). These differences may or may not have implications for learning computer programming, but if strategies for teaching computer programming can incorporate emotional content then girls may have a slight advantage over boys.

During puberty, the release of hormones influences the way in which the human brain develops and contributes to behaviour that can bias adolescent choices (Peper & Dahl, 2013). The frontal lobe, for example, is involved in decision-making, impulse control, and working memory and does not fully develop until 16 years to 20 years in females and up to 25 years in males (Njemanze, 2007). Therefore, this disparity in the genders might contribute to poor decision-making and a lack of self-control in many teenage boys.

Whilst it is beyond the scope of this research to use fMRI to identify sex differences whilst programming to solve problems, it is important to acknowledge that hormonal changes in teenagers may influence Year 9 choices, which in turn may influence future career pathways. Brizendine (2007) argues that hormonal changes may influence education choices because of girls' biological desire to develop social connections. Computer programming is, stereotypically, seen as solitary (Master, Cheryan, & Meltzoff, 2016) and so may not be appealing to teenage girls. This seems a reasonable hypothesis since present-day research shows that hormones are crucial for the development of sex-specific childhood behaviour, including toy selections, which until recently were believed to be generated by sociocultural influences (Hines, 2010). In fact, children's exposure to different types of toys seems to generate additional differences between boys and girls when it comes to their cognitive and social development (Ruble, Martin, & Berenbaum, 1998).

There are many researchers who acknowledge that there are differences between the brains of boys and girls (e.g. Eliot, 2011; Giedd, Raznahan, Mills, & Lenroot, 2012; Lenroot & Gield, 2010; Ruigrok et al., 2014), and in a meta-analysis of the relevant literature Halpern (2012) reported that the sex differences impacts upon behaviour. However, the exact relationship between these sex differences and specific behaviours is not always clear. To confound matters even more, adolescent behaviour may be attributed more to cultural norms, parental influence, or stereotypes rather than neurobiology (e.g. Ceci, Williams, & Barnett, 2009; Derks & Krabbendam, 2013; Gunderson, Ramirez, Levine, & Beilock, 2012). The clear message from a review of the literature is that when we research biological sex differences in the teenage brain, we should not ignore potential sociocultural influences, such as social class and ethnicity.

In the early 1980s, there was a suggestion that the lack of women in mathematics and science was due to biological differences rather than social factors (Benbow & Stanley, 1980; Kolata, 1980). In that era, there were a number of studies that reported a gender gap in mathematical reasoning ability in favour of boys (Beckwith, 1983; Loviglio, 1981), which led some to question whether males had a mathematics gene that females did not (Williams & King, 1980). Since a good understanding of mathematics is essential for the abstract study of programming algorithms (Sangwin & O'Toole, 2017), the question as to whether there is a gender gap in computer programming ability is justifiable.

In 1990, Turkle and Papert published an article looking at different approaches to programming and suggested that epistemological approaches to acquiring knowledge were little more than programming styles. They argued that more women than men were concrete thinkers, meaning that they could only think literally, not abstractly, and that concrete thinkers could only go so far in programming (Turkle & Papert, 1990). Of course, the implication of this line of reasoning is that the majority of women are not suited to a career in Computer Science (Ben-Ari, 2001). The prevalence of gender stereotyping led Carter and Jenkins (2002) to conduct a study to determine if it was possible to tell the gender of the programmer by looking at their programming. They found that it is not possible to determine the gender of a student simply by looking at a finished piece of code, and subsequently warned of the dangers of implicit gender-based assumptions.

A further example of gender stereotyping was evidenced in a speech given by Lawrence Summers, the former president of Harvard University. He argued that males outperform females in mathematics and sciences because of biological differences and not social causes (Summers, 2005). More recently, a Google employee posited that there were biological reasons that women were not as successful as men in the technology industry and he subsequently had his contract terminated (Young, 2017). These opinions are somewhat controversial and, although quite generic, it suggests that some people believe that males are hardwired to perform better than females in mathematics and mathematics-related subjects, such as computer programming. There are a couple of studies that claim that male undergraduate students outperform female undergraduate students in introductory computer programming test scores (Rubio, Romero-Zaliz, Mañoso, & Angel, 2015) and STEM subject performance, including Physics (Miyake et al., 2010), but these reports are uncommon.

Spatial ability is an important component of programming in LOGO and this is one area that has been shown to have a gender gap, with males, compared to females, displaying superior spatial abilities (Cherney & London, 2006; Levine, Foley, Lourenco, Ehrlich, & Ratliff, 2016; Reilly et al., 2016). Additionally, Sullivan and Bers (2013), in a study to determine whether girls and boys were equally successful in a series of building and programming robots, reported a slight gender difference in favour of boys when kindergarteners programmed robots. Since the tasks in this study incorporated a form of LOGO programming, the findings support the suggestion that boys have slightly superior spatial abilities. However, where such gender/sex differences are reported they tend to equate to very small differences and largely overlapping distributions of abilities. For example, Ingalhalikar et al. (2014) report that while males have superior spatial abilities, females have superior memory and social cognition skills. Since programming requires memorisation of syntax, males' slightly superior spatial cognition is balanced by slightly inferior memory.

Since the early days of computing, mathematics and programming have been closely linked and undergraduate Computer Science failure rates can be, partially, attributed to a lack of mathematical ability (Owolabi, Olanipekun, & Iwerima, 2014). It is worthwhile considering gender differences between South Korean students in mathematics, since this may prove to be insightful.

International studies of mathematics performance show that the gender gap in mathematics in favour of boys is shrinking (OECD, 2016). The PISA 2012 assessments reported that 15-year-old South Korean boys outperformed 15-year-old South Korean girls in mathematics (OECD, 2015). However, in the three years between PISA testing, mathematics scores dropped more steeply among South Korean boys than among girls. As a result, in 2015, girls outperformed boys, although the difference was not statistically significant. In the OECD member countries, the mathematics gender gap in favour of boys increased in those students who were within the top 10% of performers and in the top 10% in science, boys scored significantly higher than girls (OECD, 2015). Further evidence of this pattern can be seen in the 2016 Mathematics A-level results where more boys (19%) receive A*s, compared to girls (15.2%) (Joint Council for Qualifications, 2016a). This is in line with existing research with Stoet and Geary (2012) reporting a gender disparity, where boys outperformed girls in the top 10% of students in mathematics.

High failure rates in computer programming courses have been linked to a lack of problem-solving skills (Miliszewska, & Tan, 2007), mathematical ability (Gomes, Carmo, Bigotte, &

Mendes, 2006), and specifically attributed to the difficulty students have creating structural algorithms, which are sequences of programming code designed to perform specific tasks (Zehetmeier, Böttcher, Brüggemann-Klein, & Thurner, 2015; Schäfer et al., 2013; Mota, 2007). Some institutes have indicated that failure can be as much as 40% (Yadin, 2011). Reports of gender difference in computer programming ability vary in the literature. In a study on secondary school Malaysian students Mohsin, Norwawi, Hibadullah, and Wahab (2010) found that males scored higher than females when both genders had similar mathematics scores. Contrary to this, Bruckman, Jensen, and DeBonte (2002, January) in a study using a computer-supported collaborative learning (CSCL) environment, reported that gender did not affect computer programming achievement once the additional amount of time boys spend computer programming and their prior experience had been factored into the regression analysis. They argued that boys outperformed the girls only because they had more experience in computer programming and not because they had, inherently, more ability.

In a more comprehensive study, and as part of the PISA 2012 research project, the OECD conducted a computer-based problem-solving assessment with approximately 510,000 participants, aged between 15 years 3 months and 16 years 2 months, from the 65 participating countries. They defined problem-solving as students' capacity to engage in cognitive processing to understand and resolve problem situations (OECD, 2014). They reported that, on average, boys scored consistently higher than girls in problem-solving. Moreover, in South Korea, boys outperformed girls (aged between 15 years 3 months and 16 years 2 months) in a problem-solving assessment (OECD, 2015). As outlined previously, writing a programming solution requires the programmer to solve a problem, often using abstract thinking (Wing, 2006) and consequently, problem-solving is a key requirement of computer programming (Pekrun & Stevens, 2012). If it transpires that there is a gender disparity in computer programming performance, the findings from the PISA 2012 research on problem-based learning should be considered as an important factor. Conversely, if there is no gender disparity in favour of boys then the relevance of problem-solving abilities for novices learning computer programming should be questioned.

2.4.2 Gender differences in engagement in computer programming

Women who are proficient in STEM disciplines, disproportionately prefer careers in non-scientific / non-mathematical fields (Ceci et al., 2009), thereby implying that ability is not necessarily a contributing factor in choosing a STEM career. Looking at the selections of GCSE in the UK (JCQ, 2016b), it is apparent that certain subjects are female-dominated: Art & Design (122,095 girls compared to 60,990 boys, 67% versus 33%); Performing Arts (15,633 girls compared to 3,043 boys, 84% versus 16%); and Home Economics (27,733 girls compared to 3,763 boys, 88% versus 12%). The disparity outlined above has existing for several years, judging by the annual figures published by the JCQ. Core subjects including mathematics, English, and science are compulsory and so have similar numbers between the sexes. Analysis of these figures suggests that it is highly probable that girls enjoy art-related and social subjects. Thus, the lack of engagement in Computing (5,678 girls compared to 29,736 boys, 16% versus 84%) may simply be a preference for other subjects, or it might be that computing is not being taught with consideration for subjects that engage girls, such as art or social factors. It is a given that if students do not engage with the subject they are not going to learn the content effectively (Carter & Jenkins 2010), so strategies for developing pedagogical engagement have been considered in the literature (e.g. Goode, 2008; Sloan & Troy, 2008; Tew, Fowler & Guzdial, 2005).

It has been postulated that creative and highly visual environments, such as Scratch and Alice, will help support the learning of computing concepts for girls because of their interest in art (Utting, Cooper, Kölling, Maloney, & Resnick, 2010). However, these visual programming languages are not used commercially-used computer programming tools and unlike languages like Python cannot be used to create commercial applications (Peng, Sun, & Tsai, 2014). It is interesting to note that creative programming that incorporates visual tasks and storytelling has been used to encourage girls to become interested in programming (Denner et al., 2012; Kelleher, Pausch, & Kiesler, 2007). Even though Python does not have a visual interface (like the aforementioned programming environment), students can be taught computer programming constructs (e.g. sequencing, repetition, selection, subroutines) through two of Python's visual add-ons, 'Tkinter' and 'Turtle'. Python Tkinter allows programmers to add buttons, images, text boxes, and other objects by writing programming code. Python Turtle allows the programmer to draw lines and shapes to create patterns by writing programming code.

Another hypothesis for why girls are less engaged in programming is that they are less interested in computer games (Hartmann & Klimmt, 2006). However, this is in contrast with user data for the U.S. market, where female gamers make up 41% of the total population

(ESA, 2017). However, these figures relate to the US only and across OECD countries, it was reported that 75% of boys and less than 50% of girls, had played a one-player game on a computer; and 71% of boys but 29% of girls had played online collaborative games (OECD, 2015). Regardless of the reported number of gamers, these statistics may not paint a full picture since the amount of time spent playing games is not represented, nor is the type of games played. Kafai, Heeter, Denner, and Sun (2008) suggest that women tend to play solitaire or casual games, whereas Willoughby (2008) found that the gender gap increases with the frequency of gaming. However, a Greek study by Papastergiou (2009) on the learning effectiveness and motivational appeal of gaming for learning Computer Science concepts found that, despite boys' greater involvement with computer gaming and their greater initial knowledge of the content matter, the learning gains (i.e. the ability to recall computer memory knowledge) that the, 16-17 years old, boys and girls achieved did not differ significantly. Moreover, the game was found to be equally motivational for both genders.

2.4.3 Gender differences in computer programming confidence

There are several psychology-related terms relating to learners' self-confidence or anxiety. Prior to reviewing the literature on how confidence and anxiety affect females in STEM, and specifically computer programming, it is important to define some of these terms.

Self-concept can be defined as people's perception of themselves formed through life experiences (Marsh & Shavelson, 1985). Self-efficacy is a person's belief that they can successfully perform a given task at the required level (Ferla, Valcke, & Cai, 2009; Horzum & Çakır, 2009). Both self-concept and self-efficacy are useful predictors of motivation, emotion, and performance (Bong & Skaalvik, 2003). There is a reciprocal relationship between anxiety and self-concept, at least in mathematics, where higher self-concept leads to lower anxiety, which in turn, leads to higher self-concept (Ahmed, Minnaert, Kuyper, & van der Werf, 2012). Moreover, the influences of self-efficacy and anxiety are believed to be key factors in students' pursuit of STEM careers (Rice, Barth, Guadagno, Smith, & McCallam, 2013) and students' performance in computer-based tasks (Saadé & Kira, 2009).

There are numerous reasons why students' academic performance fluctuates, including cognitive ability, home environment, teaching, and school quality (Lin-Siegler, Dweck, & Cohen, 2017). Even when these considerations are accounted for, self-confidence relating to what it takes to succeed in intellectual pursuits can influence motivation, which in turn affects performance. In a well-known experiment, Stanford University professor, Carol Dweck, divided 150 high school students into two groups. One group was told that intelligence was fixed and the other that intelligence could grow with practice. After completing a series of easy problems the two groups were given more challenging ones. The fixed mindset group, handicapped by the belief that they had reached the limits of their intelligence gave up. The growth mindset group who believed that they could improve their intelligence persevered and many of them solved problems that were meant to be beyond them. One of the conclusions from this study was that students who believed their intelligence could be developed outperformed those who believed their intelligence was fixed (Dweck, 2017).

In a practical application of this approach, it has been found that a growth mindset reliably predicts achievement irrespective of social class and gender (Claro, Paunesku, & Dweck, 2016). Further, this study reported that a fixed mindset was even more detrimental to the academic achievement of economically disadvantaged students because those who lack the financial resources needed to overcome greater difficulties to succeed. Potentially, the mindset of novice programmers, and in particular girls, may influence their computer programming learning outcomes. Thus, it is important to be mindful of girls' belief in their ability to learn computer programming because it will likely affect their motivation to learn how to program computers.

Extensive research of self-belief in STEM was conducted by the OECD and they also reported that students' low self-efficacy or the belief that they can complete a task is likely to affect performance, with the PISA 2015 survey indicating that students who have low self-efficacy in science perform poorly in science relative to students who are confident in their ability (OECD, 2016). This study reported that 6% of the variation in science performance could be accounted for by how confident students feel in their science skills and knowledge.

The majority of studies addressing self-concept in STEM disciplines indicate that males have higher self-concept than females (e.g. Beilock & Carr, 2001; Else-Quest, Hyde, & Linn, 2010; Li & Kirkup, 2007; Sax et al., 2017; Skaalvik & Skaalvik, 2004; Steinmayr & Spinath, 2009; Watt, 2005), which in turn affects academic choice, educational aspirations, and academic achievement (Marsh & Seaton, 2013).

Anxiety towards computers has been considered to include psychological, operational, and sociological components (Beckers, Wicherts, & Schmidt, 2007). Anxiety towards computer programming, in particular, is a well-researched topic in undergraduate students. One collection of research findings suggest that females are less confident and more anxious in using computers than their male counterparts (Beyer, Rynes, Perrault, Hay & Haller, 2003; Broos, 2005; Cooper & Weaver, 2003; Durndell & Haag, 2002; Todman & Day, 2006). Conversely, the odd research paper found that females are more confident and less anxious than males (King, Bond, & Blandford, 2002). Perhaps the differences in attitude to computers are negligible and unsurprisingly some studies found no gender differences in attitudes (Tekinarslan, 2008; Teo, 2006; Wilson, 2002).

It should be noted that this lack of agreement is found in studies conducted between ten and thirty years ago, and none of these studies specifically looked at how anxiety or self-confidence related to computer programming. Furthermore, self-confidence and, specifically, computer programming anxiety have always been inexorably linked since learning to program inevitably involves some kind of trial and error (Fuentes, Anderson, Johanson & Nilsson, 2013). In a relatively recent study with 845 Turkish elementary and secondary student participants, Simsek (2011) found that males had higher computer self-efficacy scores than females. However, computer programming experience has a positive and significant effect on self-efficacy beliefs (Hasan, 2003), so introducing the subject at an early age may increase girls' confidence in continuing to study Computer Science.

There is a plethora of research comparing gender with computer anxiety and confidence and with good reason; since high computer anxiety is likely to reduce an individual's effectiveness when using a computer (Teo, 2006), and this factor is often attributed to the low number of females working in the computer science industry (Singh, Allen, Scheckler, & Darlington, 2007). Further, the occurrence of students, both male and female, determined to limit the amount of programming required in their final year degree options is surprisingly common (Jenkins, 2001), so anxiety towards computer programming is an area that requires further study.

2.4.4 Perception of computer scientists

An alternative explanation for girls' lack of engagement with computing is that of ingrained beliefs due to gender stereotypes. The 2015 PISA survey, which focused predominately on science, reported that gender-related differences in science engagement and career aspirations are related to differences in what boys and girls believe they can do, rather than what they are capable of achieving (OECD, 2016). Moreover, it was reported that many students had stereotypical beliefs about science-related occupations, with Computer Science seen as a *masculine* field and biology as a *feminine* field. These beliefs were accompanied by the view that scientists achieved success due to brilliance rather than hard work. Although the study didn't explore this further, it is not difficult to suggest that the logical conclusion of those two stereotypes is that a male-dominated subject that relies on brilliance must be for males. Nor are the OECD the only ones to suggest that the perception of a masculine culture affects participation in specific STEM fields. In an extensive review of why women are well represented in some STEM disciplines (biology, chemistry, mathematics) but not others (Computer Science, engineering, and physics), Cheryan, Ziegler, Montoya, & Jiang (2017) cited a masculine culture for one of three reasons for underrepresentation. The other two reasons were a lack of early exposure to those three disciplines and gender gaps in self-efficacy.

The belief that women in some STEM disciplines are unfeminine is widespread with Betz and Sekaquaptewa (2012) commenting that this social label that may discourage female students from pursuing these fields. These gender stereotypes may be embedded throughout society, with two studies on feminine appearance and suitability for STEM careers indicating that the more attractive the women the less likely they were perceived to be a scientist (Banchefsky, Westfall, Park, & Judd, 2016).

Negative stereotypes not only reduce the likelihood of pursuing a career in Computer Science, but it can also negatively affect students' performance and feelings of belonging (Cheryan, Plaut, Davies, & Steele, 2009; Shapiro & Williams, 2012). In a study assessing how Computer Science undergraduates in the U.S. assess their fit with their chosen discipline Lewis, Anderson, & Yasuhara, (2016, p. 21) summarised the Computer Science stereotype as "

- *singularly-focused* – Computer Scientists are obsessive at the exclusion of other interests and personal needs
- *asocial* – Computer Scientists lack social skills and tend to work in isolation
- *competitive* – Computer Science courses are competitive and lack collaboration

- *male* – Computer Science is a masculine field and men are innately more capable than women."

If young girls have the perception of computer scientists as being 'brainy', 'non-nurturing', and 'geeky' then this sits in opposition to girls' self-identification as 'normal', 'girly', 'caring', and 'active', and they are less likely to develop an engagement with STEM subjects (Archer et al., 2013). The views that children develop about gender roles are derived from a variety of socialising influences including parents, teachers, peers, and the media (Bandura, 2001). Negative stereotypes, if permeated through to schoolchildren are likely to affect girls' engagement with Computer Science if they perceive it as masculine (Clegg, 2001). Television has been shown to portray female characters in interpersonal roles involved in romance, family, and friends (Lauzen, Dozier, & Horan, 2008), and when female scientists are shown they are far outnumbered by male scientists (Long et al., 2010).

PISA 2015 (OECD, 2016) reported that girls and boys aspirations to work in Science take entirely different paths. The found that boys are twice as likely to work as engineers or scientists; girls are three times as likely to work as doctors, vets, and nurses. Furthermore, PISA reported that these gender differences are reinforced by the biases of parents, teachers, and textbooks. Given the above research, it seems reasonable to suggest that the influence of society may have masculinised Computer Science and negatively affected girls' engagement with this subject.

2.5 Extrinsic factors that may influence the number of female computer programmers

Since the intrinsic factors that may influence girls to opt out of Computer Science careers have been discussed in the literature, it is important to consider any potential extrinsic factors. The following list of extrinsic factors has emerged from the literature, namely, a lack of female role models, an intimidating environment for females, and the influence of single-sex classes in choosing to study Computer Science.

2.5.1 A lack of female role models

A role model is defined as a person in an influential position, who provides an example for other individuals to imitate (Erikson, 1993). The role model, according to organisational career theorists, is critical to individual growth and development (Dalton, 1989; Erikson, 1993) and may influence a career pathway (Perrone, 2002). Thus, role models offer individuals, and especially young people, the opportunity to refine their developing identities by providing them with an image of a person that they would like to become (Gibson, 2004).

It is widely accepted that there is a lack of female role models in the majority of STEM subjects, most notably computing, engineering, and physics (Herrmann, Adelman, Bodford, Graudejus, Okun, & Kwan, 2016; Ramsey, Betz, & Sekaquaptewa, 2013). Historically, there have been numerous female role models in computing, including the pioneering Ada Lovelace and Grace Hopper. However, over the past twenty years or more, the majority of computing role models have been male, with several of them receiving celebrity status. Steve Jobs, Mark Zuckerberg, Bill Gates, and Steve Wozniak have, either, had movies made to celebrate their lives or have appeared on popular TV programmes.

The lack of female computing role models is concerning because, as the chief technical officer (CTO) of Webgrrls International, an organisation that promotes women in technology, suggests the lack of suitable female role models is a self-perpetuating problem (Yusupova, 2014). In part, this can be explained because the absence of females has created situation cues for teenage girls. If a girl with an interest in the STEM disciplines of computing, mathematics, physics or engineering only encounters males in textbooks, on television, or in person then they are likely to believe that this career is not for them. Ultimately, the scarcity of female role models in these fields has decreased girls' sense of belonging and reduced their interest in pursuing STEM majors (Stout, Dasgupta, Hunsinger, & McManus, 2011).

It has been suggested that boys tend to have male role models and girls tend to have female role models (Estrada, García-Ael, & Martorell, 2015) making the research on situation cues all the more relevant. Same gender role models are helpful for women in STEM fields because they have to contend with negative masculine stereotypes (Drury, Siy, & Cheryan, 2011). Even those women who have started chosen to study STEM undergraduate majors encounter few female role models, with the National Science Board in the U.S. reporting that only 24% of STEM full professors were female (NSB, 2016). Having role models for female undergraduates is important because when the student perceived a female professor as positive role models, they identified more strongly with science and did not view it as a male-oriented discipline (Young, Rudman, Buettner, & McLean, 2013). However, if undergraduate women interact with non-stereotypical, male role models then they believe they can achieve greater success in computing, compared to those who interact with stereotypical, male role models (Cheryan, Siy, Vichayapai, Drury, & Kim, 2011). Since computer programming in secondary schools is a new education change, there is little research in schools. However, there is a strong correlation between students' subject enjoyment and their relationship with their teacher (den Brok, Levy, Brekelmans, & Wubbels, 2005).

It has already been established that there are few female Computer Science teachers and so it is important that male teachers have an awareness of gender neutrality in the classroom. Yet, Oleson and Hora (2014) have suggested that teachers teach as they were taught and as a consequence, the problem of masculine influences in the classroom may still exist. A German study (Funke, Berges, Mühling, & Hubwieser, 2015) surveyed 63 Computer Science teachers on their teaching experiences and found that they only had a limited perspective of the influence of gender, with almost one-fifth of them unaware of any gender differences in computer programming. Yet the influence of teachers on the gender roles of their students strongly impacts their students' educational outcomes as well as their roles in contemporary society (Allana, Asad, & Sherali, 2016).

The fact that Cisco Systems, Forbes #15 World's Most Valuable Brand (Forbes, 2015), is running a television series called 'Women Rock IT' (Cisco, 2015) to provide female role models for schoolgirls suggests the importance of having a role model. There are four live stream broadcasts with a total of 11 female speakers who are currently working in the technology industry. In an online conversation, Emma Reid, a Marketing and Communications Manager at Cisco, reported that the number of schoolgirls enrolling in Cisco Networking Academy spikes immediately after one of their live streams.

A note of caution exists with regards to female role models in the STEM fields. Betz and Sekaquaptewa (2012) reported that Grade 6 and Grade 7 (11 – 13 years old) girls found feminine STEM role models demotivating because their combination of femininity and success seemed unattainable. Moreover, the femininity of role models displaying non-STEM specific success was not found to be demotivating, indicating that the feminine cues were not the driving negative driving force. These somewhat counter-intuitive findings, call for a better understanding of the importance of female roles models in STEM fields.

While this research focuses on the influence of role models, the literature makes mention of mentors so some clarification is required. A mentor is a person who provides advice and support to a protégé through a cooperative relationship (Higgins & Kram, 2001) and introduces ideas, theories, activities, or careers in their own field of expertise. A mentor knows you and tries to help you succeed and may, or may not, be a role model (Gibson, 2004). There are over 5000 mentorship programmes for young people in the U.S. (DuBois, Portillo, Rhodes, Silverthorn, & Valentine, 2011) with some designed to provide the opportunity for trainees to gain confidence in their ability to do research in STEM fields (Laursen, Hunter, Seymour, Thiry, & Melton, 2010).

Dryburgh (2000) argues that mentoring has a greater impact than simply having role models. In a longitudinal study with 54,000 US undergraduate participants, Bettinger and Long (2005) found that having female faculty members positively influenced the choice of course and major in quantitative, technical, and science-related disciplines, thus supporting a possible mentor effect. There is no relevance to mentors in this study since there are no female mentors available for the female students. However, the impact of role models will be explored through qualitative research.

2.5.2 Intimidating environment for females

There is a belief that a male-dominated industry or classroom can be intimidating for females and so is avoided by many of them (Misa, 2011). Situation cues, such as being outnumbered in an environment, have been shown to create involuntary stress responses that can make minority members feel vulnerable and less engaged (Major & O'brien, 2005). This phenomenon has been labelled *social identity threat* and is the notion that aspects of a person's social identity may be at risk of being devalued in a particular context (Steele, Spencer, & Aronson, 2002). To determine whether situation cues would affect female undergraduates taking Mathematics, Science, and Engineering (MSE) degrees, an experiment was conducted where the participants watched an MSE conference video depicting either an unbalanced ratio of men to women or a balanced ratio. The female students, who viewed the unbalanced video where there were many times more males than females, reported a lower sense of belonging and less desire to participate in the conference, than did those who viewed the gender-balanced video (Murphy, Steele, & Gross, 2007).

Something as simple as replacing male-typical items or displays (e.g. Star Trek poster, video games) in a Computer Science classroom to non-stereotypical Computer Science objects (e.g., nature poster, phone books) was sufficient to increase female undergraduates' interest in Computer Science to a comparable level of their male equivalents (Cheryan et al., 2009). Of course, this argument suggests that girls cannot like male-typical items, which is untrue since the 2018 statistics report that 45% of video gamers in the U.S. are female (Statista, 2018). However, the argument against including non-stereotypical Computer Science objects is a good one because their presence may be discouraging for some students, irrespective of gender.

There is evidence to suggest that gender bias exists with the computer programming industry and in particular in the open source community (Terrell et al., 2016). Open source refers to computer software that is made freely available to other programmers to distribute or develop. Access to open source software for development is usually through a code-hosting repository, such as GitHub (GitHub.com). The social media aspect of GitHub allows its users to discuss a project publicly so that a mass of experts can collaborate and submit programming code to solve problems. Apart from the significant underrepresentation of women in open source; a 2013 survey of more than 2000 developers, whose gender was known, found that only 11.2% were women (Arjona-Reina, Robles & Dueñas, 2014), and it is shown that submissions by female programmers were rejected more often when their gender was known (Terrell et al., 2016).

In recent years it has become extremely common for undergraduate Computer Science courses to ask for prior computer programming experience, something that presents a barrier for many females, especially since boys, compared to girls, are more likely to have learned computer programming independently (Misa, 2011). Even when females do overcome barriers, such as entrance requirements, they are more likely to transfer from their Computer Science degree than their male counterpart, with a male-dominated environment cited as the biggest influencing factor (Margolis, Fisher, & Miller, 2000). A recent study at the Rutgers University in New Jersey reported that the gender gap increases towards graduation starting with females representing approximately 23% of the first undergraduates taking Computer Science and dropping to 15% by the fourth year. Most noticeably they reported that the largest decrease is between the first year and the second year, with females representing just 17% of the CS2 enrolment.

In the past, the computing curriculum has often focused on male-preferred topics and this has further embedded a masculine culture. Margolis and Fisher (2003) reported an instance where a female student was ridiculed for mistaking the name of a football team with a baseball team. Simply changing the topics to non-gender specific ones is insufficient, and research has shown that females respond positively when the assignments have a real-world social context (Carlson, 2006).

The fact that male Computer Science students are likely to have the opportunity to discuss coursework with friends within a social setting and that female students may experience exclusion and a feeling of isolation (Blum & Frieze, 2005) was identified as an issue in the UK, and so the BCSWomen Lovelace Colloquium was initiated in 2007. This annual one-day event was organised to provide an opportunity for female university students to present and discuss their research and network with other students from across the UK (Dee & Boyle, 2010). Hopefully, events such as this will reduce the feelings of isolation and reduce the dropout percentage of female undergraduates. After all, unhappy students are less likely to perform to their potential (Rodrigo et al., 2009) because affective states, the emotional response to stimulus that broadens or narrows cognitive processes (Harmon-Jones, Price, & Gable, 2012), are an important part of conceptual learning and complex problem-solving, which are both key requirements in computer programming (Pekrun & Stevens, 2012).

It was originally thought that the dropout was only a problem for academic Computer Science, but Misa (2011) reported that that in 2005 women accounted for 29% of the white-collar computing U.S. workforce, a gender-specific tail off of approximately 10 percentage points since the 1980s. A Harvard Business School report analysed the fallout of women in STEM industries in the U.S. and discovered that after women had worked in these industries for approximately 10 years, half of them left the workforce. They rationalised that this mid-career exodus is not the result of preferences but because they were pushed out by the macho work environment, isolation and job pressures (Hewlett et al., 2008).

Since there is evidence to show that an intimidating environment in the computing industry and at university is likely to discourage females choosing Computer Science, it seems plausible to suggest that this might also occur at secondary school despite the scarcity of research in schools.

2.5.3 Influence of single-sex classes in choosing to study Computer Science

Single-sex education supporters comment that there are several components of coeducational settings that reinforce conventional gender role socialisation and inadvertently dishearten girls to the point where their interest in learning, particularly mathematics and science, has waned (Cruz-Duran, 2009). It has been argued that segregating boys and girls increases students' achievement (Karpiak, Buchanan, Hosey, & Smith, 2007; Park, Behrman & Choi, 2013), promotes academic engagement and allows for classes in which learners are similar in terms of physical, mental and emotional development (Gurian, Stevens, & Daniels, 2010). Such has been the political support for this reform that in October 2006, the United States Department of Education (2006) published new regulations governing single-sex education to allow coeducational public schools to offer single-sex classrooms (National Association for Single Sex Public Education, n.d.).

Research on the effects of single-sex education has been equivocal, with a number of studies demonstrating benefits, drawbacks, or no differences. There are even some academics who are strongly opposed to single-sex classes and suggest that it increases gender stereotyping and legitimises institutional sexism (Halpern et al., 2011; Meece, Glienke, & Burg, 2006). Simply separating boys and girls into segregated classes may not improve learning outcomes if teachers are not trained in effective strategies for teaching a particular gender. Spielhagen (2013, p. 7) concurs and states, "Teacher preparation prior to heading single-sex classes is critical".

A further consideration is subject-specificity, and rather than comparing single-sex classes and coeducational classes in all subjects it may be that some subjects suit single-sex classes whilst others do not. With this in mind, Eisenkopf, Hessami, Fischbacher and Ursprung (2015) examined the impact of coeducational and single-sex classes on the mathematical academic performance of female high school students and found that single-sex schooling improved their performance. An additional survey, as part of the same research, revealed that single-sex schooling strengthened female students' self-confidence in mathematics. Conversely, Pahlke, Hyde, and Mertz (2013) found no differences between Grade 8 (13 – 14 years old) South Korean students, in single-sex and coeducational schools, in mathematics and science achievement.

Overall, though, there does appear to be slightly more academic support than opposition for single-sex education. Much of the research should be interpreted with caution, however, because it is difficult to assert, with any degree of confidence, that classes of mixed-gender or single-sex gender are responsible for higher academic engagement (Sax, Riggers & Eagan, 2013).

Undoubtedly, there is a need for additional research in this area and perhaps in more focused areas, such as self-confidence in STEM subjects. Clearly, the under-representation of females in these fields at university and in the workforce is an issue that needs to be addressed. Sax, Shapiro, and Eagan (2011) considered how educational environments contribute to women's beliefs about their STEM-related skills and abilities reported that all-girls secondary schools in the U.S. produce alumni who enter college with marginally more confidence in their mathematical and computer skills than girls, with equivalent backgrounds, who attended coeducational schools. Additionally, Sax, Arms, Woodruff, Riggers, and Eagan (2009) reported that single-sex alumnae were three times more likely than the female graduates of coeducational schools to pursue a career in engineering (4.4% versus 1.4%). Although the 21,236 participants in this research were based in the United States rather than South Korea, it does suggest that having single-sex classes, as is the case in this study, may influence the computer programming confidence levels of the female participants.

It is apparent from the review of the existing literature that the decision to continue studying Computer Science is convoluted and may be influenced by a great many intrinsic and extrinsic factors. The following section examines whether pedagogical strategies for increasing the engagement of girls are effective or not.

2.6 Strategies for learning to program

Direct teaching is inadequate if students are being prepared to meet the challenges of an unknown future (OECD, 2010). Therefore, in an effort to make this subject more accessible to all students, but especially girls, a variety of progressive teaching methods were employed. These were based on what Papert and Harel (1991) coined as a constructionist approach, where the learner builds knowledge by building things that are tangible and shareable (Ackermann, Gauntlett, Wolbers, & Weckstom, 2009).

Given the relatively short period of time that computer programming has been taught in educational establishments, strategies for teaching computer programming are underdeveloped compared to other sciences (Margulieux, Catrambone, & Guzdial, 2016). Moreover, programming is a challenging skill to acquire and consequently contributes to high dropout rates and a low number of Computer Science graduates (Bosch & D'Mello, 2013; Robins et al., 2003). It is generally accepted that learning to read code is easier than writing code and there are a number of reasons for this. Perhaps the most common one is that coding errors stop computer programs from proceeding and an impasse arises. When a student learns to write in their first language, they will make mistakes and it would be expected that an adult would be able to correct misspellings or grammatical errors and provide constructive feedback. Unfortunately, a computer is less forgiving and a small mistake will stop a computer dead in its tracks. This can lead to frustration for the novice programmer, who is often unable to correct his or her errors and is, therefore, unable to progress. Rodrigo et al. (2009) claim that undergraduate students' midterm exam results can be predicted by monitoring factors, such as the average number of errors and attempts at compilation. They suggest that this provides an opportunity to identify students at risk and provide additional support to reduce students' frustration and thereby improve retention rates.

Given the difficulties encountered by novice programmers, a number of strategies (e.g. Self-instructional tutorial approach, problem-based learning, storytelling, programming with a visual design element, game-based learning, diagrammatic learning, collaborative and cooperative learning) have been devised in the belief that they may be effective in improving learning. The rationale for choosing these strategies was based on what was frequently used in the academic literature and what would be acceptable to members of the school's community. It should be noted that these strategies are not used in isolation. For example, Soares, Fonseca, and Martin (2015) conducted a study on introductory programming that incorporated game design, problem-based learning, and participant collaboration. Despite these possible confounding issues, the relative success of these strategies, and others are explored in further detail below.

2.6.1 Self-instructional tutorials

A self-instructional approach has been shown to be an effective method of learning because it is self-paced (Rosenberg, Grad, & Matear, 2003). This strategy is the method of choice for the majority of computer programming textbooks, with the user entering statements of code and then running them. However, textbook examples used to learn programming concepts are often mathematical in nature. This is unsurprising given that many Computer Science undergraduate degrees require A-level/IB Higher Level Mathematics or equivalently rigorous qualifications. It can be supposed that this style of teaching might engage boys more than girls, since, in 2016, more than one-and-a-half times as many boys (56,535 boys compared to 35,628 girls) chose to study A-level Mathematics and nearly three times as many boys (11,054 boys compared to 4,203 girls) chose to study A-level Further Mathematics (JCQ, 2016a).

Most computer programming textbooks follow a linear learning pathway where each chapter introduces a new concept and includes example programs that focus on teaching syntax (e.g. Cunningham, 2014; Dawson, 2010; Hetland, 2006; Johansen, 2016). The structure of these books often has each chapter named after the concept to be learned, such as "Variables and Data Types"; "Conditional Statements"; "Loops"; and "Under-Defined Functions" (Johansen, 2016). Typically, there are questions where the learner is asked to write similar code or alter existing code to complete small tasks. Only if the students adapt the program significantly or create their own program are they actively constructing cognitive models and this is not always the case. Ultimately, the objective is to learn a particular programming language rather than focusing on developing a useful program. Further, the mechanism for learning the language is often mathematical, a discipline where boys report greater enjoyment than girls (Frenzel, Pekrun, & Goetz, 2007). For example, a loop is introduced to read several numbers and work out the average or total of those numbers. In fact, the commonly used test of programming ability, 'The Rainfall Problem' (Soloway, Bonar, & Ehrlich, 1983) does exactly this (see Figure 2.1).

```python
file = open('numbers.txt', "r")
data = file.readlines()
total = 0
count = 0
for line in data:
    num = int(line)
    total = total + num
    count = count + 1
average = total/count
print(average)
file.close
```

Figure 2.1 Example of the 'Rainfall Problem' algorithm shown in Python

The main problem with this style of learning is that it may encourage learners to learn programming language codes rather than computational thinking (Lye & Koh, 2014). In an effort to assess the effectiveness of independent learning of programming concepts by completing puzzles rather than working through instructional tutorials, Harms, Rowlett, and Kelleher (2014) reported that U.S. high school students performed 26% better when completing puzzles compared to completing tutorials, while taking 23% less time to learn the programming concepts. Thus, the teacher should ensure that when self-instructional tutorials are used in the classroom there are opportunities to apply the learning through problem completion. Solving a programming problem is a more effective way of learning programming compared to focusing on programming language tutorials (Petronzio, 2016).

Research that compares gender preferences for different learning styles is sparse but in one such study, Bolliger and Supanakorn (2011) reported that male undergraduates found that tutorials were more helpful in improving their performance in comparison to female undergraduates. Reviewing gender preference of learning style in STEM subjects, Kulturel-Konak, D'Allegro, and Dickinson (2011) found that researching and then testing out implications was the preferred method for both male and female undergraduates. This reinforces the view that students should be required to apply the knowledge obtained in a self-instructional tutorial by solving small problems. Certainly, the effectiveness of the self-instructional approach for learning computer programming requires additional research, especially when the participants are school children rather than undergraduates.

2.6.2 Problem-based learning (PBL)

To understand what the phrase *problem-based learning* means, it is first necessary to define the word ‘problem’. Problems are situations with no immediately obvious solution, and solving problems requires thinking and active learning, often involving experimental interactions so that the problem-solver can learn more about the effectiveness of their strategies (Raven, 2000). Problem-based learning (PBL) is an instructional methodology where students acquire knowledge by resolving complex, realistic problems, usually in collaborative groups (Albanese, 2010). Problem-based learning was originally developed for use in the medical profession, but there are a number of academics (e.g. Ambrósio & Costa, 2010; Haas & Furman, 2011; Nuutila, Törmä, & Malmi, 2005) who have applied this strategy to learning to program a computer.

According to O'Kelly and Gibson (2006), the essence of problem-based learning is that the problem is what initiates the learning. This is certainly the case in learning to program a computer because problem-based learning creates an environment where the student has to acquire new knowledge in order to solve a problem (Looi & Seyal, 2014). The challenge arises if the problem is too simplistic or too complex, which could see the learner becoming disengaged with the process of learning (Duch, Groh & Allen, 2001). However, if the level is appropriate and sufficient scaffolding is provided then this form of constructionist approach is effective and provides a deeper understanding (ACM/IEEE, 2013). Yet, O'Kelly and Gibson (2006) and Hung, Hwang and Huang (2012) believe that there are few educators who can effectively deliver problem-based learning to develop programming knowledge since the majority of literature places emphasis on the learning of a particular programming concept rather than problem-solving. The types of small problems encountered in textbooks or self-instructional tutorials cannot be classed as problem-based learning because, as Peng (2010) describes, the problem must be sufficiently large and complex in order to drive the learning process.

The description of problem-based learning and project-based learning are often used interchangeably, with problem-based learning described as having a more open-ended problem (Jones & Monaco, n.d.). A well-designed project-based task incorporating problem-solving scenarios and scaffolded computer programming instructions is an effective strategy for developing creativity and engaging students (Sáez-López, Román-González, & Vázquez-Cano, 2016); Wang, Huang, & Hwang, 2016).

Problem-based learning is built upon the theory of constructivism (Savery & Duffy, 1995), which is a cognitive approach to learning based on the learner developing their own understanding through experiences and interactions with content or other people (Siemens, 2014). In an Australian study, Kay et al. (2000) reported that it was necessary to redesign their undergraduate Computer Science foundation courses using problem-based learning to address students' negative experiences of computer programming. Their original approach to teaching foundation courses consisted of three lectures, a tutorial, and a two-hour workshop. They claimed that their new framework included broader collaborative problems (e.g. simulating a road network) that incorporated a larger set of problem-solving skills, and that this approach produced more positive feedback from students.

2.6.3 Storytelling

Humans are very good at understanding stories due to the episodic memory of the brain so using storytelling as a means of learning new skills should be an effective strategy (Letonsaari & Selin, 2017). So, if the objective is to encourage more girls to learn to program then it should not be so strongly linked to Mathematics, a discipline where there is a gender disparity in career aspirations (Gaspard et al., 2015). Instead, one approach to develop girls' interest in computer programming is to combine it with language and in particular storytelling. In a UK study, it was reported that girls enjoy reading more than boys (Hopper, 2005) so that this strategy may be a good opportunity for girls to combine their enjoyment of literature with learning computer programming. PISA (OECD, 2012) also reported that girls outperform boys in reading in all countries and economies by the equivalent of a school year. Additionally, the JCQ (2016a) report that more than two and a half times as many girls choose English A-level (61,730 females compared to 22,980 males).

One of the challenges with teaching computer programming via storytelling is ensuring that the students are learning computer programming concepts (Jacob & Warschauer, 2018). If the approach is not carefully planned then each section of the story may use similar programming code. Thus, it is important that the teacher ensures that as the students develop their stories they develop their computer programming knowledge further. So although the research (e.g. Faidi, Freihofer, and Townsend, 2017; Werner, Denner, Bliesner, & Rex, 2009) reports an improved attitude towards computer programming they do not report how effective storytelling is at improving computer programming knowledge.

Storytelling has been used to support novice computer programming (e.g. Kelleher, 2006; Powers et al., 2006) with the idea that contextualising programming will develop a greater interest. Denner, Werner, Bean, and Campe (2005) conducted a study designed to improve girls' self-identity and utilised storytelling as a mechanism for learning computer programming. The findings obtained from sixty-two Grade 6 to Grade 8 Californian female participants suggested that learning to program a story-based game was beneficial because the girls saw themselves as becoming technologically proficient. Other studies have reported similar success with improving girls' attitudes towards computer programming but often they use multiple strategies so it is difficult to say which strategy had the greatest influence. For example, Faidi et al. (2017) reported that Grade 5 girls in the U.S. had a more positive experience of programming when exposed to a combination of storytelling, female role models, and visual programming. Kelleher and Pausch (2007) also reported that storytelling with Scratch was found to motivate U.S. middle school girls (11 – 15 years old) to learn computer programming.

In Korea, a similar study with elementary students who were taught computer programming through Scratch digital storytelling (Park, 2014) found that after the experience, female students had positive perceptions that were similar to male students. Studies of this nature need further investigation because Alice and Scratch, the programming software environment that was used in these studies, are so radically different to text-based programming (e.g. Python). Both Scratch and Alice are relatively easy to use, extremely visual with sprites that resemble cartoons, so the motivation may not entirely be due to the narrative aspect of the studies. Visual programming is likely to increase both girls' and boys' motivation more than text-based programming, so gender comparison research using text-based storytelling is an area that requires further investigation.

2.6.4 Visual design

Learning computer programming can be achieved without using any graphical design elements. Some programming languages use an editor or integrated development environment (IDE) to write programming code that only outputs text-based responses. There are two ways to incorporate visual design with programming: using a programming language that allows the user to design a graphical user interface (e.g. place buttons, text boxes, labels, and images), and using a programming language that outputs lines and shapes that can produce artwork.

Using a programming language that incorporates graphics has been shown to engage students and be an effective tool to teach computer programming concepts (Costa & Miranda, 2017). One such example is Alice, a 3D programming environment designed as a gentle introduction to object-oriented programming. Alice allows users to drag programming blocks to build a 3D world and animate the objects within it (Tabet, Gedawy, Alshikhabobakr, & Razak, 2016). Alice's graphical nature encouraged more undergraduate students to take the programming module and that its use appears to support the weak but conscientious student (Mullins, Whitfield, & Conlon, 2009). Support for Alice is also from the perspective of gender, with Al-Tahat, Taha, Hasan and Shawar (2016) reporting that the graphical user interface of Alice had a positive impact on female students' performance and attitude towards computer programming. Using a 3D-world programming environment, such as Alice, can be used to engage students interested in visual design, storytelling, and game creation.

The alternative approach to using visual design with programming is to use programming code to create patterns and drawings. Many programming languages including as Python, Java, and JavaScript have a set of commands to draw on a canvas. The topics presented in lessons can have a positive or negative influence on students' motivation to learn (Pekrun, 2014) and including visual design as a topic may affect which students want to continue to study a subject. The Joint Council of Qualifications (JCQ) in the UK reported that more than three times as many girls compared to boys complete an A-level in Art and Design (JCQ, 2016a) and nearly three times as many students chose GCSE Art and Design compared to GCSE computing entries (JCQ, 2016b). Taking advantage of this interest in visual design when learning computer programming may prove to be an effective strategy to increase girls' motivation to learn. In a four-year study at Carnegie Mellon University in the U.S., Margolis et al. (2000) found that 44% of the female students associated their passion for computers with other disciplines such as medicine or the arts. Interestingly, only 9% of the male students did this because they view the computer as the object of study rather than as a tool (Fisher & Margolis, 2003).

2.6.5 Game-based learning

The popularity of computer games has been obvious for many years with 65% of U.S. households home to at least one person who plays 3 or more hours of video games per week (ESA, 2017). The Entertainment Software Association (ESA) reported that the total U.S. consumer spending on the video game industry in 2016 was $30.4 billion. Video games play an important role in helping children develop a set of additional media-driven skills and competencies (Kahn & Kellner, 2006) thereby improving technological literacy. Given the popularity of gaming, it is little wonder that they are being used for instructional learning. Taking his one step further, Cunningham (2009) suggests that not only does the use of games in learning increase a girls' technological literacy but they may also generate interest in computer programming.

The process of adding gaming elements to the learning of educational content is called *gamification*, and this approach can be seen in several online educational websites, such as Khan Academy (Morrison & DiSalvo, 2014). Typically, these gaming elements include points, badges, and leaderboards, although it is important to note that people do not play games just for points or badges, they play for mastery and to overcome challenges (Kapp, 2012). Using applications that employ gamification may be a useful way to teach programming definitions. Kahoot™ and other online quiz games that employ gamification are effective in creating engagement and active learning (Plump, & LaRosa, 2017). Moreover, they have been shown to improve examination scores at college and university level (Iwamoto, Hargis, Taitano, & Vuong, 2017).

All things being equal, few would argue that an interested student is far more likely to learn the subject content than one who is not. It is this premise that encouraged Leutenegger and Edgington (2007) to devise a successful and progressive protocol for teaching fundamental programming concepts via two-dimensional game development in Flash and ActionScript. They then transitioned to a more challenging language, like C++, to solidify concepts before moving on to a multi-phase project-based game approach using C++ with OpenGL.

A subset of game-based learning is the emergence of three-dimensional virtual worlds, which Dickey (2003) and Freitas and Neumann (2009) believe offers new and exciting opportunities for teaching and learning. In a study using Second Life, an online virtual world developed in 2003 by Linden Lab, the findings supported the notion that using a virtual world is an effective environment for learning to program a computer (Esteves et al., 2011). Developing this idea further, Florida International University is developing a 3D virtual reality programming language in the hope that this platform will increase the number of women in Computer Science education (Ortega et al., 2017).

Teaching programming through the manipulation of robots, an adjunct of game-based learning, has proven to be an effective way of introducing Computer Science (Kay & Lauwers, 2013; Kay & Moss, 2012; Rétornaz et al., 2013). Their popularity is based upon the excitement that they generate in young children. However, parents' feelings about the benefits of learning to program robots is an important factor in their children's attitudes (Valcke, Bonte, De Wever & Rots, 2010). Given that male parents have a more positive attitude towards robots than female parents (Lin, Liu, & Huang, 2012) this form of programming may favour boys. However, this study was small scale with only 39 participants so further research is required.

Research shows that interactive and goal-oriented instructional computer games in classrooms are extremely popular with students (e.g. Ernst & Clark, 2012; Rieber, 2005). They can be utilised as an effective and motivational learning intervention, regardless of students' gender (e.g. Admiraal et al., 2014; Kebritchi, 2008). The release of the neurotransmitter, dopamine, in significant amounts during gaming, generates satisfaction and happiness, which acts as a motivator (Han et al., 2011). Ultimately, the decision to incorporate game-based learning is based on the learning outcomes, and Hwang, Wu, and Chen (2012) claim that this teaching tool is effective in promoting students' learning achievements.

The idea that learners' enthusiasm for computer games will motivate them to learn Computer Science concepts through game authoring is one supported in the literature (e.g. Ibrahim, Yusoff, Omar, & Jaafar, 2010; Muratet, Torguet, Jessel, & Viellet, 2009; Robertson & Howells, 2008) and this approach may offer a gender-neutral approach to teaching computer programming (Carbonaro, Szafron, Cutumisu, & Schaeffer, 2010). Over a number of years, the U.K. examination boards have included programming questions based on simple games, such as tic-tac-toe and battleships. However, there are some reservations that utilising a game-creation approach could disadvantage girls because, unlike men, women did not expect to proceed from playing games to studying Computer Science at university (Lang, 2010). In an extensive study with 992 learners across 13 schools in the U.K., Robertson (2013) reported that girls' games scored more highly than boys', particularly if the game incorporated storytelling. Worryingly, this study also reported that the girls did not enjoy the experience as much as boys and that a game authoring approach may make pupils less inclined to study computing in the future. This is contrary to other studies, where game construction with a computer programming environment has been shown to support the learning of computer science concepts for girls (e.g. Denner et al., 2012) and provides motivation to learn computer programming (Basawapatna, Koh, & Repenning, 2010). Thus, the question of whether creating computer games as a strategy for teaching computer programming concepts is effective requires further research.

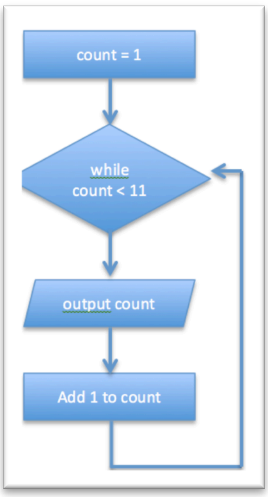
2.6.6 Diagrammatic learning

Diagrammatic learning is a method of learning how to write computer programs by using flowcharts, and other diagrams, instead of programming code. A flowchart is a type of diagram used to represent computer programming algorithms and the flow of data by using geometric shapes (Brookshear & Brylow, 2015). This strategy removes the need to write syntactically correct code and reduces the number of error messages received by computer translators. Since every program that is executed on a computer must eventually be converted into binary (Dale & Lewis, 2016), this process often generates translator error diagnostics (Figure 2.2) due to invalid syntax. So instead of focusing on computational thinking, novice programmers are held back by tiny mistakes such as a missed apostrophe, bracket or comma.

```
Enter a number between 1 and 10a
Traceback (most recent call last):
  File "/Users/admin/Desktop/t.py", line 1, in <module>
    n = int(input("Enter a number between 1 and 10"))
ValueError: invalid literal for int() with base 10: 'a'
```

Figure 2.2: Example of error diagnostics in Python

Teaching pure logic through diagrams and without computers, while keeping programs short, is beneficial for learning computer programming (Henderson, De Palma, Almstrum, Hazzan, & Kihlstrom, 2002). Flowcharts reduce some of the typical obstacles that have hindered novice-programming students (Shneiderman, Mayer, McKay, & Heller, 1977; Wing, 2006). An example of an algorithm which outputs the numbers 1 to 10 inclusively as a flowchart, displayed alongside the equivalent Java programming code, can be seen in Figure 2.3.




```
int count = 1;
while (count < 11) {
    System.out.println("Count is: " + count);
    count++;
}
```

Figure 2.3 Example of an algorithm shown as a flowchart diagram and in Java

The different flowchart symbols indicate the type of instruction: the rectangle represents a process; the rhombus represents a comparison; and the parallelogram represents an output (Watson & Williams, 2015). The text inside the geometric shapes could have been written in a number of ways. In the above example, the statement ‘while count < 11’ could also have been written ‘while count is less than 11’ if clarification of the less than symbol is needed. The same algorithm, in Java, has no such flexibility and the order and location of each symbol has to be precise or the program will not run.

Computer programming requires skills that go well beyond memorising syntax and understanding are better described by diagrammatic representation (Bravo, Marcelino, Gomes, Esteves, & Mendes, 2005). Novice-programming environments that utilise flowchart notation are based on the premise that problem-solving skills can be acquired without the burden of focusing on the syntactic details of programming languages (Crews & Butterfield, 2003b; Crews & Ziegler, 1998; Powers et al., 2006; Xinogalos, 2013; Yuan, Pan, & Zhang, 2008), and logically, it is likely that visual learners would benefit from a diagrammatical approach (Xinogalos, 2013). The burden accompanied by having to learn how to use the Integrated Development Environment (IDE) software, the programming language semantics and syntax, and an abstract way of thinking can be demotivating for beginners (Scott, Watkins, & McPhee, 2007).

Software applications, such as Flowol and Visual-One, have been developed to utilise flowcharts instead of programming code but still execute and provide immediate graphical feedback. Some, such as the Progranimate Programming Aide, allows users to create a flowchart algorithm by adding geometric shapes and the software will generate the programming language syntax from a selection of languages (Scott et al., 2007).

In an empirical study designed to assess gender differences in programming and design a programme that emphasises logic and design, Crews and Butterfield (2003) reported that males outperformed females in programming, but after using Visual-One flowchart tool found an increase in performance for both females and males whilst reducing the differences in achievement between males and females. The participants included 73 U.S. undergraduates in an introductory programming course taught over a 16-week semester. At the end of the semester, the researchers reported that the improved performance was because the flowcharts abstracted out the complexity of hardware and syntax problems, which were some of the impediments that negatively affected the performance of females learning to program a computer. Eventually though, students of computer programming need to learn the syntax of a programming language so it is common for teachers to use flowcharts alongside programming code (Hooshyar, Ahmad, Yousefi, Yusop, & Horng, 2015).

2.6.7 Collaborative learning and cooperative learning

The expressions collaborative learning and cooperative learning are often confused since there is a certain degree of similarity. For this reason, it is important to define these concepts before looking at how they are used in the context of this study. In broad terms, *collaborative learning* refers to a learning environment where two or more people attempt to learn together (Tsai, 2002). From a pedagogical perspective, collaborative learning is encouraged because it is expected that the participants will learn more efficiently than they would individually. Bravo, Duque, and Gallardo (2013) are highly supportive of collaborative programming and assert that they support problem solving, increase confidence and improve the quality of the software solution. If the participants are prepared to support each other, then this way of learning does not impair individual learning activities (e.g. reading, building) that activate recognised learning mechanisms (e.g. induction, deduction). Instead, the interaction among learners generates additional activities (e.g. explanation, disagreement), which elicit additional cognitive mechanisms, such as knowledge production and internalisation (Dillenbourg, 1999). Opponents of this type of learning question whether participants profit equally and express concern over whether boys dominate girls to the extent that it impedes their progress. Prinsen, Volman, and Terwel (2007) conclude that to avoid these circumstances it is necessary to overtly tackle inclusiveness as a characteristic of the collaborative classroom culture.

Paired programming is a form of collaborative learning where two programmers develop a program together but with only one computer and one person entering the code at a given time (Hanks, Fitzgerald, McCauley, Murphy, & Zander, 2011). After an extensive study on paired programming with 555 undergraduates at the University of California, Santa Cruz (UCSC), Werner, Hanks and McDowell (2004, p. 1) concluded that all students, and particularly women, who study in pairs in their beginners programming course are “more confident, have greater course completion and pass rates, and are more likely to persist in computer-related majors”. Their rationale is that work in the information technology industry is competitive rather than collaborative and that Computer Science, in particular, is seen as a solitary occupation. Paired programming emphasises collaboration and helps alleviate the perceptions that a career in computers is not collaborative. Other studies have reported that paired programming increases students' confidence level and enhances students' motivation to learn computer programming (e.g Braught, Wahls, & Eby, 2011; Salleh, Mendes, Grundy, & Burch, 2010).

One of the concerns for teachers implementing paired programming is, which students should be paired together. If one of the two programmers is significantly more competent than the other this may become an issue if their personalities don't allow for an imbalance. Thus, the team members should be selected with personality traits that are beneficial to carrying out paired programming with their particular partner (Maguire, Maguire, Hyland, & Marshall, 2014). Teachers wishing to implement this approach should know their students well enough to predict effective collaboration.

Cooperative learning can be expressed as a set of processes that support group interactions in order to accomplish a specific goal or develop an end product that is typically subject-specific (Panitz, 1999). When a teacher in a classroom environment encourages group-work, students will often divide the tasks and then assemble these individual components to complete the project. This clearly differs from collaborative learning where partners do the work together (Stahl, Koschmann, & Suthers, 2006). Preparing students to develop a computer system cooperatively is an important part of any programming course (Chu & Hwang, 2010) since commercial software products are developed in this way. For example, the computer game 'Grand Theft Auto 5' had 1000 developers working on the project (Makuch, 2013). Collaborative and cooperative strategies are deemed to be beneficial for both boys and girls and are often difficult to separate as learning strategies.

There are numerous strategies that have been employed to assist novice programmers to learn a computer programming language. Some have been considered to be suitable useful strategies for engaging girls, although in some cases combinations of strategies have been employed, which suggests that additional research is required.

2.7 Summary and research questions

Having outlined the relevant literature, it is clear that much of the existing research is either out-dated (i.e. prior to the new UK and US educational reforms for computer programming in high schools) or focuses on university students. Few studies address gender disparity in secondary school children where many potential Computer Science candidates opt out.

The questions in this study consider the factors outlined in the literature review and fundamentally asks the question why so many girls, compared to boys, chose to opt out of Computer Science at the end of Year 9 (13-14 years of age) and after one year of studying computer programming. The order of questions is based on the themes from the literature review and follows a logical sequence. The first question looks at students' computer programming ability; the second question considers intrinsic factors; the third examines external, or environmental, factors; and the fourth question is designed to ascertain if gender differences exist across a range of Computing teaching and learning activities. Thus, the following questions are addressed:

Research Question 1: Is there a significant gender difference in computer programming ability?
Research Question 2: Are there any intrinsic factors (e.g. programming confidence, programming enjoyment, pre-conceived stereotypes) that affect girls choosing to study computer programming in Year 10?
Research Question 3: Are there any extrinsic factors (e.g. lack of role models; peer pressure or support; parental/teacher pressure or support) that affect girls choosing to study computer programming in Year 10?
Research Question 4: Is there a significant gender difference in the preferred learning styles (e.g. programming diagrammatically; individual programming versus collaborative/cooperative programming; programming that incorporates storytelling; programming that incorporates art and design; problem-based learning; game-based learning) for computer programming?

Having completed an overview of the pertinent literature, and identified the research questions, the methodological considerations of this study will now be outlined in the following chapter.

Chapter 3 Methodology and Methods

3.1 Introduction

The methodology for this research was developed to investigate the Computer Science abilities and perceptions of secondary school students (Years 8 and 9) and determine the reasons why a small number of girls, relative to boys, had decided to pursue Computer Science as an IGCSE subject. More specifically, this chapter outlines the study's adopted research approach, concerns with gender research, followed by an explanation of how participants were selected and sampled. The data collection methods and analysis are discussed before the ethics framework is outlined and the ethics approval procedure is detailed.

3.2 Research approach

Social science research investigates human behaviour (Ritchie, Lewis, Nicholls, & Ormston, 2014) and since a major part of this study was to explore the computer programming learning preferences of school children, it can be classified as such. Research has been defined as "a scientific and systematic search for pertinent information on a specific topic" (Kothari, 2009, p. 1) and scientific researchers accomplish this through empirical observation, which is the direct observation of the 'real-world' as opposed to research into ideas or texts, as in the humanities (Newing, 2010). In this research, the 'real-world' is the environment that the participants, who are Korean students in Year 8 and Year 9 at an independent school, live and learn.

For research to have value, it is important for complete transparency when setting out the research design. Findings should not depend on who did the research, but on what was there to be found (Payne & Payne, 2009). For the purposes of this study, Crotty's knowledge framework (Crotty, 1998) provides the guidance for this inquiry into educational research. His framework consists of four elements, which consists of epistemology (linked with ontology), theoretical perspective, methodology and methods. The purpose of sectioning the research process into these elements is to provide researchers with direction because each element informs the others. Since this researcher wanted to use a mixed methodology approach, the data gathering was used to inform the methodology and subsequently the theoretical perspective. Since research should be concerned with the relationship between theory and practice, the position occupied by the researcher should be clarified (Feast & Melles, 2010). Thus, this research takes the following theoretical and practical position:

- Ontology (*Historical Realism*) Epistemology *(Constructionism)*
- Theoretical perspective (*Post-positivism, critical theory*)
- Methodology *(Survey research, quasi-experimental, case study)*
- Methods *(Data collection: pre-recorded IQ Scores, test scores, questionnaires, group interviews; Data analysis: statistical analysis, thematic coding)*

To outline the rationale for this approach, the following sections examine the author's ontological perspective, or how the nature of reality is perceived and interpreted (Cohen, Manion & Morrison, 2013), the author's epistemological viewpoint, or the perspective on how knowledge is acquired (Burrell & Morgan, 2008). Afterwards, the theoretical perspective, or the theories applied to the research, is outlined to ensure that the reader can evaluate them critically. Finally, the methodology, which is the strategy for collecting and analysing data (Kallet, 2004), and the methods, or how the data was collected, are outlined in more detail.

3.2.1 Ontology and epistemology

Ontology and epistemology issues often rise together so the ontological position of the researcher emerges with their epistemological stance (Crotty, 1998; King & Horrocks, 2016). It is for this reason that this section combines these two viewpoints and considers them together. Moreover, it is possible that the ontological and epistemological viewpoint of the researcher may influence the findings of a study (Hitchcock & Hughes, 1995) so clarification is imperative.

Ontology has been defined as the nature of reality (Hudson, & Ozanne, 1988), so a researcher may ask from an ontological perspective, "is reality to be investigated external to the individual or the product of the individual consciousness; whether 'reality' is of an 'objective' nature or the product of individual cognition; whether 'reality' is a given 'out there' in the world or the product of one's mind?" (Burrell & Morgan, 2008, p. 1).

Much of our world-view depends on our concepts of objectivity and the contrast between the objective and subjective. These terms (objective and subjective) have several different senses that include an epistemic sense of the objective-subjective distinction and an ontological sense (Searle, 2007). By this distinction, Searle means that ontologically an experience such as ‘pain’ is subjective because it requires a subject for it to occur, whereas, ‘trees’, for example, are objective because they exist irrespective of an observer or mental state.

Epistemology relates to how the researcher intends to understand or interpret reality (Carson, Gilmore, Perry, & Gronhaug, 2001) and is concerned with the acquiring of knowledge. According to Crotty (1998), there are three main epistemological positions: *objectivism*, *constructionism* and *subjectivism*. Objectivism stresses that social phenomena exist independently of social actors, whereas subjectivism asserts that social phenomena are created from the perceptions of the social actors (Bryman, 2012). Constructionism describes the world as a diverse, unstructured entity concerned with social practices of how people exist within their own world (King & Horrocks, 2016) and that the observation of lived experiences is what shapes reality (Sale, Lohfeld, & Brazil, 2002).

Within the ontological debate, there is the possibility of a middle ground referred to as critical realism (May, 2011), where reality is classed as imperfectly real; and historical realism, where reality is shaped by social, political, cultural, economic, ethnic, and gender values that are solidified over time (Guba & Lincoln, 1994). Ontologically, the author is accepting of historical realism, where the world is a virtual reality shaped by social, political, cultural, economic, ethical and gender values crystallised over time (Guba & Lincoln, 1994). This perspective acknowledges that the perceptions of the investigator and investigated object are interactively linked and the values of the investigator inevitably influence the inquiry (Guba & Lincoln, 1994). The gender disparity in the Computer Science industry has been shaped by society, politics, culture, and economics, and, over the past 40 years, has become embedded in society. Therefore, it is not enough to give females the same opportunity as males, simply because barriers have already been created and need to be overcome for equality to be observed.

An explanation of the epistemological position of this research is further clarified through the description of the theoretical perspective because the methods, methodology, theoretical position and epistemology are linked (Crotty, 1998).

3.2.2 Theoretical perspective

A theoretical perspective is a philosophical stance informed by the epistemological position and informing the methodology (Crotty, 1998). There are several competing theoretical views including the positivist approach, the interpretive view, and several others including critical theory, feminist theory and complexity theory (Cohen et al., 2011). The positivist researcher relies on deductive or 'top-down' logic where hypotheses and theories can be tested through the collection and analysis of quantitative data (Johnson & Onwuegbuzie, 2004). They assert that all phenomena can be explained empirically and that all social science research can be examined objectively (Sale et al., 2002). By comparison, the interpretive researcher employs an inductive or 'bottom-up' methodology that focuses on generated hypotheses arising from information gathered in the field (Frankel & Devers, 2000).

Crotty's framework seems to imply clear distinctions between the three main epistemological positions but it is important to recognise that within each position there are variations and overlaps (Feast & Melles, 2010). The philosophical stance of the author is that of post-positivism, a theoretical view that critiques and evolves positivism to the point where it is accepted that background, knowledge and values of the researcher can influence what is observed (Robson & McCartan, 2016). Further, this perspective or truth of a post-positivist researcher is multi-layered and there exist multiple external realities, which are regarded as subjective rather than objective (Cohen et al., 2011). Post-positivism has an affinity with phenomenological, interpretive approaches to research, in that the truth humans perceive is only accessible through inner subjectivity (Flood, 2010). For example, qualitative data gathered from a participant about their perceptions of Computer Science is interpreted by the research through thematic coding process of the researcher.

A post-positivist theoretical perspective suggests it is the problem under investigation that determines the methodologies needed for its resolution (Demetrion, 2004). Thus, in an effort to be consistent with a post-positive approach, this research will attempt to approximate objective meanings that best reflect the subjective meanings that people hold, whilst accepting that these subjective meanings and interpretations are important. This differs from a subjectivist approach with its variations of interpretivism and a methodology that may include ethnography, grounded research, heuristic inquiry, discourse analysis, action research, or feminist standpoint research (Cohen et al., 2013). Despite this, a post-positivist theoretical approach can be accommodating of interpretive approaches (Crotty, 1998) and so some of the research methods did include qualitative research, including interviews, thematic coding, and open-ended questions.

This research follows a post-positivist critical theory paradigm because, as Carson et al. (2001) explain critical theory researchers and their investigated subject are interactively linked and the belief system of the researcher inevitably influences the inquiry. The intention of critical theory is not merely to report a societal situation but to realise a society that is based on equality and democracy for all its members. Thus, critical theory is concerned not only with understanding a situation or phenomenon but also with providing a solution for changing it (Cohen et al., 2013). Consequently, this research must follow a critical theory paradigm because it is concerned with understanding why so few girls, relative to boys, are choosing not to study Computer Science, with a view to changing this phenomenon.

3.2.3 Methodology

A methodology is a general research strategy that outlines the way in which research is to be undertaken and, among other things, identifies the means and modes of data collection (Howell, 2013). Since this study uses a mixed methods approach to collecting and analysing data, the methodology changes depending upon the nature of the data collection.

In real-world educational research, it is impractical for investigators to conduct controlled experiments and so a *quasi-experimental approach* can be justified (Cohen et al., 2013). Unlike 'true' experimental design where treatments are assigned randomly, quasi-experimental designs assign by self-selection or administrator judgement (Campbell & Stanley, 2011). As a consequence, the quasi-experimental approach may be subject to confounding factors (Reichardt, 2009). One form of quasi-experimental design is *natural experiments*, where there is manipulation of a social setting that can help the researcher to understand the intricate nature of the world by enabling them to study the environments in natural settings (Bryman, 2012). This research loosely follows this form of quasi-experimental design where the teaching interventions are manipulated, and the rationale for this will be discussed in greater detail later. Following this, a survey research methodology is then used for collecting and analysing data.

Survey research is a type of field study that involves the collection of data from a sample of elements from a well-defined population through questionnaires (Visser, Krosnick, &

Lavrakas, 2000). In this instance, the elements are Year 8 and Year 9 students from a population of Korean secondary school students. The methods section outlines the specific details related to the survey research.

This investigation also takes the form of a case study, research that Yin (2014, p. 16) defines as, “an empirical inquiry that investigates a contemporary phenomenon (the ‘case’) in depth and within its real-world context”. If this definition is applied to this empirical investigation into why fewer girls, compared to boys, at an International School in South Korea choose not to continue their education in computer programming and what strategies can be employed to increase their numbers, then it can be seen that this is a case study. As an amendment to Yin’s definition, Woodside’s (2010, p. 1) definition is broader and states, “case study research is an inquiry that focuses on describing, understanding, predicting, and/or controlling the individual (i.e. process, animal, person, household, organisation, group, industry, culture, or nationality)”. In the context of this research there is a focus on understanding and controlling the students by introducing teaching strategies recommended in the literature in an effort to increase subject enjoyment and ultimately increase the numbers of students choosing to continue their computer programming education.

3.2.4 Mixed Methods

Mixed methods research is known by a variety of different names, including multi-trait research, multi-method research, integrated research, combined research, methodological triangulation, and mixed methodology (Creswell & Clark, 2011). Tashakkori and Creswell (2008) suggest that mixed methods research emerged as a practical need to explore all potential avenues, and apply multiple perspectives in order to answer a research question.

Qualitative researchers argue that the voices of participants are not directly heard in quantitative research and that quantitative researchers are in the background, with their own personal biases and interpretations seldom discussed (Creswell & Clark, 2011). Qualitative research can be viewed as deficient because of the researchers’ personal interpretations and subsequent bias, and the difficulty in generalising findings to a large group because of the limited number of participants (Creswell & Clark, 2011). Supporters of the mixed methods approach argue against the polarisation of quantitative research or qualitative research (Ercikan & Roth, 2006) and suggest that mixed methods research encourages a greater dialogue between the proponents, as well as addressing the weaknesses of mono-research models (Denzin, 2008; Brannen, 2005). Not only does mixed methods research address the deficiencies of qualitative and quantitative approaches, it also incorporates the benefits of them. Quantitative data has been utilised because this builds a more complete picture of the phenomenon under study (Denscombe, 2008). Qualitative research and data collection can achieve a more descriptive understanding of a phenomenon, with ethnography, historical and case study research the most common examples (McLaughlan, Nobert, O'Reilly, & Thorkelsson, 2009). Mixed methods research is well suited to the current investigation because quantitative data obtained from test scores and Likert-style questionnaires will provide a gender comparison overview while open-ended questions and interviews will provide greater detail.

One of the main concerns about mixed methods is that a single researcher may find it challenging to undertake both qualitative and quantitative research approaches concurrently (Johnson & Onwuegbuzie, 2004). Many researchers are only confident in quantitative or qualitative methods (Bazeley, 2004) and so attempts to use the other may cause a violation of assumptions and lead to questionable analysis Further, the research process is lengthy and may require a researcher to work as part of a team, which has its own inherent set of problems. Despite these drawbacks, "mixed methods research helps answer questions that cannot be answered by quantitative or qualitative approaches alone" (Creswell & Clark, 2011, p. 12). There is the opportunity to investigate questions in a variety of ways, allowing for a triangulation of complementary results, or creating a deeper understanding when the results are non-complementary (Teddlie & Tashakkori, 2009).

There are three broad categories of mixed methods research (Johnson, Onwuegbuzie, & Turner, 2007): *predominantly quantitatively driven approaches* that use qualitative data to augment the data collection and enhance the analysis; *predominantly qualitatively driven approaches* that use quantitative data to augment the data collection and enhance the analysis; *equal status approaches* place equal emphasis on qualitative and quantitative data collection.

This study followed a mixed methods model, in that it considered both quantitative and qualitative research, although it is predominantly quantitatively driven. The rationale for using mixed methods was that general trends could be established with quantitative research, but these trends could be explored in greater depth using interviews. Of course, collecting data relating to gender has several concerns and the following section considers and discusses these.

3.2.5 Concerns with gender research

Research into gender differences creates a number of specific problems including terminology, conceptualisation, common sense and objectivity (Archer & Lloyd, 2002). The first studies traditionally used the term ‘sex’ when distinguishing between males and females (Unger, 1979), but now ‘gender’ is being used to refer to social distinctions, whereas ‘sex’ is used when considering biological criteria (Eliot, 2011). In order to conform to current practices, this research follows the same conventions.

Since this research will be using a mixed methodology approach, statistical analysis will be conducted on the data and it is important to clarify the term *gender statistics*. Firstly, gender statistics must deal with gender issues and, therefore, directly or indirectly be connected to issues related to the lives of women and men (Genderstats.org, n.d.). There are concerns over the incorrect interpretation of statistics (Field, 2013) and if research into gender is incorrectly interpreted negative stereotypes can be strengthened (Halpern, 2012). Where differences are found, there are two dangerous assumptions in research on sex or gender differences. Firstly, the assumption that if an observable sex or gender difference is demonstrated through ability or behaviour, then this applies to all males and all females rather than just the majority (Hyde, 2005). This is simply not the case and there is an overlap between the groups (Caplan & Caplan, 2015) with males exhibiting more feminine traits and females with more masculine traits. Secondly, there is a belief that gender is understood as something that resides in the individual (Wharton, 2009). Researchers within this framework pay less attention to individual differences between women, or men, and focus more on differences between the two sex categories. Essentially this means that average differences about any group of people tell us nothing about individuals.

Whilst the current research was conducted to support girls’ learning, care was taken to avoid stereotypes, and during interpretation of the results, an attempt was made to eliminate bias by consciously trying to be impartial through reflection. One of the strategies used to avoid stereotypes was to have both teachers of Computer Science read literature on how gender beliefs and bias of educators can inadvertently affect students (e.g. Emilson, Folkesson, & Lindberg, 2016). Raising awareness of others gender bias and discussing it with your colleagues is a good way of reflection on your own beliefs. The second strategy was to try and create a gender-neutral curriculum, although Bejerano & Bartosh (2015) suggest that a hidden gender curriculum in STEM disciplines is often present even when gender-inclusive language is present. Since the classes were single-sex classes there were no concerns about student gender bias. The third strategy to reduce the effects of negative stereotypes was to promote a growth mindset about intelligence by conveying the idea that intelligence is not fixed and will grow with diligent practice (Dweck, 2010). A 20-minute presentation on a growth mindset was given to all of the students in this school and, therefore, all of the participants in this study.

3.3 Participants

The participants in this study are South Korean, are academically selected through Mathematics, English and IQ testing, and are aged between 12 and 14. The participants are from middle and upper class social economic background and, since an above average score in Wechsler Intelligence Scale for Children (Wechsler, 2003) is part of the entrance examination, the variance of intelligence should be smaller than for non-selective schools. These students attend an English-curriculum independent school, are taught in English, and study Computer Science as part of their curriculum. Since the inauguration of the school, there has been a single-sex policy that dictates that boys and girls are taught in separate classes from Years 7 through to Year 11, inclusively. The rationale for single-sex classes has never been explained but finances dictate recombining of genders in Year 12 and Year 13; It is impractical to run a class with only two students of a particular gender so a co-education approach increases the ratio of students to teachers. Furthermore, there is a limit on class sizes and no class can be larger than twenty-four students in Years 7 through to Year 11, inclusively.

The participants are taught in single-sex classes of the following sizes: the 35 Year 8 girls are divided into classes of 17 and 18; the 48 Year 8 boys are divided into classes of 24 and 24; the 46 Year 9 girls are divided into classes of 15, 15, and 16; the 65 Year 9 boys are divided into classes of 20, 22, and 23.

It is important that the participants were not informed of any expectation in the findings since this may influence their efforts. In the 1920s and 1930s, there was a series of experiments that seemed to show that factory workers were more productive in brighter working conditions. Later analysis by Henry A. Landsberger in 1950 showed that the productivity gains seemed to relate to the motivational efforts of being studied rather than the light levels (Collins, 2016).

To ensure that the Hawthorn effect did not influence the results, participants and the parents of the participants were only informed of the overriding research question and not the expectations of the researcher. They were invited to take part in a study to explore the process of how boys and girls learn computing programming, in order to identify strategies to improve student learning (Appendix B).

At the time of gathering data, there were 194 middle school students who agreed to take part in this study: 111 (65 males and 46 females) in Year 9, with a mean age of 14.29 *(SD = 0.28*), and 83 (48 males and 35 females) in Year 8, with a mean age of 13.31 (*SD = 0.24*). The mean age is calculated from the date that data gathering began. This sample took the computer programming test and the questionnaires. Later in the study, a further 12 students from the Year 9 cohort were split into groups and they participated in group-interviews. The group interviews took place the following year and the mean age of the volunteers was 14.57 (*SD = 0.19*).

3.3.1 Participants' prior experience of computing programming

In the year before this study began, computer programming was introduced to the students under the subject heading of Computer Science. Since all students in Years 7, 8 and 9, were new to formal programming lessons, it was decided to teach programming using the same method and offer extension work irrespective of age or gender.

In term one, students were taught Scratch initially and then taught JavaScript. In term two, students continued studying JavaScript and were introduced to Microsoft Visual Basic 2010. In term three, students revised the material and sat the end of year test. Students were only allocated one lesson of 40-minutes computer programming per week but were asked to support their learning by spending a minimum of 20 minutes per week using Code Academy (www.codeacademy.com). Task-based problems completed by the students were recorded every two weeks to monitor progress. The style of teaching in-lesson was predominantly individual, task-based, and student-centric with the key programming concepts (i.e. variables, sequence, selection and iteration) learned through the use of printed tutorial worksheets.

Instruction on how to create a working program was included in the tutorial but then students were expected to adapt their programs to answer related questions. Teacher explanation of the code was made through the use of flowcharts and trace tables. There was no collaborative/cooperative project-based work because the students were uncomfortable with that style of learning. South Korean state education, which all of the students had recently experienced, focuses on competition where all students are ranked by ability. Consequently, attempts to get students to work together frequently ended up with the students working separately. Over time the students became more accustomed to working with their peers but with only 40-minute lessons emphasis was placed on learning as much content as possible in a short period of time.

3.3.2 Sampling

A *sample* is a group of people who are selected to be in the study and non-representative samples cannot be used to make accurate generalisations about a population (Creswell, 2014). A *population* can be defined as the group to which you would like to generalise your findings (Creswell, 2014). In most instances, the population is too large to investigate so the aim of quantitative sampling approaches is to choose a representative sample, in order to establish broad generalisations that can then be applied back to the population (Shaughnessy, Zechmeister, & Zechmeister, 2015). According to Walliman (2011), populations can have the following characteristics: homogeneous (meaning similar or alike); stratified (classified into groups); grouped by type; and grouped by location.

The sample for the current study comprises of the students who have studied one year of computing programming (a single 40-minute lesson each week) and have moved into Years 8 and 9. These students had the same amount of formal programming classes and will follow the same curriculum for a year before data is gathered.

This study does not try to generalise the findings from this sample to all students that study computer programming for the following reasons:

1) The sample is homogenous because the participants are South Korean nationals.
2) The sample is stratified by income. School fees are approximately $26,000 (non-boarding) and an additional $14,000 for boarding per annum, and so only middle-income to high-income families are able to send their children to this independent school.

3) The sample is grouped by intelligence with only those students who score highly on the Wechsler Intelligence Scale for Children being accepted into the school.
4) The sample is grouped by year group and therefore by age, with all participants in Year 8 and Year 9.
5) The sample is grouped by location with the school being situated on a South Korean island in the Pacific Ocean.

Taking into account the above considerations, it can be seen that in this research project the population cannot be all global middle school students that study computer programming. Instead, the population must meet the requirements of the five characteristics listed above.

When choosing the type of sampling it is necessary to consider the type of data being collected. There are two over-riding categories of sampling: probability and non-probability (Newby, 2014). Within these two categories are a number of sampling methods and the method that was chosen for this current study was convenience sampling (also known as availability sampling). This non-probability sampling method uses participants who are conveniently available (Saunders, Lewis, & Thornhill, 2012) and was chosen because all of the students in the Year 8 and Year 9 were being taught computer programming using the variety of teaching strategies. Thus, this convenience sample will be used for collecting all of the quantitative data.

Quota sampling is a non-probability sampling technique that gathers representative data from a group (Saunders et al., 2012). A combination of quota and convenience sampling was used to select the 13 participants for group interviews. Researchers are often advised to follow the principles of data saturation, although practical advice on how to tell if data saturation is reached is lacking in the literature (Carlsen, & Glenton, 2011). In this study, an iterative process of reading through the data transcript after each interview until it was felt that enough data had been gathered. Of the 13 participants for the focus group, 10 were randomly chosen from the original Year 9 group but 3 participants were new to the school. All had started their IGCSEs and had completed 3 months of study in their respective subjects. The number of students was selected to give a balance between an appropriate number to provide an insight into why girls do or don't select Computer Science and the length of time that it takes to interview, transcribe, and analyse the data.

The groups that were selected for interview included: 3 females continuing with Computer Science, 3 females who chose Computer Science but were new to the school, 3 females not continuing with Computer Science, 1 female that switched to Computer Science from another IGCSE subject, and 2 males continuing with Computer Science. Although this research focuses on girls, it was decided that boys' perspective might offer additional insight despite the fact that they were taught in separate classes to girls. Only two boys volunteered to be interviewed otherwise more boys would have been selected.

These 13 participants made up the sample for the five separately conducted group-interviews. These groupings were selected at random but it was decided to separate boys from girls in case it prevented the students from discussing gender-related topics. Initially, there were going to be 4 groups of 3 students who would be selected randomly from the Year 9 girls' classes but it was thought relevant to interview the student who had switched into IGCSE Computer Science. It was important to interview the girl who transferred from IGCSE Chinese to Computer Science because this was an unusual occurrence and required written parental permission. The influence of parents was a consideration in this study so it was thought that unusual event might be enlightening to the student-parent relationship relating to IGCSE selection.

3.3.3 Timeline for the delivery of the teaching strategies

The participants in this study were all taught computer programming in HTML5/CSS and Python using a variety of teaching methods by two male Computer Science teachers. The teachers used the same resources and ran identical lesson plans to ensure a consistent experience for the students. The teachers also had the same number of male classes and female classes as each other. Since the teaching of computer programming was conducted prior to any data collection it is important to explain the timeline that was followed. Table 3.1 shows the order that the different teaching strategies were delivered. They order that they were delivered was mixed to reduce the influence of delivering one strategy before another. After analysing the results, it was felt that the order of teaching strategies had minimal impact on students' questionnaire responses. It should be noted that flowcharts were not taught as a separate topic but, on occasion, they were used to explain some more complex programming code. The decision not to teach flowcharts in isolation was to ensure that the students had a positive experience of learning to program a computer. Tracing flowcharts does not have the same interaction as a computer program running correctly.

When considering the order that the programming languages were delivered, it was decided that Python would be taught first and then HTML/CSS/JavaScript. This is because JavaScript has more exact syntactic requirements than Python. Syntax errors cause frustrations that can create cognitive-affective barriers (Scott & Ghinea, 2013) and slow the learning process. Additionally, Python code is translated using an interpreter, so provides immediate error diagnostic information on the location of the syntax error. Interpreted languages reduce software development time substantially, and although they are often rejected due to performance concerns (Wedekind, Amavasai, Dutton, & Boissenin, 2008), this matters little to novice programmers. Thus, it was expected that students would have fewer syntax errors in Python than in JavaScript and be able to correct them more easily. Moreover, a comparison between programming languages was not the focus of this study so the order did not need to be varied.

The timetabling of classes meant that there were 10 separate Computer Science classes in total. Of this total, there were 5 boys' classes (2 in Year 8 and 3 in Year 9) and 5 girls' classes (2 in Year 8 and 3 in Year 9). Thus, a class of boys and a class of girls were taught with the same strategies and resources at any one time. Table 3.1 shows a timeline for the delivery of the teaching strategies and the data collection points:

Table 3.1

Timeline of teaching pedagogies

Year and Group	Teaching pedagogy / curriculum content	Date
Year 8 Group 1	Tutorials using Python (with flowcharts)	Sept / Oct 2014 (6 weeks)
Year 8 Group 2	Art and design using Python Turtle	
Year 9 Group 1	Art and design using Python Turtle	
Year 9 Group 2	Storytelling using Python	
Year 9 Group 3	Python tutorials (with flowcharts)	
Year 8 Group 1	Storytelling using Python	Oct / Nov 2014 (6 weeks)
Year 8 Group 2	Storytelling using Python	
Year 9 Group 1	Tutorials using HTML/CSS/JavaScript (with flowcharts)	
Year 9 Group 2	Art and design using HTML/CSS/JavaScript	

Year 9 Group 3	Storytelling HTML / CSS / JavaScript	
Year 8 Group 1	Art and design using Python	Dec / Jan 2015 (6 weeks)
Year 8 Group 2	Tutorials using Python (with flowcharts)	
Year 9 Group 1	Art and design using Python Turtle	
Year 9 Group 2	Tutorials using Python (with flowcharts)	
Year 9 Group 3	Art and design in Python Turtle	
Year 8 Group 1	Art and design using HTML/CSS/JavaScript	Feb / Mar 2015 (6 weeks)
Year 8 Group 2	Tutorials in HTML/CSS/JavaScript (with flowcharts)	
Year 9 Group 1	Art and design with HTML5/CSS/JavaScript	
Year 9 Group 2	Tutorials in HTML/CSS/JavaScript (with flowcharts)	
Year 9 Group 3	Art and design in HTML/CSS/JavaScript	
Year 8 Group 1	Storytelling using HTML/CSS/JavaScript	Mar / Apr 2015 (6 weeks)
Year 8 Group 2	Art and design using HTML/CSS/JavaScript	
Year 9 Group 1	Storytelling using Python	
Year 9 Group 2	Storytelling using Python (collaborative / cooperative)	
Year 9 Group 3	Tutorials using Python (with flowcharts)	
Year 8 Group 1	Tutorials using HTML/CSS/JavaScript (with flowcharts)	Apr / May 2015 (6 weeks)
Year 8 Group 2	Storytelling using HTML/CSS/JavaScript	
Year 9 Group 1	Storytelling using HTML/CSS/JavaScript	
Year 9 Group 2	Storytelling using HTML/CSS/JavaScript	
Year 9 Group 3	Tutorials using HTML/CSS/JavaScript (with flowcharts)	

3.3.4 Curriculum content

Computer Science is a discipline covering a range of specialities (e.g. computer architecture, cloud-based computing, graphics, artificial intelligence, data mining, computational science, robotics and software engineering). Despite the title of 'Computer Science', the content of the course for students in Year 8 and Year 9 focuses only on computer programming, a subject that at university would be entitled "software engineering". Through the course of the year, the participants in this study were taught programming using two computer programming languages: Python and HTML/CSS.

Python has been chosen for five main reasons:

1) This language is widely considered to be a relatively easy language to learn due to its simple syntax;
2) Python is a commercially used language, which ensures that students are learning a language that may provide them with future employment opportunities;
3) Python is cross-platform and will work on iMacs and PCs;
4) Python has numerous built-in libraries that allow for graphics to be incorporated in user programs. In a review of the best programming languages to learn, Buckler (2015) reviewed four independent surveys and determined that Python was the fourth best language to learn. The three languages ranked above Python were Java, JavaScript and PHP. However, these languages have a more complex syntax than Python and are likely to be more difficult for novice programmers to learn because they often have trouble translating their intentions into syntactically correct statements (Kelleher & Pausch, 2005).
5) Python has several libraries that can be imported for additional functionality, including the Turtle library. This allows students to draw shapes by writing code and, therefore, provide an opportunity to link programming with Art and Design, a discipline that, judging by the high number of female GCSE students (JCQ, 2016b), many girls would seem to enjoy.

HTML / CSS is not technically classed as a programming language but in the latest version (i.e. HTML5), it does have programming capabilities. Additionally, HTML / CSS works well with JavaScript, a scripting programming language that allows the user to embed interactivity and general-purpose programming constructs.

3.3.5 Teaching and learning pedagogies

The methodology for assessing the effectiveness of teaching strategies is problematic, predominantly due to the issues of experimental control. Wise and O'Neill (2009) explain that the sheer volumes of variables cannot be controlled in a way that respects the precepts of each instructional approach. This is especially difficult in the school where this study takes place because the students are required to do 150 minutes of supervised study in the evening. Students in a class learning through one style of teaching would likely share that knowledge with students from another class during evening study time in the boarding houses.

The planning, resources and teaching materials were developed for this research project and were delivered by two Computer Science teachers. Each teacher taught the same number of boys' and girls' classes and both year groups for the same time periods. There was a weekly meeting to discuss progress and a weekly observation of each teacher to ensure consistency of delivery. The outline scheme of work for this programme is described below.

3.3.5.1 Strategy 1 - Self-instructional tutorials

This is the method of learning used in many computer programming textbooks and the student is asked to type example code into the Integrated Development Environment (IDE) or text editor. Once the student has a working program they are then asked to adapt the code to solve specific tasks. This differs slightly from the majority of computer programming textbooks in that each program usually has a purpose other than simply demonstrating one programming concept. This method was used to gain practice in the Python and JavaScript programming languages. The students spent six weeks on each programming language developing their knowledge using tutorials and solving small tasks. A JavaScript example is shown in Appendix C, and the students learned how to loop through the code to remove redundant characters, such as *, from a section of text. The self-instructional tutorials incorporated other learning strategies including game-based learning, collaboration, and diagrammatic learning.

Game-based learning with self-instructional tutorials. Several of self-instructional tutorials focused on creating simple games, so combined a number of computer programming concepts such as importing libraries, iteration, and selection. A Python example is shown in Appendix D, where the students are asked to create a game of 'Rock, Paper, Scissors' in order to learn several concepts together. Giving the tutorial the purpose of producing a game, rather than learning programming concepts only, was to done to make the activity more engaging.

Collaboration with self-instructional tutorials. It is important to note that when working through these problem-solving tutorials, students were given the option to work independently or in pairs. Typically, they completed the task on their own but they were allowed to ask a peer for help if they needed it. The students tried the different strategies of working in pairs, where one typed and the other help correct errors, but, ultimately, they all chose to type in their own programming code. The self-instructional tutorials were not implemented in a cooperative way because the students could not subdivide the tasks and worked on them together at the same time.

Diagrammatic learning with self-instructional tutorials. Flowcharts were used to explain programming concepts without the complexities of syntax. An example of this form of teaching can be seen in the Appendix E tutorial. In some of the self-instructional tutorials, students were able to see a flowchart version of the programming problem to aid their understanding. This strategy was not employed outside of the tutorial approach and students were not required to create flowcharts.

Irrespective of the effectiveness of this intervention, the ability to analyse an algorithm presented as a flowchart is an essential skill in both of the Cambridge International Examinations (CIE) IGCSE Computer Science syllabus (CIE, 2014) and International Baccalaureate Diploma Computer Science syllabus (IBO.org, 2014). Therefore, students were introduced to this strategy to ensure that they developed the skill-set necessary for further study.

3.3.5.2 Strategy 2 – Storytelling

The computer programming languages used to teach computer programming concepts through storytelling were Python and JavaScript. The stories were presented in a text-based environment rather than graphical so that it could be seen whether it was still engaging without the visual aspects of block-based programming languages. Only those students, who finished their stories before the deadline were given the opportunity to add graphics as an extension. For the students learning Python graphics were added by programming the Tkinter library widgets.

Learning to program a computer through collaborative storytelling was planned for two parts of the academic year. Both modules in storytelling ran for six weeks each but the distinction between the modules was the programming language that was taught. In one instance this was Python (see Appendix F for a tutorial) and in the other instance, it was a combination of HTML/CSS/JavaScript (see Appendix G for a tutorial).

The duration of this module was six weeks of 1-hour 20 minutes-per-week timetabled lessons. This was supported by 40 minutes of homework preparation time each week, although most student groups spent far more time than this.

Game-based learning with storytelling. Each group chose their own storyline but the example story was a text adventure game with the main character collecting items that he, or she, could use. There was even a battle scene based on random numbers where the chances of survival increased when a sword was picked-up. Thus, this module also incorporated a game-based strategy.

Project-based learning with storytelling. The difficulty in implementing problem-based learning when learning computer programming is that the students have to understand the abstract method of getting the computer to act out instructions and apply computational thinking to the problem to be solved. Without prior knowledge of programming language syntax and its application to a particular task, learning can become stagnant. To overcome this problem, students writing a storytelling project were given an example with explanations of programming syntax embedded. Once this example had been implemented, the students could adapt the example to match their planned story.

Cooperative / collaborative with storytelling. The storytelling project groups consisted of 3 students, except when the numbers would only allow a group of 2 students. The teacher placed the participants into their study groups, with the students assigned to the same boarding houses for the convenience of homework. The students decided what aspects of the project were cooperative, where they completed a task individually, and what aspects were collaborative where they worked on aspects of the project together. After six weeks of working on a project, the students presented their finished project.

3.3.5.3 Strategy 3 – Visual design

This strategy is based upon the hypothesis that students, and especially girls, will enjoy computer programming more if there is a visual design component. In this study, programming with graphics took two forms: creating artwork by using Python Turtle (Appendix H), and the HTML5 canvas tool (Appendix I); and using HTML/CSS and Python Tkinter (Appendix J), as an extension option for those students that completed their storytelling text adventure in Python within the expected timeframe.

For six weeks, students learned to program in Python by using Python Turtle to produce artwork. The first few lessons required students to work through tutorials in order to develop an understanding of the programming code. Following on from this, they were asked to design and then produce a piece of artwork. This project was an individual piece of work, although students would sometimes help each other, making it more collaborative. Students that wanted to work with a friend were not discouraged from doing so.

An additional six-week module was designed to deliver HTML/CSS/JavaScript by drawing with the canvas tool. This followed the same structure as the Python visual design module, in that students worked through a few tutorials (see Appendix H for an example) and then designed and created their own art. Again, this strategy utilised both self-instructional tutorials and problem-based learning.

It is impractical for investigators to conduct controlled experiments in real-world educational research (Cohen et al., 2013) and in this study the learning strategies could not be isolated. Further, students learn at different rates so extension activities were required to ensure that the high achieving students were not disadvantaged. The extension activity for storytelling was designed to allow students to improve the visual appearance of their projects. Since both Python Tkinter and HTML5 have the capability to add multimedia, those groups that completed their projects early were taught how to add controls, colour and images. Thus, in the interactive adventure story in Python and HTML5 nearly all of the student groups were sufficiently motivated to reach this stage, often spending many additional hours on their projects. The first option of creating artwork using Python Turtle and the HTML5 canvas tool was individual work, although the students were permitted to help each other. Once the students completed several tutorials, they were set an individual project to create a piece of artwork.

3.4 Data Collection

Prior to the data collection for this study, intelligent quotient (IQ) scores using the Wechsler intelligence scale were taken. This was a part of the admissions process to the school. The rest of the data collection was carried out after a year of teaching computer programming using a variety of teaching strategies. Table 3.2 shows the dates that the data collection occurred. The CPCA questionnaire was distributed immediately after the students looked at the computer programming test for 1 minute. The International Baccalaureate examinations give the candidates 5 minutes to read the question paper prior to the examination start (IB, 2017) so asking the students to look at the test and then answer 12 Likert questions before starting the test should have little influence on the outcomes.

Table 3.2

Data collection timeline

Research Activity	Date
Wechsler Intelligence Scale (IQ scores)	School admission
Computer Programming Confidence and Anxiety Questionnaire	01 June 2015
Computer Programming Test in HTML/CSS & Python	01 June 2015
Student Perceptions of Computer Science Questionnaire	07 June 2015
Preferred Computer Programming Learning Style Questionnaire	14 June 2015
Group interviews	21 June 2015
Influences on IGCSE Choices Questionnaire (To coincide with IGCSE options selection)	01 Dec 2015

The remainder of this section is split into a number of subdivisions with an explanation of how the data was collected for each research question.

3.4.1 Data collection - Research Question 1 ('Is there a significant gender difference in computer programming ability?')

To address the first research question *('Is there any difference in computer programming ability between the genders?')*, a programming test was administered to the students and group interviews were conducted to determine if some of the students had any pre-conceived ideas about gender and programming ability.

The test is divided into two sections: writing HTML5/CSS code and programming in Python. The students were given 75 minutes to complete the test and were allowed to access online help files to assist with syntax. Although technically HTML5/CSS is a markup and styling language rather than a programming language, it is commonly taught in Computer Science courses so that this distinction makes little or no difference to the results. Both environments only require a basic text editor in order to write statements using a learned syntax in order to achieve a desired result.

3.4.1.1 HTML/CSS assessment

The reliability of any test is an important consideration and so it was decided that this section of the assessment should be based upon an established framework. The HTML/CSS assessment in this study (Appendix K) follows a similar format to the Web Design section on Cambridge Examinations IGCSE Information Communications Technology (ICT) practical examination (0417/21(22), Paper 2), where the student is given a series of tasks and asked to complete them using a software application of their choice. Completing the tasks in this practical examination does not usually involve writing HTML or CSS and only requires the student to comprehend what is being asked and then produce a solution using a software application, typically Adobe Dreamweaver, to generate the HTML/CSS code. In the CIE IGCSE ICT examination, the student is allowed to use the software application help file, which in the case of Adobe Dreamweaver, is found on the Adobe website. Thus, the examination is designed to incorporate problem-solving and requires an understanding of how to use software to complete a task. The reason why this assessment is part of an ICT examination, rather than a Computer Science examination, is that it does not require the user to write syntactically correct statements and instead requires them to learn to use a software application.

The HTML/CSS assessment (Appendix K) required the participant to comprehend what is being asked and then write syntactically correct HTML and CSS to solve the problems. Students were allowed to use the W3schools website as their help file since it provides correct syntax for commands. Thus, the test does not rely on memorisation and focuses more on problem-solving. Giving students access to an online help file does not guarantee understanding or the ability to apply the information to a problem. Furthermore, students who spend more time looking up syntax will not have sufficient time to complete all of the tasks within the 75-minute assessment.

3.4.1.2 Python assessment

The Python assessment was designed to assess participants' understanding of basic programming concepts: sequence, selection and iteration. This was accomplished by asking the participants to write Python programming solutions to problems on two specific areas: drawing shapes with Python turtle, and replicating a program described by flowchart symbols. The questions are designed to be progressively more difficult and cover topics that have been taught earlier in the year. The test culminates with an optional, standardised, programming question called ‘The Rainfall Problem’, originally created by Soloway et al. (1983). The question asks the student to solve the following problem:

“Write a program that repeatedly reads in integers, until it reads the integer 99999. After seeing 99999 it should print out the correct average. That is, should not count the final 99999.”

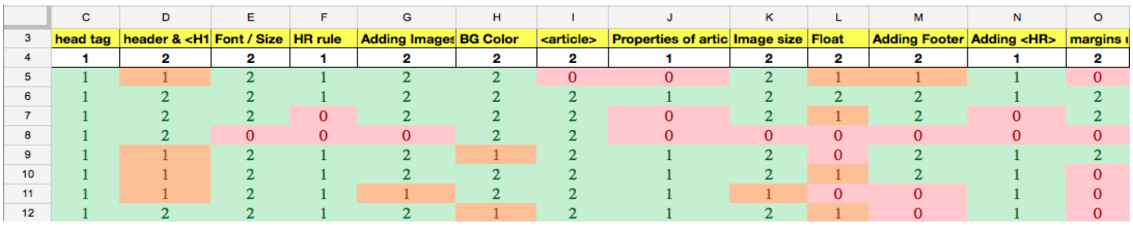
CS1 is the Association for Computing Machinery (ACM)’s introductory course for Computer Science undergraduates (Hertz, 2010), and in the original study (Soloway et al., 1983), the rainfall problem was given to CS1 students three-quarters of the way through the course. The findings were surprising, with only 14% of the participants able to write an acceptable solution to this question. This standardised test was used as the final most challenging question, with only one minor change. Since the students had not been taught to read data from a file, they were asked to input integers instead. This difference in the question has little impact on the overall score since it is anticipated that only the top programmers would be able to gain more than 1 or 2 marks. The rationale for using such a difficult question is to see if there is a gender disparity at the top end of programming ability.

3.4.1.3 Marking the programming assessment

The purpose of the programming test was to assess students’ level of understanding in answering questions on HTML/CSS and Python. All of the students in Years 8 and 9 were issued the same assessment because the range of ability in each year group varied significantly and because they had all studied computer programming for the same amount of time.

To ensure consistency in the marking, only the researcher marked the papers. After the HTML/CSS and Python programming test had been marked once, the student responses were marked a second time to check for marking errors. With 15 years of examination experience with Cambridge International Examinations (CIE) and International Baccalaureate (IB), the researcher is confident that the accuracy of the marking is high, especially since a detailed mark scheme was created.

Once a question had been assessed the mark was immediately transferred into an Excel Spreadsheet where the range of available marks is validated by a colour change (see Figure 3.1). Therefore, if a value outside the expected range was entered, the software would indicate an error had been made and the mark would have to be re-entered. The marks for each participant are automatically totalled using a formula to reduce arithmetic errors.



	C	D	E	F	G	H	I	J	K	L	M	N	O
3	head tag	header & <H1	Font / Size	HR rule	Adding Images	BG Color	<article>	Properties of artic	Image size	Float	Adding Footer	Adding <HR>	margins
4	1	2	2	1	2	2	2	1	2	2	2	1	2
5	1	1	2	1	2	2	0	0	2	1	1	1	0
6	1	2	2	1	2	2	2	1	2	2	2	1	2
7	1	2	2	0	2	2	2	0	2	1	2	0	2
8	1	2	0	0	0	2	2	0	0	0	0	0	0
9	1	1	2	1	2	1	2	1	2	0	2	1	2
10	1	1	2	1	2	2	2	1	2	1	2	1	0
11	1	1	2	1	1	2	2	1	1	0	0	1	0
12	1	2	2	1	2	1	2	1	2	1	0	1	0

Figure 3.1 Spreadsheet validation used for marking the programming assessment

3.4.2 Data collection - Research Question 2 ('Are there any intrinsic factors that affect girls choosing to study computer programming in Year 10?')

To address the second research question, two different online questionnaires were used, namely the Computer Programming Confidence and Anxiety (CPCA) Questionnaire (Appendix L) and the Student Perceptions of Computer Science (SPCS) questionnaire (Appendix M). These questionnaires were set to anonymous responses so that the students would feel comfortable expressing their opinions. As an additional point of data collection, group interviews were conducted to see if a sample of students mentioned intrinsic factors that influenced their decision to continue studying Computer Science.

Questionnaires will always be an intrusion into the life of the respondent, be it in terms of possible invasion of privacy, the level of threat or sensitivity of the questions, or even simply the time taken to complete it (Cohen et al., 2011). Certainly, when addressing personal issues such as self-confidence, care needs to be taken in the wording of the questions.

Questionnaires have the attraction of quickly obtaining opinions from a large target population, generating numerical data that can be processed statistically and providing descriptive information (Dawson, 2009). Despite the fact that questionnaires have several issues, including poor sampling (Walliman, 2011), incorrect or biased responses, and non-response or low response (Durrant, 2009), the benefits outweighed the problems.

The timing for distribution of the two questionnaires was important to ensure that students were not overloaded with too many questions. The Computer Programming Confidence and Anxiety Questionnaire (Appendix L) was administered immediately prior to the programming test and the Student Perceptions of Computer Science Questionnaire (Appendix M) was electronically distributed two weeks before the end of the academic year, which was three weeks after the programming test.

3.4.2.1 The Computer Programming Confidence and Anxiety (CPCA) Questionnaire

The questionnaire to assess computer programming anxiety, or conversely confidence, was adapted from the Computer Anxiety Rating Scale (Heinssen, Glass & Knight, 1987). The original questionnaire was designed to assess anxiety or confidence in carrying out generic computer operations and contained 19 short Likert-style questions. Since the original questionnaire was shown to have high internal consistency and test-retest reliability Heinssen et al. (1987) reported that it was a promising assessment instrument for use in future studies on the nature or treatment of computer anxiety.

Seven questions from the original questionnaire were not relevant so where not included, while others were reworded slightly to relate to computer programming. In accordance with Walliman's (2011) recommendation, there was an attempt to avoid bias, stereotyping, prejudice, intolerance, and discrimination. Care was taken with the use of language in the questionnaire because the participants were South Korean nationals with English as an additional language.

This adapted questionnaire (Appendix L) contains 12 Likert questions (with a scale of 1-strongly agree to 7 – strongly disagree) and was used to collect information on the following:

- Year group and gender
- Confidence in writing computer programs and fixing errors
- Ability to study computer programming at a more advanced level in the future
- Confidence to solve the computer test they are about to take

- Students' opinions in boys' and girls' programming ability

A Google Form online questionnaire was constructed and distributed to the participants through their school Gmail accounts. There was no chance of a participant submitting the questionnaire more than once because the setting to prohibit this was enabled. An online questionnaire has the advantage that the student does not need to hand their responses directly to the person administering the questionnaire and may encourage more honest answers (Wright, 2005) because the participants were informed that the anonymity setting was enabled. Thus, it would be impossible to identify a particular student because there were no identifying features such as handwriting.

The *CPCA* Questionnaire was administered immediately prior to the computer programming test. Since all students taking the test had immediate access to a computer and the World Wide Web this was a good opportunity to administer this survey. Furthermore, the participants were already familiar with Google Form surveys because the school used them regularly to obtain students' opinions. Participants were given five minutes to read the question paper before the start of the exam, something that is common in International Baccalaureate examinations, and were then asked to complete the self-confidence questionnaire. The timing of the questionnaire was crucial because the first three questions related to students' confidence in solving the test problems. There was some consideration regarding the ethics of asking students to complete a questionnaire when they may be anxious about the forthcoming test. However, since the purpose of the questionnaire was to assess anxiety about completing programming tasks, this was only the appropriate time. Students were told that they could opt out of the research project prior to the questionnaire being distributed but were reminded that they would still be required to take the test. Since the questionnaire was online they could simply have decided not to submit it. The data collected from the CPCA Questionnaire (Appendix L) was purely quantitative and was imported into the software, IBM SPSS v.22.1 for Mac computers.

3.4.2.2 The Student Perceptions of Computer Science (SPCS) Questionnaire

The SPCS Questionnaire was not based on a previously created questionnaire and the questions were developed to respond to areas that were identified in the literature review. The various strategies for learning to program a computer were developed to overcome learning challenges, improve enjoyment, and overcome negative perceptions. For example, a number of research papers discussed computer programming difficulty (e.g. Başer, 2013; Mow, 2008), enjoyment (e.g. Armoni et al., 2015; Rowell et al., 2003), and the perceptions of computer programming (e.g. Anastasiadou & Karakos, 2011; Carter, 2006; Faidi, Freihofer, and Townsend, 2017), so Likert questions were created to respond to these identified themes. Additional research had identified Computer Science as a masculinised discipline (e.g. OECD, 2015), so it was important to identify if middle school girls viewed this subject as a potential career. The final area of investigation related to interest in computer gaming, since some research (e.g. Buffum et al., 2005) had expressed concern over girls' prior experience of computer gaming and their progress in learning computer programming.

Prior to the distribution of the SPCS Questionnaire, a pilot group was assembled to validate the questions. The pilot group consisted of 19 students (10% of the participants) who were selected using a random number generator from the girls and boys in Year 8 and Year 9. The sample students were asked to complete the online Google Form questionnaire privately but were informed that if there was any ambiguity they should seek clarification from the supervisor. Despite the fact that five students left, at least, one open-ended question blank, no changes were made to the questionnaire after the sample was reviewed. The sample students were not asked to complete the questionnaire again. When the SPCS Questionnaire was distributed to the remaining participants their responses were added to the original 19 responses.

The SPCS Questionnaire (Appendix M) was constructed with 11 questions, 8 Likert-style questions (with a scale of 1 to 7), and 3 open-ended questions, which Murthy (2008) suggests may produce richer responses that can be encoded directly into qualitative data analysis packages such as Nvivo. The open-ended responses were included in the hope of catching unexpected information that otherwise might not have been considered. Further, it places ownership of the data into the respondents' hands (Cohen et al., 2011). These areas of investigation were developed from the review of the literature. Information was collected on the following areas:

- Year group and gender
- Difficulty and enjoyment level of computer programming class compared to other subjects

- Desire to continue studying computer programming and whether they believe it is a useful skill to learn
- Frequency, usefulness, and how enjoyable they find computer games
- Skills/attributes most useful for computer programming

This questionnaire was distributed to participants towards the end of the academic year and approximately two weeks after the programming test had been administered. As with the previous questionnaire, the SPCS Questionnaire was also administered as an online Google Form survey. The quantitative data collected from this questionnaire was imported into IBM SPSS v.22.1 for analysis and the qualitative data collected from the open-ended questions was imported into Nvivo v.11 for analysis.

3.4.3 Data collection - Research Question 3 ('Are there any extrinsic factors that affect girls choosing to study computer programming in Year 10?')

A review of the literature suggested that there might be a number of extrinsic influences affecting girls choosing a career in Computer Science. These included the following: a lack of role models; friends or older students; parental pressure or support; and interactions with teachers. To determine the extent of the potential external influences, group interviews were conducted with a sample and an online questionnaire was distributed to all participants.

3.4.3.1 Influences on IGCSE Choices (IIC) Questionnaire

To address the third research question *('Are there any extrinsic factors that affect girls choosing to study computer programming in year 10?')*, an additional questionnaire was developed. The literature that focused on the factors that influence students in their subject selections or career choice was reviewed (e.g. Chope, 2005; Malgwi, Howe, & Burnaby, 2005) and the questions were developed from the influences that emerged from the literature.

This IIC Questionnaire was administered as an online Google Form survey, but prior to distribution a pilot group was assembled to validate the questions. The pilot group consisted of 19 students (10% of the participants). They were asked to complete the questionnaire privately but were informed that if there was any ambiguity they should seek clarification from the teacher supervisor. This pilot group was selected randomly from all of the classes in the current study so there were male and female participants from both year groups. As a further way of validating the questionnaire, the final question, “Were there any other influencing factors and if so what were they?”, was included to see if any influences had been overlooked. The responses to the final open-ended question included subject difficulty as an additional influencing factor, so an additional Likert question was added on to the questionnaire. No other revisions were required.

The IIC Questionnaire (Appendix N) was made up of 11 questions, 9 Likert-style questions (using a scale of 1 to 7) and 2 open-ended questions. The scale of 1 to 7 was consistent with other two questionnaires (CPCA and SPSS) to avoid confusion for the participants. The questionnaire contained relatively few questions to encourage participants to answer all of them. Additionally, the questionnaire was set to anonymous responses so that the students would feel comfortable expressing their opinions. Information was collected on the following areas:

- Year group and gender
- The chosen IGCSE subjects
- Social influences including: parents, teachers, friends, role models
- Other influences: necessary life skills, university/career, enjoyment, difficulty

The questionnaire was distributed electronically to all of the participants, including the pilot group, in December 2015, some six months after the other questionnaires and a day after the pilot group. This date coincided with the initial phase of choosing IGCSE options for Year 9. Since Year 10s had started their IGCSE subjects in September, their final selection process was completed six months previously. Questionnaire data were collected from all Year 9 and Year 10 students. Since these students were the Year 8 and Year 9 responders in the previous questionnaires, obtaining their responses provided continuity. Additionally, many of the former Year 9 students had only finalised their classes a few months previously and many had switched classes with the first month of their IGCSEs.

The quantitative data collected from this questionnaire was imported into SPSS v.22.1 for analysis and the qualitative data collected from the open-ended questions was imported into Nvivo v.11 for analysis.

3.4.4 Data collection – Research Question 4 ('Is there a significant gender difference in the preferred learning styles for computer programming?')

To address the final research question, a combination of a questionnaire and group interviews was employed.

3.4.4.1 Preferred Computer Programming Learning Style (PCPLS) Questionnaire

The researcher constructed this questionnaire in order to assess the different strategies employed throughout the year. The questions were developed from the literature review that had informed the teaching strategies. Like the previous questionnaires, an anonymous responses setting was enabled so that the students would feel comfortable expressing their opinions.

Prior to distribution of the PCPLS Questionnaire, a pilot group was assembled to validate the questions. The pilot group consisted of 19 students (10% of the participants) who were randomly selected. They were asked to complete the questionnaire privately but were informed that if there was any ambiguity they should seek clarification from the teacher supervisor. The responses from the sample were reviewed to see if any revisions were necessary but apart from one typo none were required.

The PCPLS Questionnaire was distributed in June 2015, the week after the programming test. The questionnaire (Appendix O) contains 12 questions, of which 10 are Likert-style questions (with a scale of 1 to 7) and 2 are open-ended questions. The number of questions was kept to a minimum, in order to encourage completion. Information was collected on the following areas:

- Year group and gender
- Preferred learning style based on the methods of teaching programming

The quantitative data collected from this questionnaire was imported into SPSS v.22.1 for analysis and the qualitative data collected from the two open-ended questions was imported into Nvivo v.11 to support the analysis.

3.4.4.2 Group interviews

Woodside (2010) believes that using closed-ended responses, such as those in Likert-style questions, fails to uncover the deep nuances between thoughts and actions. He comments that interviews can often uncover deeper level understanding on the thinking/doing processes. An additional rationale for using interviews is the acknowledgement that data gathered through testing and questionnaires relate to human subjects and are not external to individuals (Kvale, 1996). Thus, to understand the thinking of the participants they need to be treated like people and not simply as data.

Several types of interviews were considered for this study, including structured, semi-structured, unstructured, informal, and focus groups. Whilst an unstructured interview means that the interviewer can adapt their questions based on the interviewee's responses (McNiff, 2016), this does require an experienced interviewer.

Children's interviews may be subject to response or social desirability bias (Bryman, 2012), where interviewees respond according to what the interviewer wants. In South Korea, students' respect for teachers is relatively stronger than those in Western countries (Shin, Lee, & Kim, 2009) and this may make students reluctant to express negative opinions about their teacher's subject. Since the researcher is the students' subject teacher, any interviews carried out by him might have produced biased data. Thus, a boarding master, who did not teach the students, conducted the interviews to ensure that the students were free to express opinions without fear of reprisals. This decision ultimately impacted on the decision to use semi-structured interview questions (Appendix P). Student opinions were recorded using the recording app on a school-owned smartphone. The recording was kept secure and deleted once it has been transcribed by the interviewer.

The use of sound recording equipment during interviews was only used with the written permission of the participant. A total of 13 (2 boys and 11 girls) year 9 students, chosen at random from those who volunteered, were asked to take part in a 20-minute semi-structured group interview. The rationale for the selection of interviewed students was outlined in the sampling section. The interview questions (Appendix P) were formed from the results of the questionnaires and focused on two areas: 1) what influenced the students IGCSE subject choices; 2) how they perceived Computer Science and whether gender has any influence on the decision to study Computer Science.

3.5 Data Analysis

Data analysis takes into consideration quantitative data and qualitative data obtained at different stages in the timeline. Where quantitative data has been analysed statistically, a combination of parametric and non-parametric tests have been used. Parametric tests are statistical tests that assume the data approximates a normal distribution and also that the variance of the data is uniform either between groups or across the range being studied (Altman & Bland, 2009). Nonparametric tests are a class of statistical procedures that do not rely on assumptions about the shape of the probability distribution.

Statistical power is the likelihood that an effect will be found when there is an effect to be found (Rossi, 2013). If statistical power is high, the probability of concluding that there is no effect when one is there to be detected goes down (Finder et al., 2014). Parametric tests such as ANOVA offer greater statistical power than nonparametric tests provided that the assumptions underlying them (e.g. normality and homogeneity of variance) are satisfied (Erceg-Hurn, & Mirosevich, 2008). There are several tests that can be used to check for homogeneity of variance but since Levene's test of homogeneity is the most commonly used one in the literature (Garson, 2012), it was selected for this study. Failure to test for underlying assumptions when using parametric tests may increase the likelihood of a Type I, or concluding that a real effect exists when there is not one (Hothorn, Bretz, & Westfall, 2008). The choice of statistical tests in this study is based on the statistical power of the tests and is discussed in each section.

The data analysis is structured to answer the four research questions and is arranged accordingly.

3.5.1 Data analysis - Research Question 1 ('Is there a significant gender difference in computer programming ability?')

To address the first research question, a programming test was administered to the students. Data analysis for this question was conducted by looking at three questions relating to ability:

- Was there a significant difference between the overall scores of the girls and the boys?

- Was there a gender difference in the number of participants who score 90% or above?
- Was there a gender difference between the number of girls and boys that successfully complete the extension task, ‘The Rainfall Problem’?

The data collected from the programming test (Appendix K) was purely quantitative and consisted of an independent variable of gender and a dependent variable of test score. To determine if there was a gender difference in programming performance, a comparison of each gender’s test scores was made using the parametric *Independent Analysis of Covariance (ANCOVA)*. The value of such a test is that by including a co-variant, in this case, IQ Scores, it is possible to ensure that any reported gender differences were not driven by *a priori* differences in intelligence. Further, they allow inferences about a population to be made from a sample, as long as the sample is representative of the population and the data does not violate the assumptions of parametric tests. These assumptions are outlined below:

1) random, independent sampling from the populations;
2) normal population distributions;
3) equal variances within the populations.

Point 1 is met because the students in the study are representative of secondary school students in South Korea. Points 2 and 3 can be relaxed when large sample sizes are used (Wilcox, 2005) but if this instance the group sizes are unequal and, depending on the data variance and sample size, this may bias the F-ratio to be conservative.

An ANCOVA can be considered robust if the group sizes are not very different in size (Wilcox, 2005), despite Keppel and Wickens (2004) suggesting that it is difficult to know when unequal sample sizes make heterogeneity of variance an issue. To ensure that the size of the groups was as large as possible, year group was discounted and only gender considered as the independent variable. Given that the population sizes for gender are not very different and are quite large, it was reasonable to assume normal distributions due to the central limit theorem. However, a test for skewness and kurtosis, prior to running the ANCOVA, confirmed that Kurtosis z-value might be an issue with the boys’ test scores. To assess equal variances within a population, Levene’s test of homogeneity of variances was conducted to assess the equality of variances between each gender’s programming test scores.

The analysis of the test scores took the form of a one-way Independent Analysis of Covariance (ANCOVA), using IQ scores as the covariate, gender as the independent variable, and test score as the dependent variable. Where an independent variable has more than 2 levels/conditions and where a significant main effect is observed, then you must carry out further tests of pairwise differences. However, in this instance, there were only 2 levels/conditions so a post hoc analysis was not run. As a precaution against the violation of homogeneity of variances, where Levene's test was significant, the Mann-Whitney test (Mann & Whitney, 1947) was conducted because a non-parametric test is recommended when the distribution variances are dissimilar (Nachar, 2008).

ANCOVA has greater statistical power over simple Analysis of Variance (ANOVA) because of a reduction in error variance and an increase in the precision of estimates (Frigon & Laurencelle, 1993). Therefore, using students' entrance IQ scores as a covariate when analysing programming ability controls for IQ because without this it is possible that any between-group differences were not due to gender but rather to the fact that the girls happened to differ from the boys in IQ. The ANCOVA was run using the software IBM SPSS v.22.1 for Mac computers to determine if there was a significant difference between the overall scores of the girls and the boys.

Additional analysis was conducted to see if there is any gender difference at the top end of scores. Thus, the total number of students who scored 90% and over in the test were counted and compared by gender using a Chi-square test of independence. Further, the frequency of participants who completed 'The Rainfall Problem' was calculated by gender using a Chi-squared test of independence. Since the rainfall problem (Soloway et al., 1983) is an established test for undergraduates it was deemed to be a suitable test for assessing the most-able middle school students. Both of these tests were calculated using Microsoft Excel for Mac 2011.

3.5.2 Data Analysis – Research Question 2 ('Are there any intrinsic factors that affect girls choosing to study computer programming in Year 10?')

The mean, standard deviation for both girls and boys was calculated from the 7-point Likert questions (1 - strongly agree to 7 - strongly disagree). The Computer Programming Confidence and Anxiety (CPCA) Questionnaire (Appendix L) contained several questions

(Items 2, 4, 6, 7, 8, 9, and 10) specifically relating to student confidence on various aspects of programming and two questions (Items 3 and 5) on anxiety.

As a general approach to all testing of the CPCA Questionnaire and the SPCS Questionnaires, the Likert responses from the Likert questions were averaged, skewness *z* value and kurtosis *z* value calculated, Levene's test of homogeneity of variances was conducted, before one-way ANOVA was carried out on the composite confidence and composite anxiety measures using SPSS v.22.1. Where assumptions of normality or homogeneity of variances were violated a Mann-Whitney U test was also conducted using SPSS v.22.1.

In order to assess the confidence of completing programming tasks immediately prior to attempting them, three of the questions (Items 8, 9, and 10) on the CPCA Questionnaire (Appendix L) that asked the participants to rate their confidence to successfully complete each section of the computer programming test were averaged. The general approach, outlined above, to testing the Likert responses was conducted.

To assess confidence in their ability to study Computer Science at undergraduate level the responses from question 7 on the CPCA Questionnaire was analysed using the general approach outlined above.

To ascertain the students' perception of which gender was more likely to do well in the computer programming assessment and which gender is the more able academically, questions 11 ('*Do you think that there is any difference in academic abilities between boys and girls in your year group?)*' and 12 ('*Do you think there will be any difference between boys and girls in the results of this assessment?*') on the CPCA Questionnaire were analysed using the general approach outlined above.

To establish whether students had a stereotypical view of which gender should work in Computer Science, question 3 ('*Do you think boys or girls are more suited to a career in computer programming?)* on the Student Perceptions of Computer Science (SPCS) Questionnaire were analysed using the general approach outlined above.

To determine whether there was a difference between girls’ and boys’ enjoyment of computer programming, question 1 (‘*On average, and compared to your other classes, how enjoyable is computer science?)* on the SPCS Questionnaire and question 1 (‘*The challenge of learning computer programming is exciting*’) on the CPCA Questionnaire were analysed separately using the general approached outlined above.

There were three open-ended questions on the SPCS Questionnaire (Appendix M) and so the open-ended questions 4, 5, and 7 were imported into nVivo v.11 for keyword comparisons. These were analysed in combination with the group interviews in the qualitative chapter.

To assess whether the interest in computer gaming was different between boys and girls the responses from question 8 (‘*How much do you enjoy playing computer games*’) on the SPCS Questionnaire were analysed using the general approach outlined above.

To develop an understanding of whether there was any difference in the opinions of boys and girls regarding the usefulness of computer gaming the responses from question 9 (‘*In general, do you think computer games are a useful activity or a distraction from doing useful things?’)* on the SPCS Questionnaire were analysed using the general approach outlined above. Further, the responses of girls’ and boys’ identifying the amount of time spent playing computer games and the type of game played by boys and girls were totalled and displayed using bar charts.

To establish whether the students intended to student computer programming in the future, question 6 (‘*Do you think you will choose to study computer programming in the future?’*) on the SPCS Questionnaire (Appendix M) was analysed using the general approach outlined above.

To ascertain whether there was any difference in the perception of computer programming difficult the responses from question 1 (‘*On average, and compared to your other classes, how difficult/easy is computer science?’)* on the SPCS Questionnaire were analysed using the general approach outlined above.

3.5.3 Data analysis - Research Question 3 ('Are there any extrinsic factors that affect girls choosing to study computer programming in Year 10?')

Data from the Influences on IGCSE Choices (IIC) Questionnaire (Appendix N) focused predominantly on external influences and so the means, standard deviations, skewness, and kurtosis for questions 2 to 9 inclusive were examined to see if there was any gender difference and to rank the external influences. Students were asked to rank from 1 – strongly influence to 7 – no influence, inclusively, how much these factors (parents, teachers, friends, useful life skills, career/university plans, lesson enjoyment, subject difficulty, and role models) influenced their IGCSE choices. Levene's test of homogeneity of variances was carried out and a one-way ANOVA was run. Where there were violations of homogeneity or normality on any of the factors a Mann-Whitney U-test was also run. Additionally, there were two open-ended questions and so the responses from these questions (Items 10 and 11) were imported into nVivo v.11 for keyword comparisons.

3.5.3.1 Qualitative data

Group interviews were carried out at the end of term three, and after the programming test, with the sample of Year 9 students. The purpose of these interviews was to discuss factors that might influence students' IGCSE options and to delve more deeply into perceptions of Computer Science.

Interviews produce copious amounts of valuable and muddled data for analysis (Banister, 2011). Therefore, thematic coding, a technique whereby meaning is established from a patterned response within the dataset (Braun & Clarke, 2006), was used following Lichtman's (2012) six-stage coding methodology in order to make analysing qualitative data more manageable. This process went as follows:

Step 1 Initial coding.

This involved a careful reading of the narrative and the adding of a word or two, and circling key phrases (Appendix Q). At this stage, the following patterns were considered: conversation topics, vocabulary, recurring activities, meanings, feelings, or folk sayings and proverbs.

Step 2 Revisiting the initial coding.

This involved collapsing or renaming some of the codes, whilst still paying consideration to the patterns suggested in step 1.

Step 3 Initial listing of categories.

At this point in the process, some of the codes were organised as subsets under a major topic (Appendix R).

Step 4 Modifying the initial list.

As recommended by Lichtman (2012), the list was examined, with a reminder that the goal is to move from the initial data through identification of categories to the recognition of important concepts or themes.

Step 5 Revisiting categories.

This is the part of the iterative process where the codes have become categories and incorporated a final organisation of categories.

Step 6 Categories to concepts.

After repeating the iterative process of reading and refining, the final stage was to move from categories to concepts with the purpose of providing meaning to what otherwise might be considered meaningless when viewed alone (Leininger, 1985).

Thus, the six-stage iterative process was completed and a handful of well-developed concepts were then produced (Appendix S). At this point, a fellow doctoral student was asked to moderate a portion of the coding and confirmed that the themes were reasonable. This process called inter-coder reliability is used to provide a degree of validity, and is widely accepted in the academic community (Kolbe & Burnett, 1991; Lombard, Snyder, Duch, & Bracken, 2002; Tinsley & Weiss, 2000). The challenge of using inter-coder reliability with more than one person is that the different coding patterns can cause a substantial bias in the resulting data (Burla et al., 2008). Further, if that person looking at the themes is not familiar with the research then their judgement may not be reliable. For these reasons, the second coder was asked only to moderate and not asked to produce their own themes.

Once the themes and sub-themes, which are like an indexed filing system to lead back to the original passage (Richards, 2009), were established, the transcripts were input into the software application, Nvivo v.11, where assigning labels to sections of information allowed for cross-case comparison and word tree analysis (Appendix T).

3.5.4 Data analysis - Research Question 4 ('Is there a significant gender difference in the preferred learning styles?')

Data from questions 1 through to 10, inclusively, from the PCPLS Questionnaire (Appendix O) was analysed using Levene's homogeneity of variances. Additionally, skewness and kurtosis were calculated to assess for normality. Following this a one-way independent ANOVA was run but where violations of homogeneity of variances or normality were revealed a Mann-Whitney U-test was also run.

Questions 11 and 12 were open-ended questions and the student responses were imported into nVivo v.11 so that keywords could be searched. There was a preliminary attempt to discuss research question 4 (*'Is there a significant gender difference in the preferred learning styles for computer programming?'*) with a group of 10 girls to see if it was worthwhile pursuing this line of inquiry, but within the first few minutes, it was apparent that it was not. Moreover, it was felt that the open-ended questions on the PCPLS Questionnaire (Appendix O) provided sufficient detail to answer research question 4.

3.6 Ethical considerations

Throughout the process of planning, researching and writing this thesis proposal, rigorous attention was paid to the ethics procedures. The University of Reading has clear guidelines for ethical approval and all steps were adhered to in obtaining ethical approval for the Ethics Committee Board. The ethics documentation is composed of six components: an ethics form; a risk assessment; student consent form; research assistant consent form; parental consent form; and a school consent form. The University of Reading's Ethical Committee gave ethical approval on 05/04/2015 (Appendix B).

Initially, a meeting regarding conducting research in the school was held with the school Principal and his consent was gained. Following on from this, it was necessary to obtain the support of a research assistant who would oversee the administering of the test. The research assistant was the other Computer Science teacher so asking him to run the test was necessary because some of the tests were run simultaneously in different rooms. Since all students were off timetable during the assessment week it was normal for teachers to run their own tests. An additional person, a boarding master, was asked to conduct the group interviews (Appendix B). This person was chosen because they had a personal relationship with the students and did not teach one of the subject options so the students could speak candidly about their opinions. Once the administration logistics had been organised, the parents needed to understand the purpose of the research and how it would affect their children. This was conducted through a group e-mail containing an informative leaflet with a consent form as an attachment (Appendix B).

The British Education Research Association ethical guidelines (BERA, 2011) identify that research should be open and honest unless subterfuge can be fully justified. Therefore, in each Computer Science class, the students were informed that purpose of this research is to determine what influences affect girls choosing to continue studying Computer Science, and therefore computer programming, and also to look at different teaching strategies to improve engagement with the subject. They were also, truthfully, informed that boys' opinions and preferences were just as important as the girls'. These data were necessary to enable gender comparisons and also so that positive teaching adjustments could be made to the ways both genders learn computer programming.

The students were then given a simplified, and more colourful, version of the informative leaflet (Appendix B) and asked to discuss with their parents. A follow-up e-mail was sent to the parents of all participants prior to data collection reminding them of the research purpose and their right to withdraw their children at any time.

In accordance with the University of Reading's ethical guidelines for research, every consideration was made to ensure that no harm or duress was inflicted on any child. Firstly, the research was explained with sufficient detail and accuracy (Alderson & Morrow, 2011), so that the participants could give informed consent or refusal. Secondly, it was important to try and develop a separate research relationship to the pre-existing teacher relationship by clearly explaining to the students when data was collected for research purposes and when it was collected for school-based assessment. Finally, and as suggested by Harcourt (2011), participants were reminded of their right to withdraw from the study at every research data collection point.

Young children have the capacity to engage in research as thinkers and communicators and the inclusion of children's opinions is crucial for a more complete picture of their life worlds (Harcourt & Einarsdottir, 2011). Yet, listening to children and recording their activities and views can be intrusive, and could be a way for the adult to influence children. Eide and

Winger (2005) question whether adults have the right to search for children's points of view; whereas Bromstom (2006) suggests that conducting research with children is a balancing act between children's rights to protection, rights to participation, and rights to privacy. The purpose of this research is to provide girls with equal opportunity to study Computer Science at IGCSE by discovering the reasons why they might choose not to do so. The right to participate in Computer Science justifies searching for children's point of view.

Privacy, which Alderson and Morrow (2011, p. 31) define as "avoiding undue intrusion into their personal affairs" is an important ethical concern. The collection of personal data needs consideration of not only the interviewee but also those that they discuss. Changing the student names provides some safeguards, and since children have the same rights to confidentiality as adults (Brazier & Cave, 2016), confidentiality was maintained by keeping data secure through the use of passwords.

In medical research and business, it is common for incentives to be offered in an effort to influence participant involvement in market research. However, the use of incentives was not used in this research and the students were informed that agreeing to participate would not, in any way, grant them favour or advantage over those who did not. With this neutrality, all students would be given the opportunity to participate, which could be of educational and emotional benefit to the children themselves (Sikes, 2004).

3.7 Summary

The ontological position of the researcher leans towards that of historical realism, with this research following a post-positivism critical theory paradigm. Methodologically, a mixed methods approach was followed in that both qualitative and quantitative data collection were utilised, although the majority of the data were quantitative. The methods used to collect data include a programming test, four online questionnaires and a small sample of group interviews. Data analysis was conducted through statistical analysis in the form of Mann-Whitney, ANCOVA, ANOVA, Chi-square, and through the six-stage coding methodology in order to analyse differences between a total of 194 female and male novice programmers, in Year 8 and Year 9, and assess the effectiveness of interventions employed to create a positive learning environment. With the theoretical perspective, methodology, methods, sampling, data collection, data analysis and teaching strategies described, the findings are presented in the next chapter.

Chapter 4 Quantitative Findings

4.1 Introduction

This chapter reports the results from the programming ability assessment by analysing gender differences in overall test performance, test performance above 90%, and those students that attempted the extension task. Following on from this, there is an exploration of the questionnaires' closed-questions to report the intrinsic influences on IGCSE choices, extrinsic influences on IGCSE choices, and preferred computer programming learning style. Throughout all of the quantitative data analysis, parametric testing was used but where assumptions (i.e. skewness, kurtosis, and homogeneity of variances) were violated, non-parametric testing was also conducted to verify the results of the parametric tests.

4.2 Quantitative Results - Research Question 1 ('Is there a significant gender difference in computer programming ability?')

To address the first research question, a programming test was administered in June 2015, with 194 students (81 girls and 113 boys) taking the test in order to ascertain whether there was any gender difference in the students' ability to write HTML5/CSS syntax to create a web page, write programming solutions in Python Turtle, and write programming solutions based on flowcharts (Appendix B). The participants were given 75 minutes to complete all three tasks and they were allowed to attempt them in any order.

The reliability of annual test results can come into question because the academic ability of a particular cohort of students can also vary, especially if you accept that students themselves contribute up to 50% of the variability of their performance (Hattie, 2013). The selective school in this study requires all applicants to take standardised entrance examinations. One of the assessments is the widely used 'Wechsler Intelligence Scale for Children' (Wechsler, 2003), and the school Principal gave permission for this data to be accessed. Intelligence quotient (IQ) is a standardised measure of human intellectual capacity that takes into account an extensive range of cognitive skills (McCall, 1977). Therefore, it is possible to use the participants' IQ scores as a way of ensuring that any apparent effects of gender were not driven by *a priori* differences in intelligence, and so an analysis of covariance (ANCOVA) was run with IQ as a covariant variable.

In order to assess students at the high end of computer programming ability, any student who completed all of the activities within the allotted time was asked to attempt an extension activity. This test of programming ability has been used with undergraduates and is called the 'Rainfall Problem' (Appendix B).

4.2.1 Gender difference in overall programming ability

A number of exploratory statistical measures, including mean; standard deviation; kurtosis; and skewness was carried out on the participants' IQ and programming test scores. To determine if the population groups demonstrate normal univariate distribution and are, therefore, suitable for an ANCOVA test skewness and kurtosis were tested. One method to achieve this is to check that the *z* values for skew and kurtosis lie between -2 and +2 (George & Mallery, 2010) and, as can be seen in Table 4.1, the data screening revealed that the data did deviate too far from normality with the boys' test score. Figure 4.1 shows graphically that the boys' test scores fall outside of a normal distribution shape slightly.

Table 4.1

Girls' and Boys' Intelligent Quotient (IQ) Scores and Test Scores

	girls' IQ	girls' test score	boys' IQ	boys' test score
Mean	124.38	26.19 (67%)	121.45	22.50 (58%)
Std. Deviation	10.46	8.75	10.198	9.94
Skewness *z* value	-0.27	-1.03	-1.44	0.73
Kurtosis *z* value	-0.63	-1.38	-0.55	-2.71

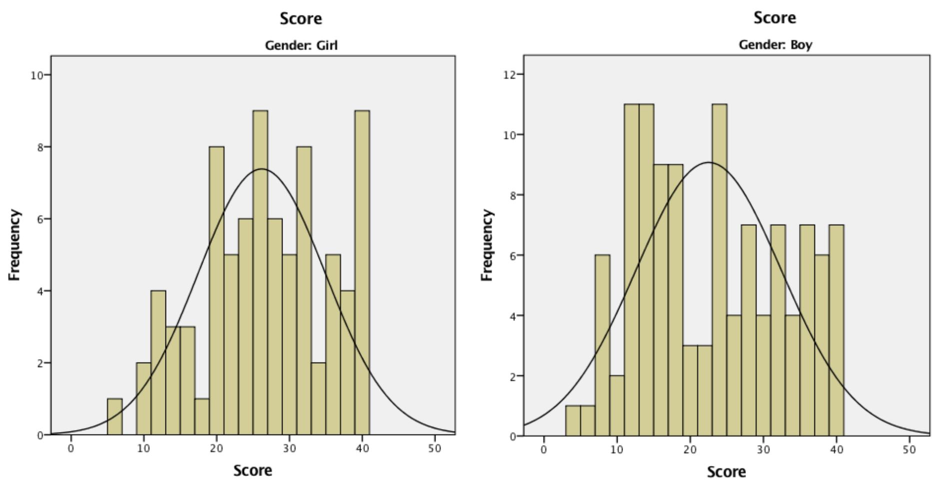


Figure 4.1 Chart showing Girls' and Boys' test scores, relative to a normal distribution

Levene's test of homogeneity of variances was conducted to assess the equality of variances between each gender's programming test scores, and produced a significant result, ***F* (1,192) = 4.35, *p* < .05**, therefore indicating that assumptions relating to homogeneity of variances were violated and that results of parametric testing may not be entirely reliable. Thus, after the parametric test results were obtained, a non-parametric test was also tested to ensure the validity of the results of the parametric test.

The data obtained from the students' test scores were analysed using a one-way analysis of covariance (ANCOVA), using a baseline covariate of IQ and gender as the independent variable. There was a significant difference in participants' scores, ***F* (1,191) = 5.54, *MSE* = 87.57, *p* < .05**. IQ was a significant covariate, ***F* (1,191) = 5.28, *MSE* = 87.57, *p* < .05**.

The results from the non-parametric, Mann-Whitney U-test indicated that programming raw test scores were significantly greater for girls (**Mdn = 26.0**) than for boys (**Mdn = 23.0**), ***U* = 3581.5, *p* = .01**. Since both tests revealed that there was a significant difference in test performance in favour of girls, the findings from the ANCOVA can be accepted, and suggest that the girls' performance was indeed superior to boys', over and above the difference in intelligence score.

4.2.2 Gender differences in programming ability in the most able programmers

To determine if there was a gender difference in students capable of achieving high marks, set at 90% or above, a Chi-square test of independence was conducted. The result was not significant χ^2 **(1, N = 194) = 0.35, *p* > .05** indicating that there is no gender difference in the relative proportions of boys and girls who scored 90% or more, with 14% (16 from 113) of boys and 17% (14 from 81) girls achieving this result. To provide a visual description of computer programming performance by gender, a histogram showing the frequency of boys' scores' and girls' scores within blocks of 10% is shown in Figure 4.2.

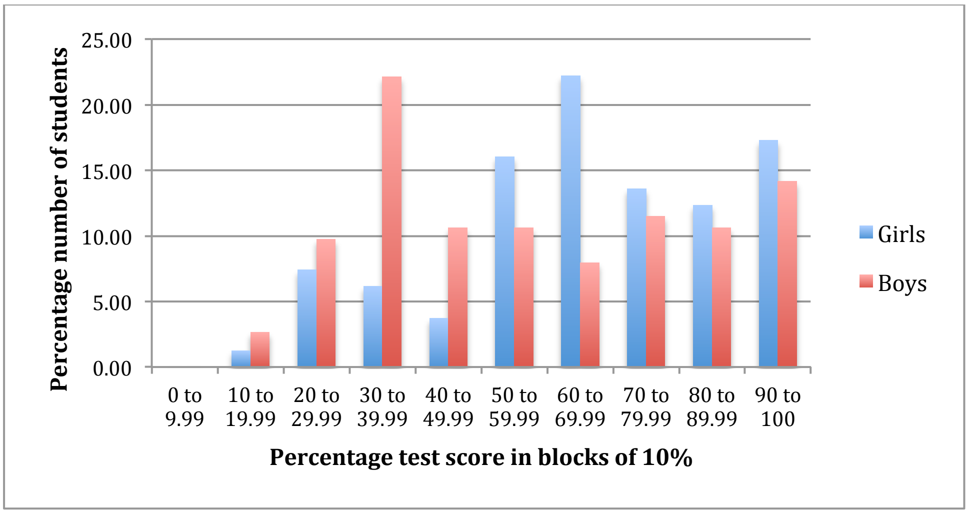


Figure 4.2 Chart showing Girls' and Boys' test scores segmented into groups of 10%

4.2.3 Gender differences in programming ability in the students that successfully complete the extension task, 'The Rainfall Problem'

There were several students who have excellent programming skills relative to their age group and experience. The 'Rainfall Problem' extension activity could only be attempted by those high-achieving participants who had completed all of the tasks well within the timeframe of 75 minutes. There were 16 students (8% of the cohort), who attempted the 'Rainfall Problem' and 11 of these 16 students successfully completed the activity. Of the entire sample, there were 6 of the 113 boys (5%) and 5 of the 81 girls (6%) that successfully solved the problem.

A Chi-square test of independence was calculated comparing the frequency of participants who completed 'The Rainfall Problem' by gender. Results showed that there was no gender difference in the rates of students who completed the extension activity χ^2 **(1, N = 194) = 0.07,** $p > .05$.

4.3 Quantitative Results - Research Question 2 (*'Are there any intrinsic factors that affect girls choosing to study computer programming in Year 10?'*)

This section of the quantitative findings is composed of the results summary from the Computer Programming Confidence and Anxiety Questionnaire (Appendix L) on students' computer programming confidence, students' computer programming anxiety, and students' gender stereotypes.

4.3.1 Computer Programming Confidence and Anxiety (CPCA) Questionnaire Findings

The CPCA Questionnaire (Appendix L) was administered to the students electronically to investigate how confident they were in aspects of computer programming, to determine their level of anxiety, and to see if they had any stereotypical beliefs about gender differences in computer programming ability (Questions 11 and 12). Of the 194 students, in Year 8 and Year 9, that completed the questionnaire, there were 81 girls and 113 boys. The first 10 questions and the mean (standard deviation) responses from the 7-point Likert scale are shown in Table 4.2. A mean score below 4 signified agreement; a mean score above 4 signified disagreement; and a mean score of 4 signified that the students as a collective neither agree nor disagree.

Table 4.2

Girls' and boys' mean (and SD) responses from the Computer Programming Confidence and Anxiety Questionnaire (Scores were on 7 point Likert scales, from 1-"strongly agree" to 7-"strongly disagree").

Questions	girls' data	boys' data
1. The challenge of learning computer programming is exciting.	*M*=2.81 *SD*=1.52	*M*=2.47 *SD*=1.19
2. I am confident that I can learn to write computer programs to solve the kinds of problems we encounter in Computer Science classes.	*M*=3.38 *SD*=1.56	*M*=3.04 *SD*=1.44
3. I get frustrated/fed up when the computer program I have written doesn't work	*M*=5.51 *SD*=0.99	*M*=5.40 *SD*=1.05
4. I am confident that I can find and correct errors in my computer programs.	*M*=3.17 *SD*=1.30	*M*=3.31 *SD*=1.35
5. I do not like to make changes to a working computer program in case it goes wrong and stops working.	*M*=5.28 *SD*=0.83	*M*=5.27 *SD*=0.85
6. I am confident that when I have a programming task, I can work independently by using the Internet to help me find a solution.	*M*=2.88 *SD*=1.55	*M*=3.05 *SD*=1.41
7. I think that, in the future, I would have the capability to successfully complete a University degree that includes modules in computer programming *(Note: This is not asking if you want to study programming at University, only if you think you have the ability).*	*M*=3.75 *SD*=1.84	*M*=3.35 *SD*=1.30

8. I will be able to correctly complete the HTML / CSS task during the exam *(Note: Look at the HTML / CSS question and choose the appropriate answer below).*	*M*=3.62 *SD*=1.51	*M*=3.41 *SD*=1.26
9. I will be able to correctly complete the Python Turtle tasks during the exam *(Note: Look at the Python Turtle questions and choose the appropriate answer below).*	*M*=3.04 *SD*=1.28	*M*=2.91 *SD*=1.34
10. I will be able to correctly complete the Python Flowchart tasks during the exam *(Note: Look at the Python Turtle questions and choose the appropriate answer below).*	*M*=3.58 *SD*=1.63	*M*=3.45 *SD*=1.31

The CPCA Questionnaire (Appendix D) was designed to consider confidence and anxiety in learning computer programming. There were a number of questions relating to computer programming confidence (Items 2, 4, 6, 7, 8, 9, and 10) and so these were collated to form a composite programming confidence score and then analysed using an ANOVA. Additionally, there were two questions relating to computer programming anxiety (Items 3 and 5) and so these were collated to form a composite programming anxiety score and then analysed using an ANOVA. The full questions for the items categorised by confidence and by anxiety can be seen in table 4.2.

4.3.1.1 One-way ANOVA on programming confidence and anxiety

The CPCA Questionnaire (Appendix L) contained several questions (Items 2, 4, 6, 7, 8, 9, and 10) designed to assess student confidence on various aspects of programming. Figure 4.3 shows the mean differences between the girls' and boys' responses to these questions.

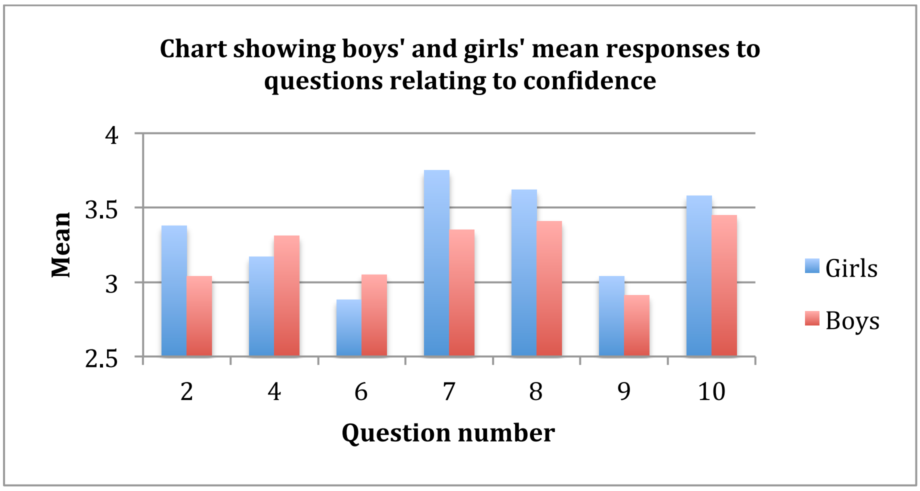


Figure 4.3 Chart showing mean scores of confidence by gender (scale of 1 – 7) with the higher the number the less the confidence

The Likert responses to the seven questions relating to computer programming confidence (Items 2, 4, 6, 7, 8, 9, and 10) were averaged to provide a composite confidence measure. The girls' mean and standard deviation for this composite confidence measure was **3.36 (1.25)** and the boys' mean and standard deviation was **3.22 (1.05)**. For girls' responses, the skewness z value was 2.63 and the kurtosis z value was 1.23. For boys' responses, the skewness z value was 2.28 and the kurtosis z value was 1.32. Additionally, Levene's test of homogeneity of variances was not significant $F(1,192) = 1.48, p > .05$. A one-way ANOVA was conducted on participants' scores and revealed that the difference between boys' and girls' self-reported confidence in computer programming was not statistically significant $F(1,192) = 0.282, p > .05$. Since an assumption of normality was violated, a non-parametric Mann-Whitney U-test was run. The results were girls (**Mdn = 3.0**), boys (**Mdn = 3.14**), $U = 4369, p = .59$ and this aligns to the one-way ANOVA findings that girls' and boys' self-reported confidence in computer programming were not significantly different.

The Likert responses to the two questions relating to computer programming anxiety (Items 3 and 5) were averaged to provide a composite confidence measure. The girls' mean and standard deviation for this composite anxiety measure was **5.39 (0.73)** and the boys' mean and standard deviation was **5.33 (0.78)**. Figure 4.4 shows the mean differences between the genders on these questions. The skewness z value for girls' responses was .57 and the kurtosis z value was -1.24. The skewness z value for boys' responses .91 and the kurtosis z value was -1.53. Additionally, Levene's test of homogeneity of variances was not significant $F(1,192) = 0.47, p > .05$, so no assumptions were violated. The result of the one-way ANOVA indicated that the difference between boys' and girls' reported computer programming anxiety was not statistically significant $F(1,192) = 1.674, p > .05$.

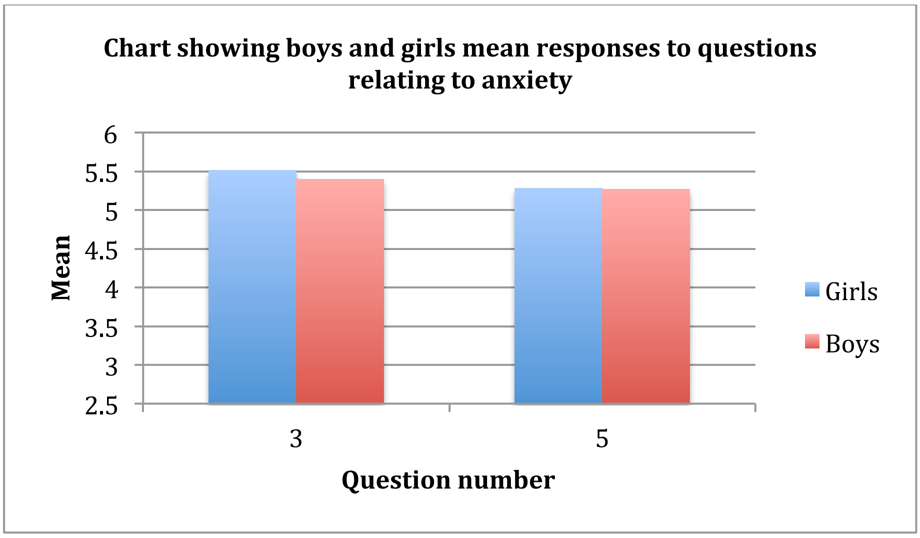


Figure 4.4 Chart showing mean scores of anxiety by gender (scale of 1 – 7) with the higher the number the less the anxiety

4.3.1.2 Confidence to complete tasks and study computer science at university

Aggregate confidence in computer programming was not significant, so additional analysis was conducted to see if there was any difference between girls' and boys' responses in confidence to complete programming tasks already studied and their confidence to study computer programming at university.

The programming test (Appendix K) contained three main sections: HTML/CSS, Python Turtle, Python solution to a Flowchart and a fourth optional assessment, the 'Rainfall Problem' extension activity. Participants were given an electronic copy of the test paper and asked to spend 30 seconds looking through it. They were then asked to rate their confidence by completing the three main sections of the CPCA Questionnaire (Appendix L) using a 7-point Likert scale (1- high confidence to 7 – low confidence). They were not asked to comment on their ability to complete the extension activity. Both male and female participants were more confident of correctly completing the Python Turtle task (girls M=3.04; boys M=2.91) than they were of completing the HTML/CSS task (girls M=3.62; boys M=3.41) or the Flowchart task (girls M=3.58; boys M= 3.45).

In order to assess the confidence of completing programming tasks immediately prior to attempting them, the three questions relating to confidence at completing the computer programming test (Items 8, 9, and 10) were averaged to provide a composite confidence measure. The composite mean score for girls was 3.43 (SD = 1.28) and the composite mean score for boys was 3.27 (*SD* = 1.14). To assess normality, a test for the skewness and kurtosis was conducted. The skewness z value was 1.79 for girls' responses and 3.59 for boys' responses. The kurtosis z value was .47 for girls' responses and 2.69 for boys' responses. Additionally, Levene's test of homogeneity of variances was run and resulted in a non-significant result, **F (1, 192) = 1.164, p > .05**. A one-way ANOVA was subsequently run and produced a non-significant result, **F (1, 192) = 0.90, p > .05**. Since an assumption of normality was violated, a non-parametric Mann-Whitney U-test was run. The results were girls (**Mdn = 3.3**), boys (**Mdn = 3.0**), **U = 4195, p = .319** and this aligns to the one-way ANOVA findings that girls' and boys' confidence of completing computer programming tasks were not significantly different.

The question asking the participants if they believed that they have the capability to successfully complete a university degree, or computer programming modules, in the future, showed slightly different responses with a mean of 3.75 (*SD* = 1.84) for the girls and a mean of 3.35 (*SD* = 1.30) for the boys. To assess normality, a test for the skewness and kurtosis was conducted. The skewness z value was 1.66 for girls' responses and 1.74 for boys' responses. The kurtosis z value was -1.33 for girls' responses and 0.002 for boys' responses. Additionally, Levene's test of homogeneity of variances was run and returned a significant result, **F (1, 184) = 9.932, p < .05**, indicating that rules of homogeneity were violated. A one-way ANOVA revealed that girls' and boys' reported confidence in completing Computer Science at university was not significant **F (1, 184) = 2.943, p > .05**. To validate this finding, a non-parametric, Mann-Whitney U-test was conducted and also revealed that there was no statistical difference between girls' (***Mdn* = 4.0**) and boys' (***Mdn* = 3.0**), **U = 3777.5, p = .253** responses.

With confidence and anxiety relating to computer programming analysed, the results from the SPCS Questionnaire (Appendix M) were considered. The purpose of the questionnaire was to investigate students' perceptions of Computer Science, in the hope that it would shed light on intrinsic factors that may influence their decision to choose to study IGCSE Computer Science. The following section outlines the quantitative responses from this questionnaire.

4.3.2 Computer programming enjoyment

If the premise that subject enjoyment is an important consideration when choosing to continue with a subject in high school is accepted, then the question do boys enjoy computer programming more than girls becomes key. To ascertain the enjoyment level of computer programming, the participants were asked the question, '*On average, and compared to your other classes, how enjoyable is computer science? (1 – much more enjoyable, 4 –- same, 7 – much less enjoyable)*' on the SPCS Questionnaire (Appendix M). The responses revealed a mean score of 2.85 (*SD* = 1.70) for girls and a mean score of 2.52 (*SD* = 1.38) for boys. This indicated that both genders felt that learning computer programming was more enjoyable than their other subjects, on average.

To assess normality, a test for the skewness and kurtosis was conducted. The skewness *z* value was 2.84 for girls' responses and 3.96 for boys' responses. The kurtosis *z* value was -.73 for girls' responses and 1.85 for boys' responses. Additionally, Levene's test of homogeneity of variances violated assumptions between the girls' and boys' responses, ***F* (1,189) = 7.01, *p* < .05**. A one-way ANOVA revealed that there was no significant difference in responses between the genders, ***F* (1, 189 = 2.189, *p* > .05).** To validate this finding a Mann-Whitney U-Test was conducted and indicated that there was no statistically significant difference (**U = 4052.5, p = .311**) between girls' (***Mdn* = 2.0**) and boys' (***Mdn* = 2.0**) enjoyment level.

As a means of adding reliability to this question, CPCA Questionnaire (Appendix L) asked the same question but in a slightly different way. The question was, '*The challenge of learning computer programming is exciting*' and using a Likert scale, with 1 equating to strongly agree and 7 equating to strongly disagree, the girls' mean score was 2.81 (*SD* = 1.52) and the boys' mean score was 2.47 (*SD* = 1.19). To assess normality, a test for the skewness and kurtosis was conducted. The skewness *z* value was 3.86 for girls' responses and 2.78 for boys' responses. The kurtosis *z* value was 1.54 for girls' responses and -.40 for boys' responses. Levene's test of homogeneity of variances was run and revealed a non-significant result, ***F* (1, 189) = 1.73, *p* > .05**. A one-way ANOVA was run and returned a non-significant result, ***F* (1, 189) = 3.079, *p* > .05**. To validate this finding, a Mann-Whitney U-Test was conducted and indicated that there was no statistically significant difference (***U* = 3958, *p* = .187**) between girls' (***Mdn* = 3.0**) and boys' (***Mdn* = 2.0**) self-reported enjoyment level. Thus it can be seen that neither question resulted in a significant gender difference in computer programing enjoyment.

As outlined in the literature review, there are researchers such as Natale (2002) that have suggested that computer gaming provided a mechanism to become computer literate through the engagement of spatial learning and cognitive processing abilities. Consequently, this has put girls at a disadvantage because of their lack of interest in computer gaming. The following section assesses if there is a gender difference in computer gaming interests.

4.3.2.1 Computer gaming enjoyment

The participants in this research study were asked to rate how much they enjoyed playing computer games on a scale from 1 (love) to 7 (hate). There was a noticeable difference in enjoyment with girls reporting a mean of 3.63 (*SD* = 1.74) and boys reporting a mean (standard deviation) of 2.36 (*SD* = 1.48). To assess normality, a test for the skewness and kurtosis was conducted. The skewness z value was 1.51 for girls' responses and 6.36 for boys' responses. The kurtosis z value was -.92 for girls' responses and 3.85 for boys' responses. Levene's test of homogeneity of variances was not significant, $\boldsymbol{F}$ **(1, 192) = 3.374,** $\boldsymbol{p > .05}$. A one-way ANOVA was run and revealed a significant result, $\boldsymbol{F}$ **(1, 192) = 29.727,** $\boldsymbol{p < .05}$. To validate this finding, a Mann-Whitney U-Test was conducted and indicated that there was a statistically significant difference ($\boldsymbol{U = 2527, p = .000}$) between girls' (**Mdn = 4.0**) and boys' (**Mdn = 2.0**) enjoyment level. Boys' enjoyment of computer games was significantly greater than girls' enjoyment of computer games.

To develop a greater understanding of whether the students felt playing games was beneficial or not, the participants were asked, '*In general, do you think computer games are a useful activity or a distraction from doing useful things? (1 – really useful, 7 – really distracting)*'. Girls' responses produced a mean of 4.43 (*SD* = 1.61) and boys' responses produced a mean (standard deviation) of 3.43 (*SD* = 1.76). To assess normality, a test for the skewness and kurtosis was conducted. The skewness z value was -1.09 for girls' responses and -1.77 for boys' responses. The kurtosis z value was 1.22 for girls' responses and -1.59 for boys' responses. Levene's test of homogeneity of variances was not significant, $\boldsymbol{F}$ **(1, 191) = 1.008,** $\boldsymbol{p > .05}$. A one-way ANOVA was run with a significant result, $\boldsymbol{F}$ **(1, 191) = 16.47,** $\boldsymbol{p < .05}$. Thus, boys tended to think computer games were somewhat beneficial, whereas girls felt that computer games were somewhat distracting. Overall though, the mean scores for both genders were close to 4 indicating some neutrality about the benefits of computer games.

It was not enough to determine how enjoyable the respective genders find playing computer games; it was also necessary to determine how much time they spent playing computer games. There were notable differences between boys' responses and girls' responses to question 11 (*'On average, how much time do you spend playing computer games?'*) on the SPCS Questionnaire, with 73% of girls, compared to 30% of boys reporting that they play computer games 20 minutes or less per week. Further, 94% of girls, compared to 57% of boys, reported that they play computer games an hour or less per week. Figure 4.5 shows clearly that the boys in this study, compared to girls, spend far more time playing computer games.

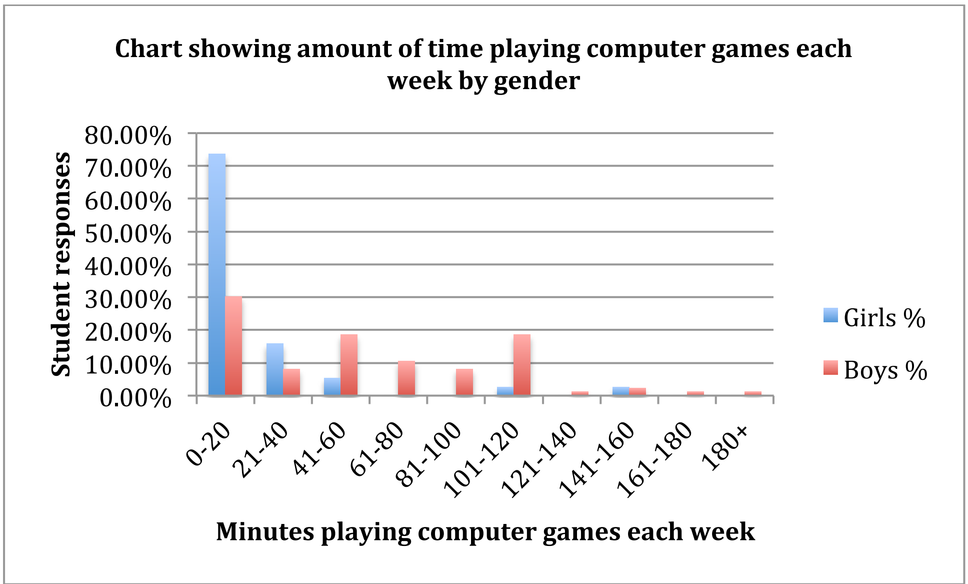


Figure 4.5 Participants' computer gaming minutes per week

Not only was there a difference in the amount of time spent playing computer games; the types of games that the boys and girls played were very different, with 46% of the boys choosing an online multiplayer as their favourite type of game, compared to 17% of the girls. Girls' favourite type of game was tied between puzzle and simulation, which both received 21% of the girls' votes, compared to boys reporting 1% for puzzle and 5% for simulation. Figure 4.6 shows the full results of the participants' preferred choice of computer game genre.

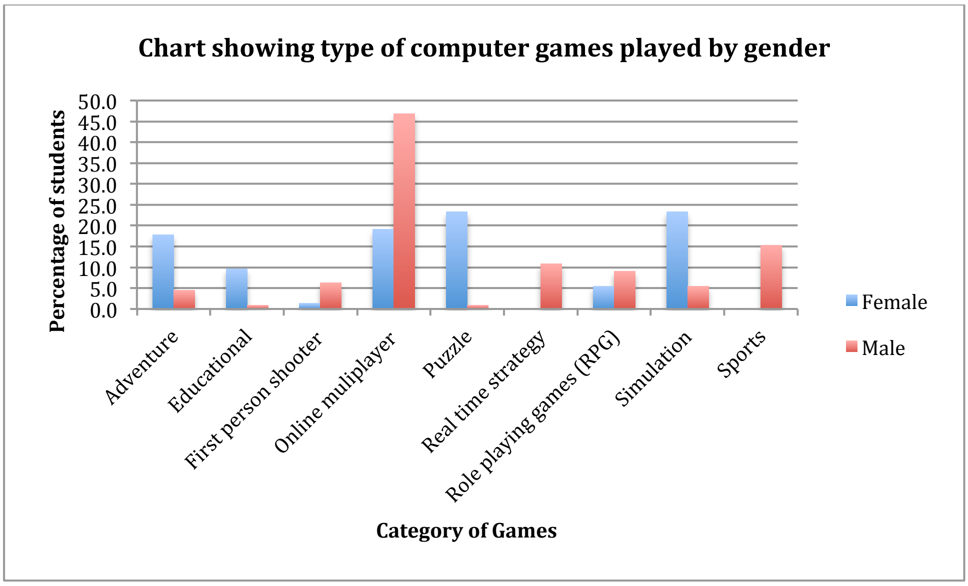


Figure 4.6 Participants' responses to the preferred game genre

From an educational perspective, there was a difference in responses, with 9% of the girls reported that educational games were their favourite game. Only 1% of boys reported that educational games were their favourite type of game. These relatively low percentages suggest that when the participants choose to play games they do so for enjoyment rather than learning. If you combine puzzle games with educational games because they are similar in many ways, then nearly a third of girls, compared to just 2% of boys, chose this genre as their favourite computer game.

Overall, the responses to questions about computer games suggest that boys choose to play games for fun and have significantly more interest in computer games. Further, girls play very different games to boys and seem more conscious of the distraction this activity can have, as evidenced above.

4.3.3 Computer programming difficulty

Students might not choose a subject if they find it too difficult or not challenging enough. To ascertain the difficulty level of computer programming, the participants were asked the question, '*On average, and compared to your other classes, how difficult/easy is computer science? (1 – much more difficult, 4 –- same, 7 – much easier)*' on the SPCS Questionnaire (Appendix M). The responses resulted in a mean score of 4.06 (*SD* = 1.44) for girls and a mean score of 4.10 (*SD* = 1.63) for boys. This indicated that both genders felt that learning computer programming was at a similar level of difficulty to other subjects.

The skewness z value was 0.27 for girls' responses and -0.04 for boys' responses. The kurtosis z value was -1.07 for girls' responses and -1.58 for boys' responses. After carrying out a Levene's test, $\boldsymbol{F}$ **(1,190) = 2.39,** $\boldsymbol{p}$ **> .05**, to ensure that the rules of homogeneity of variance were not violated, a one-way ANOVA reported that there was no significant difference, $\boldsymbol{F}$ **(1, 144.76) = 2.035,** $\boldsymbol{p}$ **> .05**, between the boys' and girls' responses. Thus, the perceived difficulty of computer programming was similar for girls and boys.

4.3.4 Future plans

Question 6 on the SPCS Questionnaire (Appendix M) asked, '*Do you think you will choose to study computer programming in the future? (1 – will study, 4 – no idea, 7 – won't study)*' with girls' responses resulting in a mean and standard deviation of 3.72 (SD = 1.86) and boys' responses resulting in a mean and standard deviation of 2.82 (SD = 1.56). The skewness z value was 1.29 for girls' responses and 4.02 for boys' responses. The kurtosis z value was -1.59 for girls' responses and -1.51 for boys' responses. Levene's test of homogeneity of variances was significant, $\boldsymbol{F}$ **(1, 176) = 3.975,** $\boldsymbol{p}$ **< .05**, and violated the rules of normality. So after the one-way ANOVA was conducted and this produced a statistically significant result, $\boldsymbol{F}$ **(1, 142.60) = 11.734,** $\boldsymbol{p}$ **< .05**, a non-parametric, Mann-Whitney U-test was run. This also produced a significant result, (girls (***Mdn*** **= 4.0**); boys (***Mdn*** **= 3.0**), $\boldsymbol{U}$ **= 2759,** $\boldsymbol{p}$ **= .001**), indicating that the one-way ANOVA was reliable.

4.3.5 Perceptions of computer programming

The 2015 PISA survey reported that gender-related differences in science engagement and career aspirations are related to differences in what boys and girls believe they can do, rather than what they are capable of achieving (OECD, 2016). Thus, in an effort to ascertain the students' perception over which gender was more likely to do well in the computer programming assessment, the Computer Programming Confidence and Anxiety Questionnaire (CPCA) (Appendix L) included two additional Likert-style questions using a scale of 1 – boys to 7 girls. The questions were, '*Do you think that there is any difference in academic abilities between boys and girls in your year group?*' and '*Do you think there will be any difference between boys and girls in the results of this assessment?*'. A score of 7 would indicate that the belief that girls have a greater academic ability or a greater programming ability. Both boys and girls reported that there was little difference in academic ability between the genders. The girls' mean score was 4.40 (*SD* = 1.44) and the boys' mean score was 4.30 (*SD* = 1.47). To assess normality, a test for the skewness and kurtosis was conducted. The skewness z value was 1.66 for girls' responses and 1.74 for boys' responses. The kurtosis z value was -1.33 for girls' responses and 0.002 for boys' responses. Levene's test of homogeneity of variances revealed that there was not a significant difference between the boys' and girls' variances, $\boldsymbol{F}$ **(1, 189) = 0.15,** $\boldsymbol{p}$ **> 0.05**. A one-way ANOVA test revealed no significant difference, $\boldsymbol{F}$ **(1, 189) = 0.23,** $\boldsymbol{p}$ **> 0.05**, between the girls' and boys' belief in how able they thought girls were relative to boys.

The findings for the second question indicated that both boys and girls believed that there was little difference in programming ability between the genders. The girls' mean score was 4.01 (*SD*=1.27) and the boys' mean score was 4.19 (*SD*=1.47). To assess normality, a test for the skewness and kurtosis was conducted. The skewness z value was 1.88 for girls' responses and -.69 for boys' responses. The kurtosis z value was 0.27 for girls' responses and -0.10 for boys' responses. Levene's test of homogeneity of variances revealed that there was not a significant difference between the boys' and girls' variances, $\boldsymbol{F}$ **(1, 191) = 1.86,** $\boldsymbol{p}$ **> 0.05**. With no assumptions violated a one-way ANOVA was run and revealed no significant difference, $\boldsymbol{F}$ **(1, 191) = 0.75,** $\boldsymbol{p}$ **> 0.05**, between the between the girls' and boys' belief in how capable of computer programming they thought girls were relative to boys.

Question 3 on the Student Perceptions of Computer Science (SPCS) Questionnaire (Appendix M) asked the question, '*Do you think boys or girls are more suited to a career in computer programming? (1 – boys, 4 – no difference, 7 – girls)*'. The girls' mean score was 3.46 (*SD* = 1.11) and boys' mean score was 3.22 (*SD* = 1.07). To assess normality, a test for the skewness and kurtosis was conducted. The skewness z value was -0.75 for girls' responses and -3.19 for boys' responses. The kurtosis z value was 2.30 for girls' responses and -0.46 for boys' responses. Additionally, Levene's test of homogeneity of variances revealed that there was not a significant difference between the boys' and girls' variances, $\boldsymbol{F}$ **(1, 191 = 0.121,** $\boldsymbol{p}$ **> 0.05)**. A one-way ANOVA revealed that there was no significant difference in responses between the genders, $\boldsymbol{F}$ **(1, 191 = 2.149,** $\boldsymbol{p}$ **> 0.05).** To validate this result, a non-parametric, Mann-Whitney U-test was conducted and also revealed that there was no statistical difference between girls' (***Mdn*** **= 4.0**) and boys' (***Mdn*** **= 4.0**), $\boldsymbol{U}$ **= 4089,** $\boldsymbol{p}$ **= .200** responses. These findings indicate that both boys and girls do not have gender stereotypes for careers in computer programming.

4.4 Quantitative Results - Research Question 3 ('Are there any extrinsic factors that affect girls choosing to study computer programming in Year 10?')

This section of the quantitative findings is composed of the results summary from the Influences on IGCSE Choices (IIC) Questionnaire (Appendix N).

4.4.1 Results from the Influences on IGCSE Choices Questionnaire

This online questionnaire was distributed to Year 9 and Year 10 students. There was a similar completion rate percentage between the genders with 74 of the 114 (65%) girls and 94 of the 143 boys (66%) completing the survey.

Students were asked to rank from 1 – strongly influence to 7 – no influence, inclusively, how much these factors (parents, teachers, friends, useful life skills, career/university plans, lesson enjoyment, subject difficulty, and role models) influenced their IGCSE choices. Figure 4.7 shows the mean responses for both boys and girls with the tallest bars representing the biggest influencing criteria and the shortest bars representing the smallest influencing criteria.

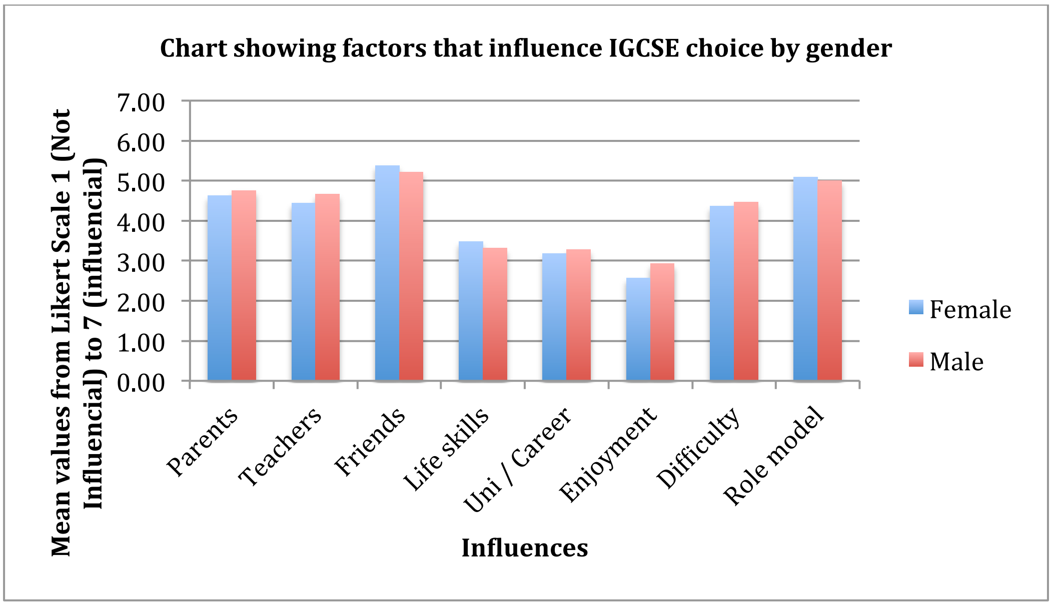


Figure 4.7: Factors that influence participants' IGCSE choices

Responses from both genders were very similar; with enjoyment ranking highest on the reason they choose an IGCSE subject. Career / university, life skills and subject difficulty were all ranking more important considerations than people. Of the direct influence of people, teachers were ranked slightly higher than parents. Friends were not considered particularly important when choosing subjects to continue to study and were ranked last.

Skewness *z* value and Kurtosis *z* value were calculated to test for normality of data. Following this, Levene's test of homogeneity of variances was conducted followed by a one-way ANOVA test. Where assumptions were violated a Mann-Whitney U-test was conducted. The results showed that there was no significant difference between the influencing factors reported by girls and boys (see Table 4.4).

Table 4.3

Responses to the Influences on IGCSE Choices Questionnaire showing the mean, standard deviation, skewness z value, kurtosis z value, Levene's test of homogeneity of variances, and results from a one-way ANOVA and Mann-Whitney U-test (Responses used a 1-7 Likert Scale with 1-strong influence to 7 – no influence)

Questions	girls' data	boys' data	Levene's test One-way ANOVA Mann-Whitney U-test
2. How much did your parent(s) influence your decision?	*M*=4.63 *SD*=1.48 *Sz*=-.49 *Kz*=-1.84	*M*=4.75 *SD*=1.71 *Sz*=-1.53 *Kz*=-1.37	F (1,163) = 1.41, $p > .05$ F (1,163) = 0.226, $p > .05$
3. How much did your teacher(s) influence your decision?	*M*=4.45 *SD*=1.67 *Sz*=.54 *Kz*=-1.90	*M*=4.67 *SD*=1.73 *Sz*=-.48 *Kz*=-2.37	F (1,165) = 0.222, $p > .05$ F (1, 165) = 0.692, $p > .05$ Girls (*Mdn* = 4.0), Boys (*Mdn* = 5.0), U = 3183, p = .398
4. How much did your friend(s) influence your decision?	*M*=5.38 *SD*=1.52 *Sz*=-2.23 *Kz*=-1.55	*M*=5.22 *SD*=1.69 *Sz*=-3.12 *Kz*=-.49	F (1,164) = 0.395, $p > .05$ F (1, 164) = 0.406, $p > .05$ Girls (*Mdn* = 6.0), Boys (*Mdn* = 6.0), U = 3265, p = .643
5. How much did useful life skills to be learned influence your decision?	*M*=3.48 *SD*=1.44 *Sz*=1.23 *Kz*=-.41	*M*=3.31 *SD*=1.40 *Sz*=.78 *Kz*=-.79	F (1,161) = 0.039, $p > .05$ F (1, 161) = 0.568, $p > .05$
6. How much did your university plans or career plans influence your decision?	*M*=3.18 *SD*=1.69 *Sz*=2.35 *Kz*=-.80	*M*=3.27 *SD*=2.00 *Sz*=2.37 *Kz*=-.49	F (1,162) = 0.024, $p > .05$ F (1,162) = 0.134, $p > .05$ Girls (*Mdn* = 3.0), Boys (*Mdn* = 3.0), U = 3174, p = .618

7. How much has lesson enjoyment influenced your decision?	M=2.57 SD=1.40 Sz=5.39 Kz=4.84	M=2.92 SD=1.45 Sz=2.91 Kz=1.03	F (1,165) = 0.025, $p > .05$ F (1, 165) = 2.59, $p > .05$ Girls (Mdn = 2.0), Boys (Mdn = 3.0), U = 2867, p = .051
8. How much has subject difficulty influenced your decision?	M=4.37 SD=1.00 Sz=-.05 Kz=-1.33	M=4.47 SD=1.07 Sz=.12 Kz=-2.48	F (1,164) = 0.377, $p > .05$ F (1, 164) = 0.389, $p > .05$ Girls (Mdn = 4.0), Boys (Mdn = 4.5), U = 3169, p = .566
9. How much has a role model (not a parent or teacher) inspired you and influenced your decision?	M=5.09 SD=1.75 Sz=-2.01 Kz=-1.61	M=5.00 SD=1.92 Sz=-1.99 Kz=-2.04	F (1,166) = 1.24, $p > .05$ F (1, 166) = 0.109, $p > .05$ Girls (Mdn = 5.5), Boys (Mdn = 5.0), U = 3455, p = .940

Note: The following abbreviations have been used: M = Mean, SD = Standard Deviation, Sz = skewness z value, Kz = Kurtosis z value, Mdn = Median.

Now that the quantitative data on computer programming ability, intrinsic influences, and extrinsic influences have been considered, the preferred learning style of both genders is considered.

4.5 Quantitative Results – Research Question 4 ('Is there a significant gender difference in the preferred learning styles for computer programming?')

Students were given the Preferred Computer Programming Learning Style Questionnaire (Appendix O) asking about the method in which they liked to learn. Identifying whether there is a gender difference in the method of learning programming is useful to determine if girls and boys should learn differently. Questions focus on learning to program utilising flowcharts, graphics, independently, collaboratively, cooperatively, project-based, problem-solving tasks, and storyline-based.

4.5.1 Results from the Preferred Computer Programming Learning Style Questionnaire

Students were given the online PCPLS Questionnaire (Appendix O) to investigate what was their preferred method of learning to program a computer. Of the 194 students that completed the questionnaire, there were 81 girls and 113 boys. The questions' mean, standard deviation, skewness z value, kurtosis z value, Levene's test of homogeneity of variances, and a one-way ANOVA are shown in Table 4.5. Where the assumptions (i.e. skewness, kurtosis, and homogeneity) were violated a non-parametric Mann-Whitney test was also reported.

Table 4.4

Preferred Computer Programming Learning Style Questionnaire responses showing the mean, standard deviation, skewness z value, kurtosis z value, Levene's test of homogeneity of variances, and results from a one-way ANOVA and Mann-Whitney U-test (Responses took the form of a 1-7 Likert Scale with 1-strongly agree to 7 – strongly disagree)

Questions	Girls Frequencies	Boys Frequencies	Levene's test ANOVA Mann Whitney U (2-tailed)
1. I really enjoy learning to program by solving small problems.	*M*=2.79 *SD*=1.40 *Sz*=3.19 *Kz*=0.51	*M*=2.96 *SD*=1.58 *Sz*=4.52 *Kz*=1.45	F (1,191) = 0.03, $p > .05$ F (1,191) = 0.642, $p > .05$ Girls (Mdn = 2.5), Boys (Mdn = 3.0), U = 4256, p = .476
2. I find completing short problems (tutorials) in code difficult and often need help!	M=3.81 SD=1.96 *Sz*=0.65 *Kz*=-2.43	*M*=3.65 *SD*=2.00 *Sz*=2.06 *Kz*=-2.25	F (1,192) = 0.039, $p > .05$ F (1, 192) = 0.307, $p > .05$ Girls (Mdn = 3.0), Boys (Mdn = 3.0), U = 4402 p = .646
3. When solving a problem it really helps if I can see the program as a flowchart.	*M*=3.13 *SD*=1.40 *Sz*=0.95 *Kz*=-0.94	*M*=3.03 *SD*=1.55 *Sz*=4.07 *Kz*=1.42	F (1, 189) = 0.016, $p > .05$ F (1, 189) = 0.208, $p > .05$ Girls (Mdn = 3.0), Boys (Mdn = 3.0), U = 4094, p = .369
4. I find completing tutorials with short program problems develops my programming skills much more than working on a project.	*M*=3.68 *SD*=1.68 *Sz*=1.56 *Kz*=-0.49	*M*=3.17 *SD*=1.72 *Sz*=3.62 *Kz*=-0.55	F (1,184) = 0.352, $p > .05$ F (1, 184) = 4 051, $\mathbf{p < .05}$ Girls (Mdn = 4.0), Boys (Mdn = 3.0), U = 3317, $\mathbf{p = .015}$
5. When learning programming concepts, (e.g. loops, subroutines) in Python, do you find it more interesting to learn it graphically (e.g. turtle, tkinter) rather than text-based?	*M*=2.77 *SD*=1.40 *Sz*=2.94 *Kz*=-0.93	*M*=3.32 *SD*=1.60 *Sz*=1.98 *Kz*=-1.09	F (1, 188) = 2.48, $p > .05$ F (1, 188) = 5.927, $\mathbf{p < .05}$ Girls (Mdn = 3.0), Boys (Mdn = 3.0), U = 3521, $\mathbf{p = .018}$
6. When learning programming, I really enjoy working independently on my own.	*M*=3.36 *SD*=1.54 *Sz*=1.37 *Kz*=-0.92	*M*=3.32 *SD*=1.86 *Sz*=1.39 *Kz*=-2.60	F (1, 183) = 9.26, $\mathbf{p < .05}$ F (1,183) = 0.018, $p > .05$ Girls (Mdn = 3.0), Boys (Mdn = 3.0), U = 3937.5, p = .668
7. When learning	*M*=3.14	*M*=3.18	F (1, 187) = 0.542, $p > .05$

programming, I really enjoy working collaboratively, where I, and a friend, write the program together.	*SD*=1.75 *Sz*=1.53 *Kz*=-1.68	*SD*=1.91 *Sz*=2.86 *Kz*=-1.03	F (1, 187) = 0.023, p > .05 Girls (Mdn = 3.0), Boys (Mdn = 3.0), U = 4349, p = .946
8. When learning programming, I really enjoy working cooperatively, where I work on one task and my friend works on another, before we combine them to complete the project.	*M*=2.86 *SD*=1.56 *Sz*=3.39 *Kz*=0.67	*M*=3.35 *SD*=1.87 *Sz*=2.85 Kz=-1.57	F (1, 190) = 7.30, $\boldsymbol{p < .05}$ F (1,190) = 3.648, p > .05 Girls (Mdn = 3.0), Boys (Mdn = 3.0), U = 3915.5, p = .118
9. I find working on a project improves my programming skills much more than completing short exercises.	*M*=2.75 *SD*=1.44 *Sz*=2.02 *Kz*=.26	*M*=2.95 *SD*=1.55 *Sz*=1.72 *Kz*=1.46	F (1,182) = 0.636, p > .05 F (1, 182) = 0.789, p > .05 Girls (Mdn = 3.0), Boys (Mdn = 3.0), U = 3949, p = .623
10. When learning to program I find having a storyline really helps my learning.	*M*=2.90 *SD*=1.44 *Sz*=3.77 *Kz*=1.18	*M*=3.52 *SD*=1.68 *Sz*=2.01 *Kz*=-0.58	F (1, 191) = 3.56, p > .05 F (1, 191) = 7.154, $\boldsymbol{p < .05}$ Girls (Mdn = 3.0), Boys (Mdn = 3.0), U = 3563, $\boldsymbol{p = .009}$

Note: The following abbreviations have been used: M = Mean, SD = Standard Deviation, Sz = skewness z value, Kz = Kurtosis z value, Mdn = Median.

The findings from the questionnaire (Table 4.5) showed that there were differences in the learning styles boys and girls preferred. Girls enjoyed learning to program in a variety of ways with short problem-solving tasks, graphical outcomes, project-work, and storyline all achieving similar mean scores. Boys expressed a similar preference for short problem-solving tasks and project-work but the mean scores for storytelling and graphical outcomes were noticeably lower. Both boys and girls enjoyed collaborating on a project but girls' responses indicated that they prefer cooperating slightly more than boys did.

Using a parametric test such as an ANOVA with Likert ordinal ranking is an accepted practice (Brown, 2011), and so the following questions displayed a *significant* gender difference. Boys, relative to girls, reported completing tutorials and short problems more useful for developing programming skills, $\boldsymbol{F}$ **(1, 163.5) = 4 081,** $\boldsymbol{p}$ **< .05**. Girls, relative to boys, reported that learning programming in Python was more interesting when it included visual design elements through Python Turtle and Python Tkinter, $\boldsymbol{F}$ **(1, 180.0) = 6 201,** $\boldsymbol{p}$ **<**

.05. Additionally, girls, relative to boys, reported that learning to program using storytelling really helps their learning, ***F* (1, 185.48) = 7 52, *p* < .05**.

4.6 Summary

The quantitative findings for Research Question 1 reported that, on average, girls significantly outperformed boys in a test of programming ability. At the top end of performance, there was no gender difference in programming ability. The findings for Research Question 2 indicated that although boys were very slightly more confident than girls, it was not statistically significant. Nor was there any gender difference in computer programming anxiety. Boys also reported slightly more computer programming enjoyment, relative to girls, but this was also not statistically significant. Both genders agreed that subject enjoyment was the biggest influence in choosing to continue studying a subject in Year 10. The findings for Research Question 3 indicated that the biggest external influence in choosing to study a subject was the teacher, followed closely by the students' parents. The IGCSE subject chosen by their friends had little influence on IGCSE choices. The findings for Research Question 4 indicated that girls enjoyed storytelling and writing programs that incorporate graphics much more than boys did. Further, boys found completing tutorials and then solving short problems more engaging than the girls did.

Given the quantitative findings, it would also be expected that the number of girls choosing IGCSE Computer Science would increase from the baseline year, when the only teaching strategy used was tutorials with short problems to solve. This was the case with 17 girls (38%) in Year 9, compared to 5 girls (13%) the year before, choosing to continue their computer programming education by selecting IGCSE Computer Science as one of their two open options. The number of boys choosing to study IGCSE Computer Science also increased from 34 boys (52%) to 58 boys (69%). Since there were 39 students taking Computer Science in the previous year, girls represented approximately 13%, and boys 87%, of the students. The following year there were 75 students taking Computer Science and 23% of these were female.

With the quantitative findings reported, the next chapter presents the qualitative findings collected from the open-ending questions and group interviews.

Chapter 5: Qualitative Findings

5.1 Introduction

This chapter reports the findings from group interviews and the open-ended questions on the questionnaires. It is structured by the thematic coding of the group discussions and the four research questions on computer programming ability, intrinsic influences on IGCSE choices, extrinsic influences on IGCSE choices, and the preferred learning style.

To reiterate, there were a total of 13 students (11 girls and 2 boys) interviewed in small groups with a total of 6 interviews (see Table 5.1). This provided an opportunity to explore aspects that were not fully understood from the questionnaires and to identify any areas that might have been overlooked. The focus of the group interviews was to answer research question 2 (*'Are there any intrinsic factors that affect girls choosing to study computer programming in Year 10?'*) and research question 3 (*'Are there any extrinsic factors that affect girls choosing to study computer programming in Year 10?'*) because it was these questions that required deeper analysis.

Table 5.1

Interviewed students by pseudonym and what subjects they took for IGCSE

Student	**Gender**	**IGCSE subject choices**
Janice	Female	Art, History*, Mandarin
Bella	Female	Drama, Geography*, History*
Eunice	Female	Art, Geography*, History*
Yuna	Female	Dance, Geography*, Mandarin
Kay	Female	**Computer Science**, History*, Mandarin
Sarah	Female	**Computer Science**, History*, Mandarin
Gina	Female	**Computer Science**, History*, Mandarin
Seo Yon	Female	**Computer Science**, History*, Mandarin
Dain	Female	**Computer Science**, Geography*, History*
ChaeEun	Female	**Computer Science**, Geography*, History*
Senna	Female	**Computer Science**, Dance, Geography*
Linus	Male	**Computer Science**, Drama, History*
Eric	Male	**Computer Science**, Drama, Geography*

** Students were required to take either History or Geography or could take both*

There was a preliminary attempt to discuss research question 4 (*'Is there a significant gender difference in the preferred learning styles for computer programming?'*) with a group of 10 randomly selected girls who studied IGCSE Computer Science to see if it was worthwhile pursuing this line of inquiry. After ten minutes of trying different lines of questioning, it was apparent that they were not able to explain their preferences fully. When asked about their preferred method of learning to program a computer the girls went quiet, looked at the ground, and a couple said that they weren't sure. The girls' body language gave the impression that they were uncomfortable answering a question about their learning. This may have been because Korean students wouldn't want to criticise a teacher's strategies for learning or because they were asked in front of their peers. Regardless, it was not appropriate to continue with this line of questioning so it was not pursued. Moreover, the open-ended questions on the Preferred Computer Programming Learning Style Questionnaire (Appendix O) provided sufficient detail to answer research question 4. None of the semi-structured interview questions related to research question 1 ('*Is there a significant gender difference in computer programming ability?'*), because this was assessed through the programming ability test. Although, some of the discussion points related to the students' perception of whether there was a gender difference in computer programming ability.

To assist with the analysis of the group interviews inquiring about research questions 2 and 3, Lichtman's (2012) six-stage approach was used and the comments were added to the initial reading of the group interviews (Appendix Q). After several re-readings, writing and collapsing themes (Appendix R), the final themes and sub-themes, shown in Table 5.1, emerged.

Table 5.2

Narrative interviews analysed using Lichtman's thematic coding showing frequency and number of sources

Level 1 Theme	**Level 2 Theme**	**Level 3 Theme**
Intrinsic (69, 13)	• Ability (8, 6) • Confidence (2, 2) • Creativity (5, 4) • Difficulty (9, 7) • Dislike (4, 4)	

	• Enjoyment (15, 8)	
	• Future Aspirations (13, 8)	
	• Interest (13, 8)	
Extrinsic (42, 13)	• Adults (38, 13)	o Parents (26, 13) o Teachers (12, 8)
	• Friends (2, 2)	
	• Required (2, 2)	
Gender perception (12, 7)	• Genetic (3, 3)	
	• Social (9, 4)	

Note: The first number in brackets relates to the frequency that a theme was mentioned and the second number relates to the number of participants that mentioned the theme.

The use of the software application, nVivo, provided additional benefits like colour-code themes (Appendix S) and rapidly quantifying data. Moreover, the process of creating themes and sub-themes was helpful in obtaining an inclusive view of the information and assisted in answering the research questions. The number of times that a particular theme was mentioned is shown in Table 5.1, although the importance of a theme is not determined by the number of occurrences but by its 'substantive significance' (Patton, 2002), meaning that the significance of what was being said is more important than the number of times a theme appeared in the text.

5.2 Qualitative Results - Research Question 1 ('Is there a significant gender difference in computer programming ability?')

This research question was partially answered in the quantitative findings chapter, with girls achieving a significantly higher score compared to boys. To expand on these findings, additional qualitative information was gathered on students' perceptions of programming ability through group interviews with the Year 10 students, who were in Year 9 when the programming test was taken. The thematic coding of the keyword 'genetic' helped identify that 3 of the 8 girls, who were asked why they thought more boys than girls chose IGCSE Computer Science, suggested that there is a belief that boys have greater ability in mathematics and the sciences because of neurobiological differences. When asked why more boys, compared to girls, choose computer science, Janice stated that:

> "it's a stereotype but we tend to think that boys have more developed left-brain."

"left brain people tend to be better at Math and Computer Science"

In a different interview, Kay also suggested that boys' brains are different from girls and more suited to using a computer. Yuna had a similar opinion to Janice and Kay, and she stated that:

> "I heard men's brain are much developed in Maths or Science thing than girls and that influence them"

When Yuna was asked where she had heard or seen that information she responded that it was a newspaper article reviewing a research paper. Two other interviewees suggested that there were differences in ability but these were due to social factors, as identified by the 'social' theme. For example, Senna suggested that South Korea parents tend to believe that boys are better at Mathematics and Sciences and encourage them to study these subjects. She went on to say that parents' beliefs might affect girls' thinking and cause them to choose different subjects.

The remaining girls did not comment that there was a difference in ability and instead reasoned that boys were more interested in robotic animations and gaming, whereas girls prefer art, dance and drama. This was also the belief of the two boys who were interviewed, although one of them suggested that boys were generally good at Computer Science because they spent more time using computers for gaming.

All of the interviewed girls who commented that boys were more suited to Computer Science, scored highly on their computer programming test. Thus, it would seem that social stereotypes, rather than actual ability, may play an important role in influencing girls' decisions to choose IGCSE Computer Science. To determine if there are any other influences, the qualitative findings are examined in relation to the second research question.

5.3 Qualitative Results - Research Question 2 ('Are there any intrinsic factors that affect girls choosing to study computer programming in Year 10?')

This section of the qualitative findings is based on students' responses in the group interviews and from the Student Perceptions of Computer Science Questionnaire's (Appendix M) open-ended questions.

5.3.1 Computer programming confidence and anxiety

The responses from the group discussions produced no evidence of either confidence or anxiety towards Computer Science from either gender. When designing the semi-structured interview questions, it was decided not to ask about confidence or anxiety directly but to see whether the conversation around subject choices related to aspects of confidence or anxiety. Further, the four female students who did not choose IGCSE Computer Science were asked if they would consider choosing Computer Science at International Baccalaureate Diploma level to see whether confidence or anxiety were mentioned in their decision not to choose Computer Science. However, the thematic coding of the keyword 'confidence' only produced two hits and both related to a lack of confidence in the creative subjects, not a lack of confidence in computer programming. Thus, it can be concluded that after a year of learning computer programming, using a variety of different programming strategies, girls' confidence was similar to that of boys. This was in line with the quantitative results obtained from the CPCA Questionnaire (Appendix L) that showed there was not a significant gender difference in confidence or anxiety.

5.3.2 Computer programming enjoyment

The responses from the group interviews did indicate that some girls believe that boys are more interested in Computer Science than girls. Two of the girls who did not choose to study IGCSE Computer Science commented that boys grew up with an interest in computers. To the question "Why do you think many more boys than girls choose to study Computer Science?", Yuna stated that:

> "I think also mostly boys watch robotic animation, like cartoon, which is about technology. Um, since girls watch girlish thing, and pretty thing, kind of related to art. I think boys as they grow up tend to [be] interested in computing and other technology and girls are more interested in art or drama."

These were not the only responses of this nature and several girls and both boys commented that more boys choose to study Computer Science because they enjoy robotics, gaming, and technology-related topics. It would seem that the girls relate boys' interest in technology-related areas, like robotics and gaming, to an interest in computer programming. Their statements that girls are interested in art-related, girlish and pretty television programmes appear to support the PISA 2015 findings (OECD, 2015) that Computer Science is a *masculine* subject. If children are surrounded by stereotypes such as these, then it seems reasonable to suggest that this influences their intrinsic belief system and that girls would, perhaps subconsciously, opt out of Computer Science.

When students were asked during the group interviews why they chose particular subjects for IGCSE, there was an indication that subject enjoyment was important, with this theme mentioned on 15 occasions by 8 participants. The following comments from two different students were fairly typical of the reasons given for continuing to study a subject:

> "I always got good grades in Computer Science and it was really fun so I definitely chose it"

> "I was already interested in coding and programming and so at first when I see Computer Science was in the subject [options] and I can learn, I was happy to see and happy learn about it so I choose it first."

The keyword 'dislike' supported the view that subject enjoyment is important. Four girls commented that that did not choose certain subjects, notably Geography, Mandarin, and History because they disliked the subject. Further, the theme keyword 'enjoyment' was identified in the statements made by 6 girls and 2 boys. All of their comments related to choosing a particular subject because they liked to study it. In a similar way, the thematic keyword 'interest' was identified in the comments of 8 girls who related to the links between their interests and the subjects they chose. For example, Eunice, who was one of the girls who did not choose IGCSE Computer Science, stated that she was not interested in computers and implied that was why she did not choose to study Computer Science.

The importance of subject enjoyment can be seen in the interviews, and this reaffirms the quantitative findings that identified that subject enjoyment was the most important factor in choosing to continue studying a subject. Further, the mean response from both boys and girls indicated that Computer Science was enjoyable, relative to other subjects. It can, therefore, be concluded that subject enjoyment is an important intrinsic motivation for choosing to study IGCSE Computer Science.

5.3.3 Computer programming difficulty

Students rated subject difficulty as an important contributing factor when choosing IGCSE subjects. Additionally, both genders on average reported that learning to program a computer in Computer Science was similar in difficulty to other subjects. To investigate this further group interviews were conducted and Bella, who did not choose to study Computer Science in Year 10, commented, "I think Computer Science um, in the beginning it's quite easy but later on is quite difficult."

Dain who chose to study IGCSE Computer Science also remarked that computer programming was difficult by commenting, "it is a language but a language is difficult to learn so I am getting used to it but it is not easy."

Only these two of the thirteen interviewees mentioned subject difficulty and, of these, one had chosen to study Computer Science but the other one had not. Thus, there is little, or no, evidence to suggest that the difficulty of learning computer programming influenced girls' decision not to continue their education in Computer Science.

The thematic keyword 'difficulty' did not always relate to computer programming with only 4 of the 9 instances relating to that subject. The other instances related to the difficulty of other subjects or the skills required for other subjects. For example, stage fights in drama.

5.3.4 Future plans

Computer programming is widely considered an important skill for future employment opportunities and for fitting in with the digital world. The quantitative findings suggested that life skills and future career plans were important considerations when choosing IGCSE subjects. The interviews suggest that some students are aware of the importance of learning to use technology, with Bella, who did not choose to study ICGSE computer programming, commenting that she wished that she had chosen Computer Science. When asked why she had changed her mind, she replied that:

> "Cause, um since the world, since the world and company kind of develop and want to develop their technology so they tend to find more people who did major of Computer Science and also when I am doing IGCSE Computer Science it could be better for my future career."

Gina also considered her future when choosing IGCSEs and said that:

> "I choose Computer Science because um I want to work in Computer thing in the future, so I thought it would be helpful to choose Computer Science."

In the group discussions, there were 13 instances from 5 of the 6 different interview transcripts of the theme 'Future Aspirations' being mentioned in relation to future study or career. This supports the quantitative findings that students consider what they want to do in the future when choosing to continue studying a subject.

5.3.5 Perception of computer programming

In an effort to answer Research Question 2 (*'Are there any intrinsic factors that affect girls choosing to study computer programming in Year 10?'*) and uncover gender differences relating to computer programming, students were asked 'What skills or abilities are useful for computer programming?' using an open-ended question on the SPCS Questionnaire (Appendix M). The rationale for the question was to see if boys identified computer programming with mathematics and girls identified computer programming with language. Of the 194 students, 67% of the girls' responded and 74% of the boys' responded, totalling 138 (71%) who wrote answers, although a number of them lacked meaning. Of the skills included in the students' responses, the most common themes mentioned were fast typing, maths abilities and logical thinking (see Table 5.2). There was a distinct difference in one particular theme, and of the 16 students that identified mathematics as an important ability, only 3 were girls, or 6% of responders, in comparison to 13 boys, or 16% of responders.

Table 5.3

Frequency of common themes from question 4 on the Student Perceptions of Computer Science Questionnaire

	% Girls	N	% Boys	N	% Total
Fast typing	11%	11	13%	17	12%
Maths ability	6%	3	16%	13	11%
Logical thinking	11%	10	12%	15	11%
Perseverance	6%	3	4%	6	4%

Memory	4%	3	4%	5	4%
Problem-solving	2%	2	2%	3	2%

In the hope of ascertaining if there is a gender difference in the importance of being able to program a computer, students were asked *if they believe that learning computer programming was an important activity or not.* There were 146 responses (75% of the students) to this open-ended question although a few of the comments were little more than one or two words. Of these responders, 60 were girls (74%) and 86 were boys (76%).

The results of the thematic coding showed that the overwhelming majority of boys and girls that responded believed that learning computer programming was an important activity because computers are ubiquitous in society (see Table 5.3). Further, twice as many boys than girls (14% versus 7%) identified that learning to program a computer provided career opportunities. However, considerably more girls, relative to boys, felt that learning computer programming was not an important activity (Girls 20% versus Boys 3%).

Table 5.4

Frequency of common themes from question 5 on the Student Perceptions of Computer Science Questionnaire

Level 1 Theme	Level 2 Theme
Future Boys 42 (49%), Girls 27 (45%)	Life skills – Boys 31 (36%), Girls 23 (38%) Employment – Boys 9 (11%), Girls 3 (5%) University – Boys 2 (3%), Girls 0 (0%)
Thinking Boys 3 (4%), Girls 8 (13%)	Problem-solving – Boys 1 (1%), Girls 2 (3%) Creativity – Boys 3 (4%), Girls 6 (10%)
Not important Boys 2 (3%), Girls 12 (20%)	
Enjoyment Boys 4 (5%), Girls 4 (7%)	
Supports learning Boys 1 (2%), Girls 2 (3%)	

In the group interviews, the thematic keyword 'creativity' was attributed to statements from 4 girls. The identification related to the girls choosing a subject because they considered it to be creative and would give them balance. However, in one instance, Chae Eun mentioned that computer programming was more creative than she had previously thought. She said that she appreciated a creative outlet because she was not very good at drawing or acting.

5.4 Qualitative Results - Research Question 3 ('Are there any extrinsic factors that affect girls choosing to study computer programming in Year 10?')

This section of the qualitative findings is based on student responses in the group interviews and from the open-ended questions on the SPCS Questionnaire (Appendix M).

5.4.1 Influence of teachers

In an effort to explore what external influences affect students choosing Computer Science, participants were asked if teachers had any influence on their IGCSE choices. Janice's response was intriguing because she felt the grades that the teachers awarded influenced her decisions somewhat:

> "For me, teachers did not have a lot of influence, but the grades the teachers have given me before have influenced me because the grades prove that, no the grades doesn't prove a lot of things, but grades still tell me that I am good at this subject so I consider what grades I've got for a subject."

Both Sarah and Gina considered which teacher would be teaching them when choosing a particular subject. Gina commented that this is only related to Mandarin. She chose Mandarin because it was going to be taught by a different teacher and if her current teacher had been scheduled to be the IGCSE Mandarin teacher, then she would not have chosen it. Kay felt that the choice of subject was more important than which teacher was going to be delivering the subject. Bella and Senna also felt that the teachers did not influence their choices. The question of teachers was moot for Seo Yon, Chae Eun and Dain because all three were new to the school that year.

The findings from the interviews supported the quantitative findings that reported that teachers' influence on IGCSE choices were very slightly higher, or more influential, than parents but much less than lesson enjoyment. No students made a comment about the

Computer Science teachers, so there is little evidence of the teacher directly influencing students' decision to study Computer Science, or not.

5.4.2 Influence of parents

The discussion about parents should be viewed through the lens that the students' parents have a high social economic status and are all the same ethnicity. The students were not asked about their parents' background or status because these were group interviews and students are sensitive about commenting on social status in Korea. However, the school's fees are high so only wealthy parents could afford them.

During the group interviews, students were asked if their parents influenced their IGCSE choices. Since IGCSE options are limited, parents pushing a student to study something other than Computer Science may affect the number of girls choosing to continue with their Computer Science education. This response from Janice was, fairly, typical:

> 'My parents support my decision. They really did not influence my options choice because they support my decision, but because I was struggling between Art and Geography and I asked my mum whether I should take Art or Geography and she said that its she wants me to do Art, in her opinion, but she didn't force me to do art. She showed her opinion to me.'

However, earlier in the conversation Janice, who chose Mandarin as one of her IGCSE choices, commented that:

> 'And China is like increasingly getting more developed and so [my] parents and I thought speaking Chinese will help my future career and help me to be more successful in the future, so I chose Mandarin.'

So although Janice's parents were not directly telling her which subjects to take, they were involved in the decision-making process and were influencing her more than, perhaps, she realised. Senna was also influenced by her parents and encouraged to take Mandarin instead of Computer Science. Although, students could take Mandarin and Computer Science, to do so would mean that they couldn't take an artistic subject like art, dance, or drama. Other than

Janice and Senna, this pattern of parental influence was also observed in the conversations with Bella, Kay, Sarah, Gina, Chae Eun, Seo Yeon, Dain and Gina.

When Senna was asked the question “Do you think that, in general, students’ parents influence their IGCSE choices, or not?” she said “I think they are because the parents are the ones who have to sign the forms at the last date and if they are not happy they will change it”.

Senna had also commented that she originally chose Mandarin because her parents thought that it would be useful. She had wanted to study a subject related to mathematics or Science, such as Computer Science, but was unable to do so. However, after struggling with Mandarin she eventually convinced her parents to allow her to change her IGCSE from Mandarin to Computer Science. This process required Senna’s parents to email a signed subject transfer form to the Head of Curriculum at the school. In other discussions with students, there was a noticeable trend of parents persuading students to take Mandarin, and that Seo Yon’s parents were curious about her taking Computer Science, but subsequently agreed.

Of the 13 interviewed students only three felt that their parents had no bearing on their IGCSE choices. The quantitative findings showed that parents’ influence on IGCSE choices was very slightly lower, or less influential, than teachers and quite a bit higher than the influence of role models and friends. However, the qualitative findings suggest that parental influence is somewhat greater than the students reported in the questionnaire on factors that influence subject choices (Appendix N).

5.4.3 Influence of role models

As part of the IIC Questionnaire (Appendix N) on factors that influence subject choices, the students were asked the open-ended question, ‘If you were influenced by a role model who was it or were they?’ The majority of students left this question blank with only 22% of the girls and 38% of the boys writing anything. Of the 16 girls that wrote a comment, 6 of them were to say that they did not have a role model. Thus, 10 girls from the 74 female respondents identified a role model. Of the 10 role models, 3 were older students, 3 were female politicians (2 x Hillary Clinton, 1 x Angela Merkel), 2 were family members. Most importantly, none of the girls identified with female scientists.

Of the 36 boys that wrote a comment, 15 of them responded that they did not have a role model. Thus, 21 boys from the 94 male respondents identified a role model. 4 role models were older students, 5 were family members, 5 were celebrities, and 2 were teachers. Perhaps more noticeably, role models were not mentioned in the group interviews although there was no direct question about role models.

5.4.4 Influence of friends

When discussing IGCSE choices in the group discussions, friends were barely mentioned. It was thought that girls in the previous year group might not have chosen IGCSE Computer Science because their friends did not. Therefore, seven of the interviewed girls were asked if they chose the same IGCSE subjects as their friends and all of them said no. Thus, it would seem that choosing a subject because their friends approved or chose the same subject are low on the list of priorities and reflect that the influence of friends ranked last in the list of eight influences (i.e. parents, teacher, friends, life skills, university plans, lesson enjoyment, subject difficulty, role model).

5.4.5 Influence of future careers

The open-ended question 11 on the IIC Questionnaire (Appendix N) asked if there were any other influencing factors received few responses but, of those, 2 students commented that whether the subject was available at International Baccalaureate (IB) level affected their choice. Further, in the open-ended question 5 on the SPCS Questionnaire (Appendix M) participants were asked, "do you think that learning computer programming is an important activity or not". There was a notable gender difference in the number of students who did not think computer programming was important (1 boy versus 12 girls). Of those 12 girls that felt computer programming was not important, their explanations included 'it does not have any relationship to my career goal' and 'there is lots of people that you can get help from if you need help with computer science'. However, there were 3 girls (5% of those that responded) who commented that computer programming is important because of their career goals. This was in contrast to 9 boys (11% of those that responded) that felt computer programming was important to their future careers.

5.4.6 Other influencing factors

The open-ended question 11 on the questionnaire, about factors that influence subject choices, asked, '*Were there any other influencing factors and if so what were they?'*. The majority of students left this question blank with only 23% of the girls and 34% of the boys writing anything. Of the 23% female responses, the majority of the comments were “no” or repeated one of the options already rated. However, 3 respondents mentioned their grades influencing their decision. Of the 34% male responses, the majority repeated one of the eight influencing factors already rated. Only 1 respondent stated that their grades influenced their decision and 1 respondent gave a political response about the need to study Mandarin due to China’s economic strength.

Analysis of the group interview transcripts revealed the theme of ‘required’ and this theme related to the school policy of requiring a student to take either history or geography. The Korean Government also insists that Korean students study Korean in Year 10. One student mentioned that they only chose history because they had to choose a humanity subject and another student said that they were required to take Korean instead of choosing another subject.

5.5 Qualitative Results – Research Question 4 (‘Is there a significant gender difference in the preferred learning styles for computer programming?’)

Throughout the course of an academic school year, a variety of teaching and learning strategies were employed to determine which methods of teaching computer programming were most engaging and challenging for boys and girls. They included self-instructional tutorials with short problems, problem-based learning, storytelling, programming with visual design, game-based learning, diagrammatic learning, group work (collaborative learning and cooperative learning).

This research question was explored qualitatively by using two open-ended questions (Items 11 and 12) on the PCPLS Questionnaire (Appendix O). The first question listed the different ways that computer programming had been taught and asked students to choose which method (see table 5.4) was the most effective method and explain why it was effective. Of those that wrote answers, 70% (57) of the girls and 57% (64) of the boys did so, totalling 121 responders. Analysis of the responses indicated that of the listed teaching strategies there was a different focus between the genders. Table 5.4 shows the number of times a particular strategy was mentioned by boys and by girls, although many mentioned more than one strategy.

Table 5.5

Learning strategies mentioned in Question 11 on the Influences on IGCSE Choices Questionnaire

	N	% Girls	N	% Boys	N	% Total
Project / Problem-based	27	47%	12	19%	39	32%
Self-instructional tutorials	10	18%	18	28%	28	23%
Group work	9	16%	14	22%	23	19%
Visual design	7	12%	6	9%	13	11%
Storytelling	5	9%	4	6%	9	7%
Independent work	5	9%	3	5%	8	7%

The second open-ended question asked, *'Are there any other ways of learning that are used in other subjects, which might improve the way computing programming is taught?'* to see if a style of learning computer programming had been overlooked. Given that this was the last question it is not surprising that only 100 of the 194 students (52%) responded (41, or 51%, girls and 59, or 52%, boys). Many of the responses did not really answer the question, with responses such as "more time and care would be useful" and "make more interesting" proving unsuitable. Table 5.5 shows the number of times a particular theme was mentioned by boys and by girls that responded.

Table 5.6

Strategies for improving learning from Question 12 on the Influences on IGCSE Choices Questionnaire

	N	% Girls	N	% Boys	N	% Total
More group project work	9	22%	8	14%	17	17%
Positive about curriculum	8	20%	7	12%	15	15%
Teacher explanations	3	7%	1	2%	4	4%
Code help sheet	3	7%	0	0%	3	3%
More game-based learning	2	5%	5	8%	7	7%
Stream classes by ability	0	0%	2	3%	2	2%
Extra support (e.g. T.A.)	0	0%	4	7%	4	4%

5.5.1 Self-instructional tutorials

There was a noticeable difference in the responses to question 11 (*'Look at the following list of different ways we have learned computer programming, and explain which was the most effective method and why it was effective?')* on the PCPLS Questionnaire (Appendix O), at least in relation to self-instructional tutorials. Of the 64 boys and 57 girls that answered this question, 18 boys (28%) and 10 girls (18%) commented that self-instructional tutorials were beneficial (see Table 5.4). A boy (who was anonymous because of the setting on the online questionnaire) responded that he found the self-instructional tutorial useful because "I feel learning something each at a time helps me better to remember the method and the code". Another anonymous boy stated that he liked self-instructional tutorials because "We can basically straight away see how to do things". However, most responders simply identified it as a useful method and did not give a rationale as to why it was useful.

5.5.2 Problem-based learning / Project-based learning

The responses to question 11 (*'Look at the following list of different ways we have learned computer programming, and explain which was the most effective method and why it was effective.')* on the PCPLS Questionnaire (Appendix O) revealed that 27 girls (42% of the total girls), compared to 12 boys (19% of the total boys) stated the project work was a useful learning strategy (see Table 5.4). The project work in this curriculum incorporated problem-based learning, so when answering this question the students often referred to project work rather than problem-based learning. It was clear, however, that the students understood that project work incorporates problem-based learning with comments such as this anonymous response: "Project work because it engages us to think and inquire more, and try to solve some problems".

Some of the anonymous responses from the girls are shown below:

> "Project Work. I found working on a project was most effective because by putting efforts in and researching methods in the Internet, I could learn many new code words and functions."

> "Project work / Independent. We get to use the skills we have learned creatively and we have to really think about how the programme works instead of just copying what we learn, and we also have to learn new skills through extra research."

> "Project work - allows us to think creatively and explore various features of coding without set limitations. Works effectively when worked with friends as active discussions about the topic given is available and often lead to a better result."

> "When I do a project work it develops my skills more effectively since I actually research about it and use it - I can be independent. I can learn skills that I actually need and will use. Furthermore, because I build what I am interested in, it makes me have more interest in computer science."

The nature of the comments suggests that the students who enjoyed this method of learning computer programming found that the additional scope and creativity were the most appealing aspect of this learning style. A number of students also identified that this method involved thinking rather than simply copying code.

5.5.3 Computer programming with visual design

The responses to question 11 ('*Look at the following list of different ways we have learned computer programming, and explain which was the most effective method and why it was effective.')* on the PCPLS Questionnaire (Appendix O) revealed that 7 girls (12% of the total girls), compared to 6 boys (9% of the total boys), stated that programming with a graphical output was a preferred style of learning (see Table 5.4). Of those that did respond, two of the anonymous girls stated that the reason this was their preferred method of learning computing programming was "Because it helps us understanding by the result of our code easily" and "It was enjoyable being able to see graphically the work that I have produced with my code". Further, two students (1 boy and 1 girl) stated that they enjoyed using Python Turtle and 1 girl, but no boys said that they liked using flowcharts. However, it is difficult to draw conclusions from so few responses, especially when 30% of the girls and 43% of the boys left this open-ended question blank.

5.5.4 Working independently or in groups

The responses to question 11 ('*Look at the following list of different ways we have learned computer programming, and explain which was the most effective method and why it was effective.')* on the PCPLS Questionnaire (Appendix O) revealed that more boys than girls (22% boys versus 16% girls) preferred working in groups (see Table 5.4).

The rationale behind why boys and girls enjoyed this method of working tended to be similar. The anonymous comments of three boys included “Because we can learn more things from each other”, “a peer could point out one's weaknesses and strengths”, and “because I can check whether I am wrong or right or I can compare my computer programming skills”. The comments made by three anonymous girls included “because I like it <3 even though its hard there is friends so I can get help and can be cooperative ^^”, “because I learn from the others by getting more helps and it’s not that I only learn about computing but also about the teamwork”, and “because the students who are good at coding help me”. It would seem that both the boys and girls that enjoyed working with others did so because of the support and help received from their peers.

5.5.5 Computer programming with storytelling

Storytelling as a strategy for learning computer programming was mentioned by 9% of the girls and 6% boys in the open-ended question 11 on the PCPLS Questionnaire (Appendix O). A higher percentage of girls might have been expected given that girls, relative to boys, indicated a much greater liking for this strategy in the quantitative results. Further, the participants did not elaborate on why this was a preferred style of learning.

5.5.6 Game-based learning

Programming to create computer games was used in some of the self-instruction tutorials and students were given the option to incorporate puzzles in the storytelling modules. Consequently, students were specifically asked if this was a useful strategy and instead it was something that the interviewees brought up in the group interviews. For example, in response to the question, why do think there are more boys choosing Computer Science than girls, Eunice responded,

> “I am no sure but just in my opinion, I am just not interested into computers and stuff. I think that boys are more interested in playing computer games and living with their phones than girls.”

Eunice wasn’t alone in this assessment as both of the boys and another girl in a different interview expressed the opinion that because boys play more computer games than girls they are more interested in Computer Science.

5.5.7 Combination of learning strategies

Rather than focus on which style of learning best suits a particular gender, perhaps using a combination of learning strategies is more effective. An anonymous Year 9 girl believed this to be the case and wrote,

> "Tutorials and project work was the most effective method to me because by tutorials, I was able to know more codes like <p style= blabla > </p> and by doing projects, I was able to test my knowledge."

A considered answer from an anonymous Year 9 boy reasoned that a variety of techniques is the most suitable strategy, and stated:

> "All cover different aspects of Computer Science, so I suggest all are important and is useful if studied."

5.6 Summary

The qualitative findings for Research Question 1 did indicate that several girls believe boys are better at computer programming because of biological differences. The findings for Research Question 2 provided no evidence that girls lack confidence or have increased anxiety in relation to computer programming. The responses from the group interviews and the open-ended question did provide some evidence that boys are slightly more interested in computer programming than girls. There was no evidence that girls found computer programming any more difficult than boys, although many more boys, relative to girls, noted that mathematics was an important skill necessary for computer programming. The findings for Research Question 3 indicated that students' parents and teachers contributed to the choosing of IGCSE subjects but role models and friends had little influence. The IGCSE subject chosen by their friends had little influence on IGCSE choices. Career aspirations did affect students' perception of the importance of computer programming, with more boys than girls commenting that computer programming was important because of the employment opportunities. The findings for Research Question 4 indicated that boys enjoyed completing tutorials with short problems more than girls but that girls found group project work much more engaging than boys.

Chapter 6: Discussion

6.1 Introduction

Some critics of research on gender differences have suggested that by conducting the research itself there is a danger of ostracising one group. However, ignorance will not provide a solution to issues surrounded the gender disparity in the computing industry. Sensitivity to the potential dangers is paramount with consideration given to similarities and differences between the sexes. From this perspective, this research examined a large number of variables that could potentially affect the choices boys and girls make in choosing to study IGCSE Computer Science. If boys choose to continue their Computer Science education in Year 10 and girls do not, then this disparity is likely to continue at university and, ultimately, to the workplace.

This chapter is structured to answer the four research questions:

Research Question 1: Is there a significant gender difference in computer programming ability?

Research Question 2: Are there any intrinsic factors that affect girls choosing to study computer programming in Year 10?

Research Question 3: Are there any extrinsic factors that affect girls choosing to study computer programming in Year 10?

Research Question 4: Is there a significant gender difference in the preferred learning styles for computer programming?

6.2 Discussion - Research Question 1 ('Is there a significant gender difference in computer programming ability?')

In order to answer this research question, the programming test was designed to assess the participants' ability to apply programming knowledge in HTML/CSS and Python to particular programming problems. Thus, gender difference between the overall test scores of the girls and the boys were examined based on their performance in the test. There were concerns that if the girls lacked confidence in their programming ability, or perceived programming to be a boys' domain, then they would not be motivated to perform well in the test. There is evidence to suggest that this might have occurred in previous research, with Başer reporting that a statistically significant correlation was found between students' programming attitudes and their introductory programming course performance (Başer, 2013a), and Brauner, Leonhardt, Ziefle, and Schroeder (2010) reporting that Grade 7 students' self-reported confidence in technical ability negatively correlated with performance when programming robots. Another area of potential concern was that the programming assessment was carried out on a computer. PISA 2012 reported that boys, between the ages of 15 years 3 months and 16 years 2 months, tend to do better in both mathematics and reading when they take the test on a computer, rather than on paper (OECD, 2015). They conjectured that this is a by-product of boys' familiarity with computers through video games. Since the programming test used in this current research was conducted on a computer rather than on paper, it is possible that the form of assessment might have skewed the results in favour of boys.

Despite the aforementioned concerns, the results of the computer programming test in HTML/CSS and Python showed that the girls, compared to the boys, achieved significantly higher test scores. The test results used participants' Intelligent Quotient (IQ) scores as a covariate, which demonstrates that the difference between the boys' and girls' programming ability was not due to IQ differences. Thus, even controlling for the fact that girls' IQ scores were slightly higher than the boys, this study found that South Korean middle school girls outperformed boys of the same age in a computer programming assessment under examination conditions. This was unexpected because one of the few studies on programming ability, Rubio, Romero-Zaliz, Mañoso, and de Madrid (2015) reported that male students find programming easier and show higher learning outcomes than female students.

The second area of programming ability to be assessed was a gender comparison on those students in the top end of performance. This was accomplished by looking at the number of participants that scored 90%, or above, and completed the extension task, the 'Rainfall Problem'. The results showed that there was no significant gender difference in the number of students who achieved 90%, and above, and no significant gender difference in the number of students who completed the 'Rainfall Problem'. So, unlike mathematics, where boys have been shown to significantly outperform girls at the high end of achievement (Bergold, Wendt, Kasper, & Steinmayr, 2017; Stoet & Geary, 2013; Wai, Lubinski, Benbow, & Steiger, 2010), there was no difference in the most-able computer programmers.

It is impossible to say with complete certainty that the teaching of computer programming in a non-linear, traditional, way directly impacted upon girls' programming ability. However, in agreement with Buffum et al. (2015), it seems reasonable to suggest that the increased social element of programming, along with incorporating themes or topics that interested the girls, increased their engagement with computer programming. In turn, this is likely to have positively influenced the amount of time the girls spent programming and consequently improved their overall performance in the test. Students' grades in CS1 programming positively correlated with the decision to continue studying Computer Science (i.e. CS2) at Rutgers University (Babes-Vroman et al., 2017), so it seemed likely that the percentage of girls who chose IGCSE Computer Science would increase since their grades were higher than the boys. In reality, and after the teaching strategies were employed, the percentage of Year 9 girls who chose to study IGCSE increased from 13% to 38% of the female cohort for that year. Despite this, the percentage of girls was far smaller than the percentage of boys who chose to study IGCSE Computer Science. Therefore, this study reaches the same conclusion as Babes-Vroman et al. (2017) programming ability is not the only indicator of whether girls choose to continue studying Computer Science.

To further support these findings, potential confounding factors that might have influenced the test results are considered in the following section. These included pre-existing programming knowledge, external support through academies and tutors, more than one teacher delivering the curriculum, deliberately trying to increase the number of girls.

6.2.1 Potential confounding factors that might influence the test results

Social science research is invariably complex with numerous variables to consider. These variables may or may not impact upon the findings, and so it is important to acknowledge these potential confounding factors. Within this study, there are a number of areas that required additional investigation including pre-existing programming knowledge, external academic support, more than one teacher delivering the curriculum, and class sizes.

6.2.1.1 Pre-existing programming knowledge

One of the main problems of gender-based testing is that it assesses participants' ability at a particular time in their development, and usually when the participants have had different exposure to the subject for which they are being tested. For example, the majority of mathematics tests are influenced by participants' different experiences of this subject for numerous years prior to the testing. Unlike many other studies, virtually all participants had the same programming background, experienced the same pedagogical methods and were in the same environment because they had not studied programming in the Korean state system. Certainly, exposure to programming prior to studying a subject is likely to impact upon the results. Researching Malaysian undergraduates, Mohsin et al. (2010) found that prior programming knowledge was an effective predictor for high grades. Of the 194 students in this study, only four had ever experienced programming in a previous school, since computer programming is not taught in the South Korean state system. Of these 4 students, all were male and none were particularly accomplished at the start of the programme.

6.2.1.2 External support through academies and tutors

Another potential confounding influence is additional structured study outside of the classroom through the use of personal tutors, attending after-school classes, studying with parents and learning through online courses. In South Korea, parents often send their children to after-school educational institutes, with PISA reporting that, on average, boys spend 3.8 hours and girls spend 3.4 hours per week in such classes (OECD, 2015). Happily, the majority of the participants in this study are boarders and the parents of the day students are asked not to send their children to after-school classes because they have two hours of homework each day. Furthermore, there is no computer programming academy in the vicinity of the school so the impact of these confounders was minimised.

6.2.1.3 More than one teacher delivering the curriculum

The first year that computer programming was introduced to the school it was only taught by one teacher and there were only 40 minutes of computer programming per week. At the end of that year, 5 girls and 34 boys chose to study IGCSE Computer Science. It was this that prompted this research. In the second year, there were two teachers delivering the programme and there were 2 x 40 minutes of computer programming lessons. This may have influenced the results although every effort was made to ensure that all lessons were taught using the same strategies and resources. The two teachers taught an equal number of boys' and girls' classes to ensure that no bias would be observed. Further, most of the learning was student-centric and so it is unlikely that this influenced the results.

6.2.1.4 Deliberately trying to increase the number of Computer Science girls

There was a clearly defined agenda to make the subject more engaging for girls in order to increase the numbers at IGCSE. Despite this intention, boys were given the same opportunities as girls. This does not appear to have negatively impacted upon the number of boys taking IGCSE Computer Science because 54 boys (69%) chose to continue studying this subject.

6.2.1.5 Class size

Research into class size and achievement has been largely inconclusive (Schanzenbach, 2014) with Glass and Smith (1979) suggesting that this is largely due to overly selective literature searches, the difficulty of quantifying narrative and discursive reviews, and researchers' statistical mistakes when attempting to do so. Bressoux (2016) argues that the effects of class size appear to be modest at middle school and negligible at high school, whereas Biddle and Berliner (2011) state that extra gains from small classes are larger when the class has fewer than 20 students. In this study, all of the classes had 20 students or fewer and some of the girls' classes had as few as 12 students. With no clear evidence on the impact of small classes, it is assumed that this potential confounding factor had little impact.

Now that the results of the programming test and the potential confounding factors have been discussed, the next section considers other intrinsic factors that may influence the number of girls who choose to study Computer Science.

6.3 Discussion - Research Question 2 ('Are there any intrinsic factors that affect girls choosing to study computer programming in Year 10?')

In order to provide answers to Research Question 2 (*'Are there any intrinsic factors that affect girls choosing to study computer programming in Year 10?'*) a mixed methods approach was adopted and data was gathered on several factors, including programming confidence and anxiety, enjoyment, difficulty and perceptions.

6.3.1 Computer programming confidence and anxiety

Since students' confidence and anxiety are believed to be intrinsic factors that influence the pursuit of STEM careers (Rice et al., 2012), it was reasonable to believe that the gender disparity in choosing IGCSE Computer Science may be attributed to computer programming confidence levels or anxiety levels. However, the findings from the Computer Programming

Confidence and Anxiety (CPCA) Questionnaire and the analysis of the group interviews provided no significant evidence of a gender difference in confidence or anxiety. This differs from recent literature (e.g. Connolly, Murphy, & Moore, 2009; Sax et al., 2017) which has indicated that males have greater computer programming self-confidence than females, and Singh, Allen, Scheckler and Darlington (2007) who reported that anxiety towards computer programming contributed to the shortage of females in the Computer Science industry.

It has been shown that students' confidence levels are extremely important in determining performance outcomes. The large-scale PISA 2015 survey (OECD, 2016) reported that students with low confidence in science demonstrated lower science ability. Given that the 15-year old girls in this study outperformed the 15-year old boys in the programming assessment, and that the confidence and anxiety levels were assessed at the end of the year, it would have been contrary to expectations had the girls demonstrated a lower confidence level and high anxiety. It should be mentioned that the computer programming curriculum for the Year 8 and Year 9 students was designed to build confidence and reduce anxiety by scaffolding the learning and providing peer support.

As recommended by Margolis et al. (2003), there was careful consideration of topics and an attempt to use gender-neutral themes. For example, Buffum et al. (2015) expressed a concern that because girls on average have less prior experience with games, they may be disadvantaged if gaming is used as a teaching strategy. Therefore, programming games using code tended to incorporate puzzles, since this type of game appeals to girls (Scharkow, Festl, Vogelgesang, & Quandt, 2015). Outside of the deliberate strategies of storytelling and the use of visual design, there were several other strategies used by the Computer Science teachers to build girls' confidence in the classroom. For example, girls were set homework to design a poster of successful women in STEM and boys were set homework to design a poster of successful men in STEM. Some of the posters were displayed on the wall with an equal balance between successful male and female computing specialists. Retrospectively, had confidence and anxiety been assessed at the beginning of the course, instead of just at the end of the course, then the lack of gender difference could have been attributed to the teaching strategies. Since it was only assessed after the teaching strategies were delivered, the girls' confidence and lack of anxiety might not be attributed to the teaching strategies, even if this was the case. Future research could be conducted to determine if girls' confidence levels change after employing strategies to develop their interest in computer programming. It was decided that students' confidence and anxiety levels at the time of choosing their IGCSE subjects was most relevant to this research and that additional data gathering by questionnaire would be an imposition.

The number of girls, relative to boys, that chose IGCSE Computer Science improved after the teaching strategies were delivered but it still existed. Of the 78 IGCSE students, 23% were girls and 77% were boys. Thus, it would appear that neither computer programming anxiety nor confidence are the only contributing factors in choosing to continue studying computer programming.

6.3.2 Computer programming enjoyment

In a survey of 585 U.S. undergraduate students, 46% of the respondents reported that "expected enjoyment/excitement" was the number one criteria they used in career selection. This influencing factor, for both males and females, exceeded high earning potential, job security and helping others (Rowell et al., 2003). These findings are by no means isolated to the U.S., with 65% of international computer programming students reporting that they wanted to study Computer Science because of their interest in the subject (Craig, Paradis, & Turner, 2002).

This study found that boys enjoyed computer programming slightly, but not significantly, more than girls but that girls enjoyed computer programming more than most of their other subjects. Given these findings, it would be expected that more boys, compared to girls, would choose to study IGCSE Computer Science. Further, both genders identified subject enjoyment as the most important factor in choosing a subject. Therefore, it would also be expected that the number of girls choosing IGCSE Computer Science would increase from the baseline year, and before the teaching strategies were implemented. As previously stated, this was the case with 17 girls (38% of 45 girls) in Year 9, compared to 5 girls (13% of 38 girls) in the previous year, choosing to study IGCSE Computer Science as one of their two options. Additionally, the number of boys choosing to study IGCSE Computer Science also increased from 34 boys (52% of 66 boys) to 58 boys (69% of 84 boys). Since the different computer programming teaching strategies were not employed in the first year, logically it can be assumed that both boys and girls enjoy computer programming more when a variety of learning strategies was employed, as opposed to the self-instructional tutorial method used in the first year.

6.3.2.1 Computer gaming enjoyment

A lack of interest in computer gaming is seen as a significant concern for girls because of its links to computer literacy and comfort with technology (Pande, Weide, & Pande, 2016). Moreover, an increase in the amount of time spent playing PC/Mac computer games is linked to higher scores on a computer knowledge assessment (Appel, 2012). Since this study reported that girls spend far less time than boys playing computer games, this might account for some of the difference between the number of boys and the number of girls that chose IGCSE Computer Science.

Not only do the girls in this study spend less time playing computer games but they also choose to play different games from boys. Scharkow et al. (2015) reported that females favoured puzzle games, whereas males had a strong liking for strategy, sport, simulation, and action games. The participants in this study reported similar preferences, with girls favouring puzzle and simulation games and boys favouring online multiplayer games, which tend to be "first person shoot 'em-ups". Certainly, the differences of these responses should impact upon the subject matter when learning to program. Choosing to develop action-style computer games using applications and languages, like the programming language 'Unity', would be a good strategy for engaging boys. Conversely, a focus on creating action-style computer games would, perhaps, be a poor strategy for many girls, and instead developing a computer program that has educational or puzzle content is likely to be more appealing to girls.

Girls' perception of computer games was more negative than boys with more girls than not seeing it as a waste of time. Thus, encouraging the girls to program computer games may be perceived as a waste of time and reduce the appeal of learning to program. Since over 70% of girls responded that they spent between 0-20 minutes playing computer games, they would also be unlikely to spend additional time programming if the purpose was to create a computer game.

6.3.3 Computer programming difficulty

Computer programming is widely considered to be a cognitively challenging subject for novices to learn (Başer, 2013; Lahtinen, Ala-Mutka, & Järvinen, 2005; Mow, 2008). At an undergraduate level, computer programming courses report high failure rates due to the difficulty in writing code (e.g. Bennedsen & Caspersen, 2007; Haungs, Clark, Clements, &

Janzen, 2012; Kinnunen & Simon, 2012; Schäfer et al., 2013). Such is the difficulty of computer programming that even when students don't drop out, several final year students graduate without being able to write code (Carter & Jenkins, 2010).

It was considered that subject difficulty might account for the lack of girls choosing to continue studying Computer Science at IGCSE level. Further, the perception that computer programming is difficult develops mostly negative attitudes towards the subject (Başer, 2013) and so this may have accounted for girls not choosing to continue studying the subject. The findings from this study, however, indicate that male participants and female participants perceived the difficulty of computer programming similarly. Further, both genders place similar importance on subject difficulty when choosing IGCSE subjects. Consequently, there was no evidence that computer programming difficulty had any bearing on girls deciding to choose other IGCSE subjects instead of Computer Science.

6.3.4 Future plans

Many university students' subject choices are made on the basis for an improved standard of living (Cavus, Geri and Turgunbayeva; 2015) and career considerations may even influence GCSE choices (Warrington, Younger, & Williams, 2000). The findings indicate that many students considered what they wanted to do in the future when choosing to continue studying a subject at IGCSE level. Although both genders thought that learning to program a computer is a useful skill that will be beneficial later in life, boys expressed this more frequently than girls. Additionally, more boys than girls considered computer programming to be something that would be useful in their future careers.

There is evidence that students' career choices are influenced by parents with more than one student choosing IGCSE Mandarin because their parents believed that learning this language will be beneficial to the student's future employment. Since there are limited subject choices at IGCSE, students may not study computer programming even though they see the benefits. So, although future plans can be a contributing factor for choosing to study Computer Science, it is not the only consideration.

6.3.5 Perceptions of computer programmers

A high percentage of students are opposed to choosing to further their studies in Computer Science because they hold the perception that computer programmers sit in front of a screen all day (Carter, 2006) and that Computer Science is a masculine field (Master, Cheryan, and Meltzoff (2016); OCED, 2016). Further, if Computer Science educators have similar perceptions and assume that girls lack aptitude and motivation, then girls' own perceptions are unlikely to change (Pears et al., 2007).

Positive self-concept and other academic self-beliefs are important factors in choosing a career path (Bandura, 1986), and the positive relationship between self-concept and aspirations is supported in the literature (e.g. Lent, Brown, & Hackett, 1994; Multon, Brown, & Lent, 1991). Some of the girls in this study, however, appeared to believe that computer programming is a masculine discipline. Despite evidence to the contrary, they expressed a stereotypical belief that boys' brains are more suited to computer programming,

There was no evidence that girls perceived Computer Science to be a solitary profession and since the lessons were largely collaborative then there was little reason for them to do so.

6.4 Discussion - Research Question 3 ('Are there any extrinsic factors that affect girls choosing to study computer programming in Year 10?')

This section of the qualitative findings is based on student responses in the group interviews and from the open-ended questions on the SPCS Questionnaire (Appendix M).

It is difficult to entirely separate intrinsic and extrinsic influences because people's belief systems are undoubtedly influenced by their social interactions (Lent et al., 1994). Despite the awareness that students may not really know why they have the opinions they have, the participants in this study were questioned about influencing factors on their IGCSE choices.

The responses from the survey were extremely similar for both genders and indicated that intrinsic factors, which included lesson enjoyment, career and university aspirations, life skills, and subject difficulty were ranked more highly than external influencing factors in determining whether students chose to continue studying a particular subject. The external factors considered were teachers, parents, role models, and friends. The following section discusses the influence of these factors.

6.4.1 Influence of teachers

Hattie (2003) believes that of the major sources of variance associated with student achievement, the teacher accounts for about 30% of the variance. In short, it is what teachers know, do, and care about that makes such a significant difference. Roorda, Koomen, Spilt and Oort (2011) in an extensive meta-analysis demonstrated that students' motivation and their relationship with their teachers correlate strongly. When the male and female participants in this study both rated lesson enjoyment as the biggest influence on their IGCSE choices, it suggested that teachers influence their choices significantly, since it is they that plan and teach the lessons, and award the grades.

Some of the interviewed students suggested that teachers did not influence their subject selection, although it is likely that they were referring to direct influence rather than indirect influence through lesson enjoyment. It was considered that some students might choose a subject based on how much they like the teacher rather than how much they like the subject. However, of the three interviewed students who elaborated on teachers' influence, the discussions related more to the lesson enjoyment rather than any attachment to a particular teacher. In fact, teachers' influence on students tended to take a negative perspective, in that the student did not choose a subject because they did not like the teacher.

6.4.2 Influence of parents

Given that South Korea is a Confucian society it might be expected that parental expectations would significantly influence students' subject choices. In South Korea, the majority of parents believe that unselfish devotion and sacrifice to their children is their parental responsibility and this closeness instils a kind of indebtedness in their children, which drives academic aspirations and performance (Park & Kim, 2006). In a study assessing the values of children and the child-parent relationship, South Korean children were surveyed about the reasons for feeling indebtedness towards their parents, and the most frequent response was failing to meet expectations, following by failing to obey parents (Kim, Park, Kwon, & Koo, 2005). Moreover, South Korean adolescents view parental control as a sign of warmth and low neglect (Kim & Choi, 1994).

Despite the importance of family to South Koreans and the probable indebtedness towards their parents, the participants reported a parental influence mean of 3.37 for girls and a mean of 3.25 for boys on a Likert scale from 1 (least influence) to 7 (most influence). Of the eight factors that the participants were asked to rank as most influencing their IGCSE choices, parents rated more influential than friends and role models and about equivalent to teachers. However, students made numerous references to their parents in the group discussions, thereby suggesting that parental influence may be subtle and the student may not be overtly aware of just how much they are being influenced.

6.4.3 Influence of role models

Role models are individuals whose behaviours, personal styles, and specific attributes provide a template for emulation that is useful in achieving success (Shapiro, Haseltine, & Rowe, 1978). Further, females are more likely to be inspired by outstanding role models of the same sex (Lockwood, 2006). The literature provides many studies that support the view that female role models positively influence adolescent girls' career aspirations (e.g. Beaman, Duflo, Pande, & Topalova, 2012; Campbell & Wolbrecht, 2006; Lang, Craig, Fisher, & Forgasz, 2010; Shortland, 2014) and particularly in the pursuit of STEM careers (e.g. Drury et al., 2011; Ehrhart & Sandler, 1987). Of concern in this study is that both Computer Science teachers are male and consequently, a lack of role model or mentor may negatively affect the number of girls choosing to study IGCSE Computer Science. To provide some balance to this inequality, the female participants were taken off timetable in order to attend the four-part Cisco TV series 'Women Rock IT' (Cisco, 2015). Each episode lasted for 90 minutes and provided an interactive live stream of two successful women role models who work in the information technology industry.

Participants in this study were asked to rank their biggest influencing factors from a selection of eight (i.e. parents, teacher, friends, useful life skills, university/career aspirations, lesson enjoyment, subject difficulty, and role models) on their IGCSE choices. For both genders, role models were ranked seventh of the eight factors, with only friends being less important in their decision-making. A Likert scale of 1 to 7, inclusive, was used with 1 indicated extreme influence and 7 indicated no influence. The mean for role models was 5.09 for girls, 5.00 for boys compared to a mean of 2.57 for girls, 2.92 for boys for lesson enjoyment. It may be that role models are not important to these students, it may be that these students are not aware of the impact a role model has, or it may be they felt there was not a suitable role model. One thing is certain, there was no obvious difference in the girls' responses compared to the boys', suggesting that having two male teachers probably made little difference in this study.

6.4.4 Influence of friends

Adolescents often look to their friends to gauge what type of pursuits they should follow and if a student's friendship group values STEM, it may strengthen her or his interest and commitment to a STEM career path (Robnett & Leaper, 2013). However, in South Korea, there is a strong culture of expectation and pressure to achieve high academic performance (Lee et al., 2010). Consequently, many South Korean adolescents suffer from extreme academic stress due to excessive competition (Kim, Lee, Kim, Choi, & Lee, 2015). At parent-teacher conferences, parents commonly ask where their child ranks in their class and there is a general reluctant in the children to support each other.

The influence of friends and parents on career choices in international students was explored by Singaravelu, White, and Bringaze (2005) who reported that there was no significant difference in friends' influence on career choices between Asian and non-Asian students. They also reported that friends influence for both Asians and Non-Asians was significantly less than parental or family influence, so it was not surprising that participants of both genders rated friends' influences as the least important factor when choosing IGCSE options, with a mean of 2.62 for girls and 2.78 for boys, from a Likert scale of 1 to 7 inclusive. Of course, it may be that their friend's influences are more subtle than asking to rate the influence of friends on a Likert scale.

Overall, the extrinsic factors were virtually identical for both genders and ranked in the same order of lesson enjoyment, university/career plans, life skills, subject difficulty, teachers, parents, role models and then friends. Now that computer programming ability and potential influences have been considered, the impact of learning styles is discussed.

6.5 Discussion – Research Question 4 ('Is there a significant gender difference in the preferred learning styles?')

Over the years, researchers have developed as many as 70 different learning styles (Coffield, Moseley, Hall, & Ecclestone, 2004). Often, research into learning styles focuses on a particular style of learning for individuals and do not consider gender. Even when studies do make a comparison on gender, they often use different frameworks, making it difficult to compare results. For example, Kolb's experiential learning theory (Kolb, 2014) was applied to the process of learning computer programming (Buerck, Malmstrom, & Peppers, 2003) with the authors reporting that distance learners tended to use a *converger* learning style (a combination of active experimentation and abstract conceptualisation) when learning online and an *assimilator* learning style (a combination of reflective observation and abstract conceptualisation) when learning face-to-face. However, in this small-scale study with 33 participants, they acknowledged that additional research on gender and learning style is needed.

Numerous studies report that boys and girls have different preferences in their generic learning styles (Garland & Martin, 2005; Nuzhat, Salem, Hamdan, & Ashour, 2013; Ramayah, Sivanandan, Nasrijal, Letchumanan, & Leong, 2009; Wehrwein, Lujan, & DiCarlo, 2007) and the findings from this study indicated that boys and girls prefer to learn computer programming in slightly different ways. The following learning strategies indicate where the differences occurred in relation to learning computer programming.

6.5.1 Self-instructional tutorials

This strategy for learning computer programming is useful for identifying key concepts and learning them in isolation. The short problems at the end of the tutorials allow students to understand abstract concepts and adapt their programs experimentally, although this is only attempted after the tutorial has been completed. In relation to Kolb's (2014) learning style framework, this could be described as the *converger* learning style. The boys in this study enjoyed this method of learning more than the girls, and these findings are consistent with Fisher and Margolis (2003) who reported that male students view the computer as the object of study whereas girls associated their interest in computer programming as a tool for other disciplines. Given the importance of subject enjoyment in choosing to continue studying a subject, this strategy should not be used exclusively if girls are to be engaged in this subject.

6.5.2 Problem-based learning / Project-based learning

The literature (e.g. Ambrósio & Costa, 2010; Haas & Furman, 2011; Nuutila, Törmä, & Malmi, 2005; O'Kelly & Gibson, 2006) suggests that this method of learning is most successful for reducing dropout rates in undergraduate programmes and this study found that this style of learning was helpful in preventing students electing to stop studying Computer Science. The findings indicated that both genders enjoyed this method of learning, but that girls, in particular, found it much more engaging than the self-instructional tutorials. It can be surmised from the girls' feedback that learning to program a computer in order to complete a project was enjoyable. Certainly, this method provides greater opportunity for creativity and is less restrictive than the self-instructional tutorials that ask the student to solve specific tasks.

6.5.3 Computer programming with visual design

The findings in this study supported the expectation that if visual design (topics that, typically, interest girls) is included with learning to program a computer, then girls will be more engaged with the subject. This was expressed in the quantitative findings more than the qualitative findings, but that is more likely because the qualitative research did not explore this avenue directly. When choosing a computer programming language to deliver a computer programming syllabus it is important to choose a language that supports graphics.

6.5.4 Working independently or in groups

In a study investigating gender differences in asynchronous learning, Blum (1999) found that female students placed emphasis on relationships and preferred learning in a cooperative environment. Both genders seemed to enjoy working in groups more than working independently, with girls slightly, although not significantly, more so. The rationale they expressed was that having a friend to support learning and help fix errors was useful. This is unsurprising since it is in the nature of computer programming to have errors, or bugs, to fix. Given that these findings are similar to those reported by Liebenberg, Mentz and Breed (2012) in a small-scale study in Luxemburg, girls should be encouraged to learn computer programming in groups.

6.5.5 Computer programming with storytelling

The quantitative findings from the IIC Questionnaire (Appendix N) reported that girls, relative to boys, found programming with a storyline helpful when learning computer programming. The responses to the open-ended question on preferred learning strategies indicated that storytelling was not as important as the collaboration and cooperation aspect of the storytelling learning strategy. There are several studies (e.g. Denner et al., 2005; Kellerher, 2006; Kelleher & Pausch, 2007; Hu, 2008) that purport that storytelling is appealing to girls, but none of these studies used a text-based programming language. Instead, they used the highly visual programming environment, Alice, and it may have been the graphical element that was attractive to girls. The findings in this study focused more on creative writing by developing a text-based interactive story using Python and then using

HTML5/JavaScript, although students that completed their stories early were given permission to add images as an extension activity.

6.5.6 Game-based learning

Creating games rather than following the more traditional introduction to Computer Science is more likely to engage today's students (Kelleher & Pausch, 2007). Jenkins (2001) concurs and argues that the primary role of a computer-programming educator is that of a motivator and the role of communicating technical information is decidedly secondary. The findings in this study suggest that learning to program a computer game is perceived to be an effective strategy, but girls are more likely to be engaged if they are programming puzzle games.

6.5.7 Combination of learning strategies

Learning by self-instructional tutorial and then by completing short problem-solving tasks and learning by completing a project are both forms of constructivism, defined as knowledge that is actively constructed by the student (Ben-Ari, 2001). Both methods teach the student the abstract syntax of the programming language but then require the student to adapt the programs, which helps construct cognitive models of how the programming instructions work in combination.

There were a number of students who identified multiple ways of learning as an effective way of learning computer programming. Garretson, Krause-Phelan, Siegel, and Thelen, (2014) concur and suggested that using a variety of approaches is an effective strategy. The findings in this study, however, indicate that the majority of students preferred projects that incorporated problem-solving, but in an effort to reach all students a variety of learning strategies may ultimately prove the most beneficial. It is the amount of time devoted to each strategy that would need to be adjusted and incorporated. For example, self-instructional tutorials could be included at the start of a project.

Chapter 7: Conclusions

7.1 Introduction

The purpose of this research was to understand why so few middle school girls, relative to boys, choose to stop studying Computer Science and to determine the most effective learning strategies for engaging them. If the programme was effective then there should have been an increase in the number of Year 9 students choosing to continue their Computer Science education at IGCSE and, hopefully, beyond. This was the case with 17 girls (38% of the year group), compared to 5 girls (13%) in the previous year, choosing to continue their computer programming education by selecting IGCSE Computer Science as one of their two options. The number of boys choosing to study IGCSE Computer Science also increased from 34 boys (52%) to 58 boys (69%), so these strategies did not negatively affect boys' decisions to study IGCSE Computer Science.

Black and Wiliam (2003, p. 632) state that, "we believe strongly that the majority of research in education should be undertaken with a view to improving educational provision" and, with regards to the teaching of computer programming, there are several ways to achieve this. If programming is taught using strategies to engage girls, then girls' programming ability will be at least as proficient as boys. After all, "Aptitude knows no gender. Given equal opportunities, boys and girls, men and women have equal chances of fulfilling their potential" (OECD, 2015, p. 20). Educators who employ teaching interventions beneficial to both genders will give all students the opportunity to realise their potential.

7.2 Key Findings

This section outlines the key findings drawn from the investigation into the four research questions and is ordered by each question.

7.2.1 Key findings concerning Research Question 1 ('Is there a significant gender difference in computer programming ability?')

The findings reported that, of the students in the current sample, middle school girls scored significantly better on a programming test than middle school boys. However, there was no gender difference in those students who scored 90% or above in the programming test, or in those students who completed the extension question. These findings suggest that middle school girls are at least as competent as middle school boys at learning programming. Thus, it can be seen that a lack of programming ability is not a reason why more girls than boys decide not to continue studying Computer Science.

7.2.2 Key findings concerning Research Question 2 (‘Are there any intrinsic factors that affect girls choosing to study computer programming in Year 10?’)

The findings from the Computer Programming Confidence and Anxiety Questionnaire (Appendix D) reported no significant difference in either confidence or anxiety with regard to computer programming. Boys tended to report slightly more enjoyment than girls but this was not significantly different. Further, both genders reported high enjoyment of the computer programming compared to other subjects. Since lesson enjoyment was cited as the major reason why Year 9 students chose their IGCSE subjects, this factor may have influenced the number of students choosing to continue with Computer Science. It could not have been the only reason, however, since the number of girls relative to boys who chose IGCSE Computer Science was still much smaller.

Contrary to the evidence, some girls still have gender-biased stereotypes and believe that boys are more biologically suited to Computer Science than are girls. This is likely to have been a factor in why some girls who exhibited programming ability still chose not to study IGCSE Computer Science.

The literature suggests that undergraduate students find learning to programme challenging (e.g. Bennedsen & Caspersen, 2007; Kinnunen & Simon, 2012; Schäfer et al., 2013), yet this was not the case with the participants in this study. Both genders reported that learning computer programming was similar in difficulty to their other subjects.

Since one of the learning strategies employed to teach computer programming incorporated creating a computer game, computer gaming enjoyment was assessed and participants reported that boys spent considerably more time playing computer games than did girls. Moreover, boys preferred online multiplayer games, whereas girls preferred puzzle games. Thus, it would be productive to incorporate programming of puzzle games to increase girls’ enjoyment of computer programming

7.2.3 Key findings concerning Research Question 3 ('Are there any extrinsic factors that affect girls choosing to study computer programming in Year 10?')

Students, of both genders, rated extrinsic factors that influenced subject choices less important than any of the intrinsic factors. Data gathered from the online questionnaires and group interviews indicated that parents and teachers were the greatest extrinsic influences. In some instances, students were unaware of their parents' influence, but it became apparent from the language used in the group interviews. Certainly, some students chose subjects because their parents approved of those selections. There was no evidence that parents were encouraging their daughters to study Computer Science. Given that Shin et al. (2015) reported that parents' perceived science-related careers to be of lower importance for their daughters, this was not unexpected. The influence of teachers was evidenced through lesson enjoyment, which was rated as the most influential factor for choosing an IGCSE, via the online questionnaire, and in the group interviews where one student chose a subject because a particular teacher was teaching the class.

The other potential extrinsic motivating factors of future career opportunities - key skills, role models and friends - were considered to be of lesser importance.

7.2.4 Key findings concerning Research Question 4 ('Is there a significant gender difference in the preferred learning styles for computer programming?')

A variety of teaching strategies were employed to see if there was a gender difference in the preferred way of learning to program a computer. These strategies included self-instructional tutorials with short problems, problem-based learning, storytelling, programming with a visual design element, game-based learning, diagrammatic learning, group work (collaborative learning and cooperative learning). Inevitably, it was impractical to isolate these activities entirely and so a combination of activities was taught at a particular time.

The collated findings from the PCPLS Questionnaire (Appendix E) indicated that there were differences in the way male and female novice programmers preferred to learn computer programming. Most notable was that girls, relative to boys, found learning computer programming far more enjoyable when applied to a project that incorporated problem-solving skills as compared to completing self-instructional tutorials. The responses suggested that the underlying rationale for this preference was because they were required to research, inquire and think more than in the more-structured tutorials. Conversely, when learning computer programming, boys tended to have a slight preference for completing self-instructional tutorials rather than developing a project that incorporated problem-solving skills. Responses suggested the boys' slight preference for self-instructional tutorials was because the skills that they were learning were clearly defined and immediately apparent.

The second difference in the way novice programmers preferred to learn computer programming was that girls, relative to boys, preferred writing code that produced a visual output, like a pattern created through Python turtle or the Canvas tool in HTML5. The responses did not provide any explanation for why the girls' preferred programming that included a visual element but given that nearly three times as many girls, relative to boys, take GCSE Art and Design it is most likely because they enjoy visual design more than boys.

The third difference in the way novice programmers preferred to learn computer programming was that girls, relative to boys, preferred if the project incorporated creative writing through storytelling. This gender difference tended to be because boys preferred to complete self-instructional tutorials rather than undertake a project incorporating storytelling. Conversely, girls preferred to undertake a project that incorporated storytelling rather than complete self-instructional tutorials. The girls' responses indicated that they enjoyed the creativity of creating an interactive story and that having an example as a template helped.

7.3 Potential limitations of the current research

While every effort has been made to present these findings honestly and with integrity, this research cannot be said to be completely objective. Eisner (1992, p. 9) believes that "objectivity is one of the most cherished ideals of the education research community". Yet, the researcher, unashamedly, has had the goal of encouraging all students, irrespective of gender, to choose IGCSE Computer Science. Consequently, there is a vested interest in the outcomes of the learning strategies employed in this study. Whether this is an issue or not is subjective. After all, it is impossible for researchers to achieve objectivity because the personal experience of the researcher is an integral part of the research process (Ezzy, 2013).

One of the aims of this research was to uncover children's perceptions and to report the information that they perceive as central and not what they see as marginal. Interviewing children raises the question of whether they are telling the interviewer their opinions and beliefs or are they telling the interviewer what they think they want to hear. This is even more of a concern in this research due to the position of power in a teacher-student relationship (Miller, Mauthner, Birch, & Jessop, 2012). Consequently, the interviewer/researcher needs to be sensitive to what the participant is saying and guard against making interpretations due to personal bias and beliefs.

One of the most frequent issues within the field of educational research is that findings in one part of the world are assumed to apply in others (Walker & Dimmock, 2002). A 2008 survey of the best-known psychology journals found that 96% of the research subjects were from Western industrialised countries that account for approximately 12% of the world's population (Henrich, Heine, & Norenzayan, 2010). Furthermore, the overwhelming majority of the participants were psychology undergraduates because they were most readily available subjects. To generalise the findings from such a skewed population is clearly fraught with bias.

With this in mind, this is a relatively small-scale study, set in the context of one independent English-Curriculum school in South Korea, and thus the findings, on their own, may not generalise to other contexts. It is hoped that they will provide a useful insight into improving the Computer Science curriculum within the school. Further to that, this research will add to the body of knowledge about Computer Science, delivering insight into children's perceptions of programming and how these might differ by gender.

7.4 Implications for schools

The gender imbalance in Engineering and Computer Science is associated with the subject choices girls make at school (Kiwana, Kumar, & Randerson, 2011). If girls do not have a positive experience with STEM-related subjects then they could be inadvertently opting out of a STEM career. To address this problem the findings of this study relating to preferred learning style should be considered carefully. For academics wishing to deliver a suitable computer programming programme for girls, it would be appropriate to organise a Computer Science advisory group and task them with linking computer programming with female-dominated subjects, such as Health and Social Care and Home Economics. The advisory group should include expert teachers from these subjects and their brief would be to develop a creative programming curriculum that would teach the basics of computer programming in a way that girls find engaging. This should not be at the expense of boys, who should also be allowed to develop their computer programming skills using strategies that they find engaging.

The curriculum described above should also factor in gender equality and contain gender-neutral or gender-appropriate resources. The British Council, an organisation that specialises in international, cultural, and educational opportunities, has outlined an approach to teaching equality to boys and girls (Özdemir Uluç, 2017). This approach is to ensure that all trainee teachers are taught a module in gender issues with the underlying message that every child has the right to pursue any educational subject or interest. Trainee teachers should be taught about how people inadvertently reinforce gender stereotypes and encouraged to communicate to parents the importance of gender equality at home.

7.5 Implications for further research

One of the advantages of this study was that students were new to computer programming because at the time of data gathering it was not taught in the South Korean secondary school system. Only four students had done any programming before moving to the school where the study was conducted and none of them could program beyond a novice level. Consequently, there was no need to consider motivating students with a range of abilities.

Determining the best strategy with a range of abilities and still motivating exceptional students should be a consideration. One way of motivating and stretching the most-able novice programmers not considered in this study was by running a collaborative competition between four universities (White, Carter, Jamieson, Efford, & Jenkins, 2007). The authors concluded that this was a successful strategy for extending more-able novice programmers and a similar project with school children might also be a useful strategy and is an area that is recommended for further research.

To really understand the neurobiological implications of computer programming, there is an opportunity for a study to scan participants' brains using fMRI scans, while they are thinking in a computational way by solving computer programming problems. Biologically, it appears that areas of the left side of the brain responsible for language development develop first in girls, and the areas of the right side of the brain devoted to spatial skills develop first in boys (Njemanze, 2007; Shucard & Shucard, 1990). Since, PISA (OECD, 2015) reports that girls outperform boys in reading in all countries and economies by the equivalent of a school year, it seems reasonable to assume that biological differences in brain development are a contributing factor. What influence this has in learning to program a computer has received little, or no, research, but computer programming is a form of language.

7.6 Summary

This study demonstrated that creating a learning environment designed to support middle school girls learning computer programming resulted in three times as many girls, relative to the previous year, deciding to study IGCSE Computer Science. Lessons were planned to offer a variety of learning strategies, but most specifically incorporated collaborative problem-based projects. Moreover, the topics for these projects were gender neutral and included topics that the majority of girls found enjoyable, including visual design and storytelling. Under the aforementioned conditions, girls' programming ability surpassed that of boys, and the girls' self-confidence and anxiety did not differ significantly from boys'. Girls found the lessons following these curriculum strategies enjoyable and reported that lesson enjoyment was the most important factor in choosing to continue studying a subject in Year 10. The learning environment did not negatively affect boys, and 1.3 times as many boys, compared to the previous year, chose to continue studying Computer Science. Overall, there was a higher percentage of boys (77%), relative to girls (23%), that chose to study IGCSE Computer Science, but the participants suggested that boys' greater interest in computer games and girls' interest in creative subjects (e.g. art, design, dance, and music) exacerbated this differential. Thus, if computer programming teachers employ teaching and learning strategies that incorporate collaborative, problem-based projects with visual design or storytelling then this should help address the issue of girls dropping out of Computer Science programs in high school.

References

Ackermann, E., Gauntlett, D., Wolbers, T., & Weckstrom, C. (2009). Defining systematic creativity in the digital realm. *LEGO Learning Institute*. Retrieved May 7, 2017 from file:///Users/admin/Downloads/Systematic_Creativity_In_The_Digital_Realm_Report .pdf.

ACM (n.d.). Computer Science. Retrieved January 23, 2017 from http://computingcareers .acm.org/ ?page_id=8.

ACM/IEEE Computer Society (2013) Computer science curricula 2013: Curriculum guidelines for undergraduate degree programs in computer science. Retrieved on October 05, 2018 from http://www.acm.org /education/CS2013-final-report.pdf.

Admiraal, W., Huizenga, J., Heemskerk, I., Kuiper, E., Volman, M., & ten Dam, G. (2014). Gender-inclusive game-based learning in secondary education. *International Journal of Inclusive Education*, *18*(11), 1208-1218.

Ahmed, W., Minnaert, A., Kuyper, H., & van der Werf, G. (2012). Reciprocal relationships between math self-concept and math anxiety. *Learning and Individual Differences*, *22*(3), 385-389.

Albanese, M. A. (2010). *Problem-based learning.* In Jeffries, W. B., & Huggett, K. (Eds.) *An introduction to medical teaching* (pp. 41-53). New York, NY: Springer Science & Business Media.

Alderson, P., & Morrow, V. (2011). *The ethics of research with children and young people: a practical handbook*. London, UK: Sage Publications Ltd.

Alesandrini, K., & Larson, L. (2002). Teachers bridge to constructivism. *The Clearing House*, *75*(3), 118-121.

Allana, A., Asad, N., & Sherali, Y. (2010). Gender in academic settings: Role of teachers. *International Journal of Innovation, Management and Technology*, *1*(4), 343-348.

Altman, D. G., & Bland, J. M. (2009). Parametric v non-parametric methods for data analysis. *British Medical Journal*, *338*, 31-67.

Al-Tahat, K., Taha, N., Hasan, B., & Shawar, B. A. (2016, July). The impact of a 3d visual tool on female students attitude and performance in computer programming. In *SAI Computing Conference (SAI),* (pp. 864-867). London, UK: Institute of Electrical and Electronics Engineers.

Ambrósio, A. P. L., & Costa, F. M. (2010, March). Evaluating the impact of PBL and tablet pcs in an algorithms and computer programming course. In *Proceedings of the 41st ACM technical symposium on computer science education,* (pp. 495-499). Milwaukee, Wisconsin: Association for Computing Machinery

Amon, M. J. (2017). Looking through the glass ceiling: A qualitative study of STEM women's career narratives. *Frontiers in psychology*, *8*, 1-10.

An, J., & Park, N. (2011). Computer application in elementary education bases on fractal geometry theory using LOGO programming. In J.J. Park, H. Arabnia, H.B. Chang, & T.Shon (Eds.) *IT convergence and services* (pp. 241-249). Dordrecht, Netherlands: Springer,

Anastasiadou, S. D., & Karakos, A. S. (2011). The beliefs of electrical and computer engineering students' regarding computer programming. *International Journal of Technology, Knowledge & Society*, *7*(1), 37-51.

Anthes, G. (2012). HTML5 leads a web revolution. *Communications of the ACM*, *55*(7), 16-17.

Appel, M. (2012). Are heavy users of computer games and social media more computer literate? *Computers & Education*, *59*(4), 1339-1349.

Archer, L., DeWitt, J., Osborne, J., Dillon, J., Willis, B., & Wong, B. (2013). 'Not girly, not sexy, not glamorous': primary school girls' and parents' constructions of science aspirations 1. *Pedagogy, Culture & Society*, *21*(1), 171-194.

Archer, J., & Lloyd, B. (2002). *Sex and gender*. Cambridge, UK: Cambridge University Press.

Arjona-Reina, L., Robles, G., & Dueñas, S. (2014). The FLOSS2013 Free. *Libre/Open Source Survey*. Retrieved January 17, 2016 from https://floss2013.larjona.net/results.en.html.

Armoni, M., Meerbaum-Salant, O., & Ben-Ari, M. (2015). From scratch to "real" programming. *ACM Transactions on Computing Education (TOCE)*, *14*(4), 1-15.

Ashcraft, C. (2014). Increasing the participation of women and underrepresented minorities in computing: missing perspectives and new directions. Orlando, Florida: National Science Foundation Computer Science Education Summit. Retrieved May 12, 2016 from http://web.stanford.edu/~coopers/2013Summit/AshcraftCatherineColorado.pdf.

Azmat, G. (2014). Gender diversity in teams, *IZA World of Labor,* 29, 1–10.

Babes-Vroman, M., Juniewicz, I., Lucarelli, B., Fox, N., Nguyen, T., Tjang, A., ... & Chokshi, R. (2017, March). Exploring gender diversity in CS at a large public R1 research university. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education,* (pp. 51-56). Leeds, UK: ACM.

Banchefsky, S., Westfall, J., Park, B., & Judd, C. M. (2016). But you don't look like a scientist!: Women scientists with feminine appearance are deemed less likely to be scientists. *Sex Roles*, *75*(3-4), 95-109.

Bandura, A. (1986). *Social foundations of thought and action: a social cognitive theory*. Englewood Cliffs, London, UK: Prentice-Hall.

Bandura, A. (2001). Social cognitive theory of mass communication. *Media Psychology*, *3*(3), 265-299.

Banister, P. (2011). *Qualitative methods in psychology: a research guide*. Maidenhead, UK: Open University Press.

Baron-Cohen, S. (2004). *Essential difference: Male and female brains and the truth about autism*. New York, NY: Basic Books.

Basawapatna, A. R., Koh, K. H., & Repenning, A. (2010, June). Using scalable game design to teach computer science from middle school to graduate school. In *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education,* (pp. 224-228). Ankara, Turkey: Association for Computing Machinery.

Başer, M. (2013a). Attitude, gender and achievement in computer programming. *Online Submission*, 14(2), 248-255.

Başer, M. (2013b). Developing attitude scale toward computer programming. *International Journal of Social Science*, *6*(6), 199-215.

Bazeley, P. (2004). Issues in mixing qualitative and quantitative approaches to research. In R. Buber, J. Gadner, & L. Richards (Eds.) *Applying qualitative methods to marketing management research,* (pp. 141-156). Basingstoke, UK: Palgrave Macmillan.

BCS (2015). The women in IT scorecard - British Computer Society. Retrieved October 29, 2016 from http://www.bcs.org/upload/pdf/women-scorecard-2015.pdf.

Beaman, L., Duflo, E., Pande, R., & Topalova, P. (2012). Female leadership raises aspirations and educational attainment for girls: A policy experiment in India. *Science*, *335*(6068), 582-586.

Beaubouef, T., & Zhang, W. (2011). Where are the women Computer Science students? *Journal of Computing Sciences in Colleges*, *26*(4), 14-20.

Beckers, J. J., Wicherts, J. M., & Schmidt, H. G. (2007). Computer anxiety: “trait” or “state”? *Computers in Human Behavior*, *23*(6), 2851-2862.

Beckwith, J. (1983). Gender and math performance: Does biology have implications for educational policy? *Journal of Education*, 165(2), 158-174.

Beilock, S. L., & Carr, T. H. (2001). On the fragility of skilled performance: what governs choking under pressure? *Journal of Experimental Psychology: General*, *130*(4), 701-725.

Bejerano, A. R., & Bartosh, T. M. (2015). Learning masculinity: Unmasking the hidden curriculum in science, technology, engineering, and mathematics courses. *Journal of Women and Minorities in Science and Engineering*, *21*(2).

Ben-Ari, M. (2001). Constructivism in computer science education. *Journal of Computers in Mathematics and Science Teaching*, *20*(1), 45-74.

Benbow, C. P., & Stanley, J. C. (1980). Sex differences in mathematical ability: Fact or artifact? *Science*, *210*(4475), 1262-1264.

Bennedsen, J., & Caspersen, M. E. (2007). Failure rates in introductory programming. *ACM SIGCSE Bulletin*, *39(2),* 32-36.

BERA (2011). Ethical guidelines for educational research, BERA, London. Retrieved May 29, 2014 from http://www.bera.ac.uk/wp-content/uploads/2014/02/BERA-Ethical-Guidelines-2011.pdf.

Bergold, S., Wendt, H., Kasper, D., & Steinmayr, R. (2017). Academic competencies: Their interrelatedness and gender differences at their high end. *Journal of Educational Psychology*, *109*(3), 439-449.

Berry, M. (2013). *Computing in the national curriculum. A guide for primary teachers.* Computing at School and NAACE. Retrieved 03 July 2017 from www.computingatschool.org.uk/primary.

Bettinger, E. P., & Long, B. T. (2005). Do faculty serve as role models? The impact of instructor gender on female students. *American Economic Review*, *95*(2), 152-157.

Betz, D. E., & Sekaquaptewa, D. (2012). My fair physicist? Feminine math and science role models demotivate young girls. *Social Psychological and Personality Science*, *3*(6), 738-746.

Beyer, S., Rynes, K., Perrault, J., Hay, K., & Haller, S. (2003, February). Gender differences in computer science students. *ACM SIGCSE Bulletin*, *35(1)*, 49-53.

Biddle, B. J., & Berliner, D. C. (2011). Small class size and its effects. In Ballantine, J. H., & Spade, J. Z. (Eds.). *Schools and society: a sociological approach to education*. Los Angeles, (pp. 86-95). California: Pine Forge Press.

Black, P., & Wiliam, D. (2003). 'In praise of educational research': Formative assessment. *British Educational Research Journal*, *29*(5), 623-637.

Blum, K. (1999). Gender differences in asynchronous learning in higher education: Learning styles, participation barriers and communication patterns. *Journal of Asynchronous Learning, 3*(1), 46-66.

Blum, L., & Frieze, C. (2005). The evolving culture of computing: Similarity is the difference. *Frontiers: A Journal of Women Studies*, *26*(1), 110-125.

Bock, S. J., Taylor, L. J., Phillips, Z. E., & Sun, W. (2013). Women and minorities in computer science majors: results on barriers from interviews and a survey. *Women*, *14*(1), 143-152.

Bolliger, D. U., & Supanakorn, S. (2011). Learning styles and student perceptions of the use of interactive online tutorials. *British Journal of Educational Technology*, *42*(3), 470-481.

Bong, M., & Skaalvik, E. M. (2003). Academic self-concept and self-efficacy: How different are they really? *Educational Psychology Review*, *15*(1), 1-40.

Bosch, N., & D'Mello, S. (2013, July). Programming with your heart on your sleeve: Analyzing the affective states of computer programming students. In *International Conference on Artificial Intelligence in Education* (pp. 908-911). Heidelberg, Germany: Springer.

Boylan, M., & Willis, B. (2015). *Independent Study of Computing At School Master Teacher programme.* Sheffield, UK: Sheffield Hallam University. Retrieved July 08, 2018 from http://shura.shu.ac.uk/id/eprint/14886

Brannen, J. (2005). Mixing methods: The entry of qualitative and quantitative approaches into the research process. *International Journal of Social Research Methodology*, *8*(3), 173-184.

Braught, G., Wahls, T., & Eby, L. M. (2011). The case for pair programming in the computer science classroom. *ACM Transactions on Computing Education (TOCE)*, *11*(1), 2-21.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*(2), 77-101.

Brauner, P., Leonhardt, T., Ziefle, M., Schroeder, U., (2010). The effect of tangible artifacts, gender and subjective technical competence on teaching programming to seventh graders. In *Proceedings of the 4th International Conference on Informatics in Secondary Schools—Evolution and Perspectives* (vol. 5941, pp. 61–71). Heidelberg, Germany: Springer.

Bravo, C., Duque, R., & Gallardo, J. (2013). A groupware system to support collaborative programming: Design and experiences. *Journal of Systems and Software*, *86*(7), 1759-1771.

Bravo, C., Marcelino, M. J., Gomes, A. J., Esteves, M., & Mendes, A. J. (2005). Integrating educational tools for collaborative computer programming learning. *Journal of Universal Computer Science*, *11*(9), 1505-1517.

Brazier, M., & Cave, E. (2016). *Medicine, patients and the law*. Manchester, UK: Manchester University Press.

Bressoux, P. (2016). Research on class size in France. In *Blatchford, P., Chan, K. W., Galton, M., Lai, K. C., & Lee, J. C. Class size: Eastern and Western perspectives (pp. 80-91)*. London, UK: Routledge.

Brizendine, L. (2007). *The female brain*. London, UK: Bantam.

Broström, S. (2006). Care and education: Towards a new paradigm in early childhood education. *Child and Youth Care Forum*, 35, 391-409.

Brookshear, J. G., & Brylow, D. (2015). *Computer science: An overview*. Upper Saddle River, New Jersey: Pearson.

Broos, A. (2005). Gender and information and communication technologies (ICT) anxiety: Male self-assurance and female hesitation. *CyberPsychology & Behavior*, *8*(1), 21-31.

Brown, M. B., & Forsythe, A. B. (1974). Robust tests for the equality of variances. *Journal of the American Statistical Association*, *69*(346), 364-367.

Bruckman, A., Jensen, C., & DeBonte, A. (2002, January). Gender and programming achievement in a CSCL environment. In *Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community* (pp. 119-127). Boulder, Colorado: International Society of the Learning Sciences.

Bryman, A. (2012). *Social research methods* (4th ed.). Oxford, UK: Oxford University Press.

Buerck, J. P., Malmstrom, T., & Peppers, E. (2003). Learning environments and learning styles: Non-traditional student enrollment and success in an Internet-based versus a lecture-based computer science course. *Learning Environments Research*, *6*(2), 137-155.

Buffum, P. S., Frankosky, M., Boyer, K. E., Wiebe, E., Mott, B., & Lester, J. (2015, August). Leveraging collaboration to improve gender equity in a game-based learning environment for middle school computer science. In *Research in Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT), 2015* (pp. 1-8). Charlotte, North Carolina: Institute of Electrical and Electronics Engineers.

Burla, L., Knierim, B., Barth, J., Liewald, K., Duetz, M., & Abel, T. (2008). From text to codings: intercoder reliability assessment in qualitative content analysis. *Nursing research*, *57*(2), 113-117

Burrell, G., & Morgan, G. (2008). *Sociological paradigms and organisational analysis: elements of the sociology of corporate life*. Aldershot, UK: Ashgate.

Cameron, D. (2014). David Cameron: computer coding a priority for schools - video. Retrieved February 08, 2016 from http://www.theguardian.com/education/video/2014/dec/08/david-cameron-computing-coding-priority-schools-video.

Campbell, D. E., & Wolbrecht, C. (2006). See Jane run: Women politicians as role models for adolescents. *Journal of Politics*, *68*(2), 233-247.

Campbell, D. T., & Stanley, J. C. (2011). *Experimental and quasi-experimental designs for research*. Belmont, California: Central Learning.

Canli, T., Desmond, J. E., Zhao, Z., & Gabrieli, J. D. (2002). Sex differences in the neural basis of emotional memories. *Proceedings of the National Academy of Sciences, 99(*16)*,* 10789-10794.

Caplan, P. J., & Caplan, J. (2015). *Thinking critically about research on sex and gender*. London, UK: Taylor and Francis.

Carbonaro, M., Szafron, D., Cutumisu, M., & Schaeffer, J. (2010). Computer-game construction: A gender-neutral attractor to computing science. *Computers & Education*, *55*(3), 1098-1111.

Carlsen, B., & Glenton, C. (2011). What about N? A methodological study of sample-size reporting in focus group studies. *BMC medical research methodology*, *11*(1), 26-38.

Carson, D., Gilmore, A., Perry, C., & Gronhaug, K. (2001). *Qualitative marketing research*. London, UK: Sage Publications.

Carlson, S. (2006). Wanted: Female computer-science students. *The Chronicle of Higher Education*, *52*(19), 35-37. Retrieved December 12, 2015 from http://inthenews.unt.edu/sites/default/files/PDF/2006/1/13/01_13_2006_che_informat.pdf.

Carter, J. (2010). What are the important gender-related issues in computing at present? *Computer Science Education, 20*(4), 261-263.

Carter, J., & Jenkins, T. (2001). Arresting the decline: how can we encourage female students back into computer science. In *Proceedings of Higher Education Close Up 2 international research conference*. Lancaster: Higher Education Development Centre.

Carter, J., & Jenkins, T. (2002, June). Gender differences in programming. *ACM SIGCSE Bulletin, 34(3)*, 188-192.

Carter, J., & Jenkins, T. (2010). The problems of teaching programming: Do they change with time? In *11th Annual Conference of the Subject Centre for Information and Computer Sciences* (pp. 6-10). Ulster, UK: HE Academy.

Carter, L. (2006). Why students with an apparent aptitude for computer science don't choose to major in computer science. *ACM SIGCSE Bulletin*, *38(1),* 27-31.

Cavus, S., Geri, S., & Turgunbayeva, K. (2015). Factors affecting the career plans of university students after graduation. *International Journal of Humanities and Social Science*, 5(5) 94-95.

CAWMSET (2000) Land of plenty: Diversity as America's competitive edge in science, engineering and technology. *Report of the Commission on the Advancement of Women and Minorities in Science, Engineering and Technology Development.* Retrieved October 5, 2016 from *www. nsf. gov/pubs/2000/cawmset0409/cawmset _0409.pdf.*

Ceci, S. J., Williams, W. M., & Barnett, S. M. (2009). Women's underrepresentation in science: sociocultural and biological considerations. *Psychological Bulletin*, *135*(2), 218-261.

Cherney, I. D., & London, K. (2006). Gender-linked differences in the toys, television shows, computer games, and outdoor activities of 5-to 13-year-old children. *Sex Roles*, *54*(9-10), 717-726.

Cheryan, S., Drury, B. J., & Vichayapai, M. (2012). Enduring influence of stereotypical Computer Science role models on women's academic aspirations. *Psychology of Women Quarterly, 37,* 72–79.

Cheryan, S., Plaut, V. C., Davies, P. G., & Steele, C. M. (2009). Ambient belonging: How stereotypical cues impact gender participation in Computer Science. *Journal of Personality and Social Psychology*, *97*(6), 1045.

Cheryan, S., Siy, J. O., Vichayapai, M., Drury, B. J., & Kim, S. (2011). Do female and male role models who embody STEM stereotypes hinder women's anticipated success in STEM? *Social Psychological and Personality Science*, *2*(6), 656-664.

Cheryan, S., Ziegler, S. A., Montoya, A. K., & Jiang, L. (2017). Why are some STEM fields more gender balanced than others?. *Psychological Bulletin*, *143*(1), 1-35.

Cho, J., & Huh, J., (2017). *New Education Policies and Practices in South Korea*. Retrieved July, 30 from https://bangkok.unesco.org/content/new-education-policies-and-practices-south-korea.

Choi, J., An, S., & Lee, Y. (2015). Computing education in Korea—current issues and endeavors. *ACM Transactions on Computing Education (TOCE)*, *15*(2), 8-22.

Chope, R. C. (2005). Qualitatively assessing family influence in career decision making. *Journal of Career Assessment*, *13*(4), 395-414.

Chu, H. C., & Hwang, G. J. (2010). Development of a project-based cooperative learning environment for computer programming courses. *International Journal of Innovation and Learning*, *8*(3), 256-266.

CIE (2014). Syllabus: Cambridge IGCSE computer science 0478. Retrieved November 3, 2016 from http://www.cie.org.uk/images/167128-2016-syllabus.pdf.

Cisco (2015). Women rock-IT Cisco TV series. Retrieved February 21, 2016 from http://www.cisco.com/web/SG/partners/emails/woman_rock_it.html.

Claro, S., Paunesku, D., & Dweck, C. S. (2016). Growth mindset tempers the effects of poverty on academic achievement. *Proceedings of the National Academy of Sciences*, *113*(31), 8664-8668.

Cleaves, A. (2005). The formation of science choices in secondary school. *International Journal of Science Education*, *27*(4), 471-486.

Clegg, S. (2001). Theorising the machine: gender, education and computing. *Gender and Education*, *13*(3), 307-324.

Code Academy (n.d.). Retrieved August 08, 2014 from http://www.codecademy.com.

Coffield, F., Moseley, D., Hall, E., & Ecclestone, K. (2004). *Learning styles and pedagogy in post-16 learning. A systematic and critical review.* London: Learning and Skills Research Centre.

Cohen, L., Manion, L., & Morrison, K. (2013). *Research methods in education*. Abington, UK: Routledge.

Collins, S. (2016). *Neuroscience for learning and development: how to apply neuroscience and psychology for improved learning and training*. London, UK: Kogan Page.

Cooper, J., & Weaver, K. D. (2003). *Gender and computers: Understanding the digital divide.* Mahwah, New Jersey: Lawrence Erlbaum Associates.

Costa, J. M., & Miranda, G. L. (2017). Relation between Alice software and programming learning: A systematic review of the literature and meta-analysis. *British Journal of Educational Technology*, *48*(6), 1464-1474.

Cotter, D. A., Hermsen, J. M., Ovadia, S., & Vanneman, R. (2001). The glass ceiling effect. Social forces, 80(2), 655-681.

Cox, M., Webb, M., Abbott, C., Blakeley, B., Beauchamp, T., & Rhodes, V. (2015). *ICT and pedagogy: A review of the research literature.* London, UK: Department for Education and Skills.

Craig, A., Paradis, R., & Turner, E. (2002). A gendered view of computer professionals: preliminary results of a survey. *ACM SIGCSE Bulletin*, *34(2),* 101-104.

Creswell, J. W. (2014). *Research design: Qualitative, quantitative, and mixed methods approaches.* Thousand Oaks, California: Sage Publications.

Creswell, J. W., & Clark, V. L. P. (2011). *Designing and conducting mixed methods research*. Thousand Oaks, California: SAGE Publications.

Crews, T., & Butterfield, J. (2003a). Gender differences in beginning programming: an empirical study on improving performance parity. *Campus-Wide Information Systems*, *20*(5), 186-192.

Crews, T., & Butterfield, J. (2003b). Improving the learning environment in beginning programming classes: An experiment in gender equity. *Journal of Information Systems Education*, *14*(1), 69-76.

Crews Jr, T., & Ziegler, U. (1998, November). The flowchart interpreter for introductory programming courses. *Frontiers in Education Conference,* 1, 307-312.

Crotty, M. (1998). *The foundations of social research: Meaning and perspective in the research process*. London, UK: Sage.

Cruz-Duran, E. (2009). *Stereotype Threat in Mathematics: Female High School Students in All-Girl and Coeducation Schools*. Ann Arbor, Michigan: ProQuest LLC.

Cunningham, C. A. (2009). Transforming schooling through technology: Twenty-first-century approaches to participatory learning. *Education and Culture*, *25*(2), 46-61.

Cunningham, K. (2014). *Sams teach yourself Python: In 24 hours*. Indianapolis, Indiana: Sams.

Dale, N. B., & Lewis, J. (2016). *Computer science illuminated*. Burlington, Massachusetts: Jones & Bartlett Learning.

Dalton, G. W. (1989). Developmental views of careers in organisations. In M. B. Arthur, D. T. Hall & B. S. Lawrence (Eds.), *Handbook of career theory,* (pp.89-109). New York, NY: Cambridge University Press.

Dawson, C. (2009). *Introduction to research methods: A practical guide for anyone undertaking a research project*. New York, NY: Constable & Robinson.

Dawson, M. (2010). *Python programming for the absolute beginner*. Boston, Massachusetts: Course Technology Cengage Learning.

DeClue, T., Kimball, J., Lu, B., & Cain, J. (2011). Five focused strategies for increasing retention in Computer Science. *Journal of Computing Sciences in Colleges*, *26*(5), 252-258.

Dee, H. M., & Boyle, R. D. (2010, June). Inspiring women undergraduates. In *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education* (pp. 43-47). Ankara, Turkey: Association for Computing Machinery.

Deitel, P., & Deitel, H. (2014). *Java how to program-early objects*. Harlow, Essex: Pearson Education Limited.

Demetrion, G. (2004). Postpositivist scientific philosophy: Mediating convergences. Retrieved Jan 17, 2016 from http://library.nald.ca/item/5428.

den Brok, P., Levy, J., Brekelmans, M., & Wubbels, T. (2005). The effect of teacher interpersonal behaviour on students' subject-specific motivation. *The Journal of Classroom Interaction, 40*, 20–33.

Denner, J., Werner, L., Bean, S., & Campe, S. (2005). The girls creating games program: Strategies for engaging middle-school girls in information technology. *Frontiers: A Journal of Women Studies*, *26*(1), 90-98.

Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers & Education*, *58*(1), 240-249.

Denning, P. J. (2005). Is computer science science? *Communications of the ACM*, *48*(4), 27-31.

Denning, P. J. (2009). The profession of IT Beyond computational thinking. *Communications of the ACM*, *52*(6), 28-30.

Denscombe, M. (2008). Communities of practice a research paradigm for the mixed methods approach. *Journal of Mixed Methods Research*, *2*(3), 270-283.

Denzin, N. K. (2008). *Symbolic interactionism and cultural studies: The politics of interpretation*. Oxford, UK: Blackwell Publishers.

Derks, J., & Krabbendam, L. (2013). Is the Brain the Key to a Better Understanding of Gender Differences in the Classroom? *International Journal of Gender, Science and Technology*, *5*(3), 281-291.

Devsaran. (2015). 10 best programming languages of 2015: You should know. Retrieved February 09, 2016 from https://www.devsaran.com/blog/10-best-programming-languages-2015-you-should-know.

Dickey, M. D. (2003). Teaching in 3D: Pedagogical affordances and constraints of 3D virtual worlds for synchronous distance learning. *Distance Education*, *24*(1), 105-121.

Dillenbourg P. (1999) What do you mean by collaborative learning?. In P. Dillenbourg (Ed) *Collaborative-learning: Cognitive and Computational Approaches* (pp.1-19). Oxford, UK: Elsevier.

Dorling, M., & White, D. (2015, February). Scratch: A way to logo and python. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (pp. 191-196). Kansas City, Missouri: Association for Computing Machinery.

Draudt, D. (2016). The struggles of South Korea's working women. Retrieved December 24, 2016 from http://thediplomat.com/2016/08/the-struggles-of-south-koreas-working-women/.

Drury, B. J., Siy, J. O., & Cheryan, S. (2011). When do female role models benefit women? The importance of differentiating recruitment from retention in STEM. *Psychological Inquiry*, *22*(4), 265-269.

Dryburgh, H. (2000). Underrepresentation of girls and women in Computer Science: Classification of 1990s research. *Journal of Educational Computing Research*, *23*(2), 181-202.

DuBois, D. L., Portillo, N., Rhodes, J. E., Silverthorn, N., & Valentine, J. C. (2011). How effective are mentoring programs for youth? A systematic assessment of the evidence. *Psychological Science in the Public Interest*, *12*(2), 57-91.

Duch, B. J., Groh, S. E., & Allen, D. E. (Eds.). (2001). *The power of problem-based learning: a practical" how to" for teaching undergraduate courses in any discipline*. Sterling, Virginia: Stylus Publishing, LLC.

Durndell, A., & Haag, Z. (2002). Computer self efficacy, computer anxiety, attitudes towards the Internet and reported experience with the Internet, by gender, in an East European sample. *Computers in Human Behavior, 18*(5), 521-535.

Durrant, G. B. (2009). Imputation methods for handling item-nonresponse in practice: methodological issues and recent debates. *International Journal of Social Research Methodology*, 12(4), 293-304.

Dweck, C. (2010). Even geniuses work hard. *Educational Leadership*, *68*(1), 16-20.

Dweck, C. (2017). *Mindset-updated edition: Changing the way you think to fulfil your potential*. London, UK: Hachette Publishing Group.

Eagly, A. H., & Wood, W. (2013). The nature–nurture debates 25 years of challenges in understanding the psychology of gender. *Perspectives on Psychological Science*, *8*(3), 340-357.

Ehrhart, J., & Sandler, B. R. (1987). *Looking for more than just a few good women in traditionally male fields.* Washington, D.C.: Project on the Status and Education of Women.

Eide, B. J., & Winger, N. (2005). From the children's point of view: methodological and ethical challenges. In A. Clark, P. Moss, & A. T. Kjohoit (Eds) *Beyond listening: Children's perspectives on early childhood services*, (pp. 71-90). Bristol, UK: Policy Press.

Eisenkopf, G., Hessami, Z., Fischbacher, U., & Ursprung, H. W. (2015). Academic performance and single-sex schooling: Evidence from a natural experiment in Switzerland. *Journal of Economic Behavior & Organization*, *115*, 123-143.

Eisner, E. (1992). Objectivity in educational research. *Curriculum Inquiry*, *22*(1), 9-15.

Eliot, L. (2011). *Pink brain, blue brain: How small differences grow into troublesome gaps-and what we can do about it*. Richmond, UK: Oneworld Publications.

Else-Quest, N. M., Hyde, J. S., & Linn, M. C. (2010). Cross-national patterns of gender differences in mathematics: a meta-analysis. *Psychological Bulletin*, *136*(1), 103-127.

Emilson, A., Folkesson, A. M., & Lindberg, I. M. (2016). Gender beliefs and embedded gendered values in preschool. *International Journal of Early Childhood*, *48*(2), 225-240.

Ensmenger, N. (2012). *The computer boys take over: computers, programmers, and the politics of technical expertise*. Cambridge, Massachusetts: MIT Press.

Erceg-Hurn, D. M., & Mirosevich, V. M. (2008). Modern robust statistical methods: an easy way to maximize the accuracy and power of your research. *American Psychologist*, *63*(7), 591-601.

Ercikan, K., & Roth, W. M. (2006). What good is polarizing research into qualitative and quantitative? *Educational Researcher*, *35*(5), 14-23.

Erikson, E. H. (1993). *Childhood and society*. New York, NY: Norton.

Ernst, J. V., & Clark, A. C. (2012). Fundamental computer science conceptual understandings for high school students using original computer game design. *Journal of STEM Education: Innovations and Research*, *13*(5), 40-45.

ESA (2017). 2017 Sales, demographic, and usage data: Essential facts about the computer and video game industry. Retrieved August 22, 2018 from http://www.theesa.com/wp-content/uploads/2017/09/EF2017_Design_FinalDigital.pdf.

Esteves, M., Fonseca, B., Morgado, L., & Martins, P. (2011). Improving teaching and learning of computer programming through the use of the second life virtual world. *British Journal of Educational Technology*, *42*(4), 624-637.

Estrada, J. D., García-Ael, C., & Martorell, J. L. (2015). Gender differences in adolescents' choice of heroes and admired adults in five countries. *Gender and Education*, *27*(1), 69-87.

Ezzy, D. (2013). *Qualitative analysis: Practice and innovation*. London, UK: Routledge.

Feast, L. & Melles, G. (2010). Epistemological positions in design research: A brief review of the literature. In *2nd International Conference on Design Education* (pp. 1-4). Sydney, Australia: University of New South Wales.

Ferla, J., Valcke, M., & Cai, Y. (2009). Academic self-efficacy and academic self-concept: Reconsidering structural relationships. *Learning and Individual Differences*, *19*(4), 499-505.

Field, A. (2013). *Discovering statistics using IBM SPSS statistics*. London, UK: Sage.

Fisher, A., & Margolis, J. (2002). Unlocking the clubhouse: the Carnegie Mellon experience. *ACM SIGCSE Bulletin*, *34*(2), 79-83.

Flood, A. (2010). Understanding phenomenology: Anne Flood looks at the theory and methods involved in phenomenological research. *Nurse Researcher*, *17*(2), 7-15.

Forbes (2015). The world's most valuable brands. Retrieved February 21, 2016 from http://www.forbes.com/companies/cisco-systems/.

Fowler, A., Fristoe, T., & MacLaurin, M. (2012). Kodu game lab: A programming environment. *The Computer Games Journal*, *1*(1), 17-28.

Frampton, D., Blackburn, S. M., Cheng, P., Garner, R. J., Grove, D., Moss, J. E. B., & Salishev, S. I. (2009). Demystifying magic: high-level low-level programming. In *Proceedings of the International Conference on Virtual Execution Environments*, (pp. 81-90). Washington D.C.: Association for Computing Machinery.

Frankel, R. M., & Devers, K. J. (2000). Study design in qualitative research--1: Developing questions and assessing resource needs. *Education for Health*, *13*(2), 251-261.

Freitas, S. D., & Neumann, T. (2009). The use of 'exploratory learning' for supporting immersive learning in virtual environments. *Computers & Education*, *52*(2), 343-352.

Frenzel, A. C., Pekrun, R., & Goetz, T. (2007). Girls and mathematics—A "hopeless" issue? A control-value approach to gender differences in emotions towards mathematics. *European Journal of Psychology of Education*, *22*(4), 497-514.

Frigon, J. Y., & Laurencelle, L. (1993). Analysis of covariance: A proposed algorithm. *Educational and Psychological Measurement*, 53(1), 1-18.

Fuentes, A., Andersson, J., Johansson, A., & Nilsson, P. (2013). Gender and programming: A case study. In *Project and Conference Reports-Genombrottet.* Lunds Tekniska Högskola. Retrieved October 10, 2016 from https://www.lth.se/fileadmin/lth/genombrottet/konferens2005/Proc05/GenderProgramming.pdf.

Funder, D. C., Levine, J. M., Mackie, D. M., Morf, C. C., Sansone, C., Vazire, S., & West, S. G. (2014). Improving the dependability of research in personality and social psychology: Recommendations for research and educational practice. *Personality and Social Psychology Review*, *18*(1), 3-12.

Funke, A., Berges, M., Mühling, A., & Hubwieser, P. (2015, November). Gender differences in programming: research results and teachers' perception. In *Proceedings of the 15th Koli Calling Conference on Computing Education Research* (pp. 161-162). New York, NY: Association for Computing Machinery.

García-Peñalvo, F. J. (2016). What computational thinking is. *Journal of Information Technology Research 9*(3)*,* 5-8.

Garcia-Retamero, R., & López-Zafra, E. (2006). Prejudice against women in male-congenial environments: Perceptions of gender role congruity in leadership. *Sex Roles*, *55*(1-2), 51-61.

Garland, D., & Martin, B. N. (2005). Do gender and learning style play a role in how online courses should be designed? *Journal of Interactive Online Learning*, *4*(2), 67-81.

Garner, S. (2002). Reducing the Cognitive Load on Novice Programmers. In P. Barker & S. Rebelsky (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2002* (pp. 578-583). Chesapeake, Virginia: Association for the Advancement of Computing in Education.

Garson, G. D. (2012). *Testing statistical assumptions.* Asheboro, North Carolina: Statistical Associates Publishing.

Gaspard, H., Dicke, A. L., Flunger, B., Schreier, B., Häfner, I., Trautwein, U., & Nagengast, B. (2015). More value through greater differentiation: Gender differences in value beliefs about math. *Journal of Educational Psychology*, *107*(3), 663-704.

George, D., & Mallery, P. (2010). *SPSS for windows step by step. A simple guide and reference.* Boston, Massachusetts: Allyn and Bacon.

Gibson, D. E. (2004). Role models in career development: New directions for theory and research. *Journal of Vocational Behavior*, *65*(1), 134-156.

Giedd, J. N., Raznahan, A., Mills, K. L., & Lenroot, R. K. (2012). Review: magnetic resonance imaging of male/female differences in human adolescent brain anatomy. *Biology of Sex Differences*, *3(1),* 3-19.

Glass, G. V., & Smith, M. L. (1979). Meta-analysis of research on class size and achievement. *Educational Evaluation and Policy Analysis*, *1(1)*, 2-16.

Gomes, A., Carmo, L., Bigotte, E., & Mendes, A. (2006, September). Mathematics and programming problem solving. In *The 3rd E-Learning Conference–Computer Science Education* (pp. 1-5). Coimbra, Portugal: FCT.

Goode, J. (2008, March). Increasing Diversity in K-12 computer science: Strategies from the field. *ACM SIGCSE Bulletin, 40*(1), 362-366.

Gov.UK (2013a). National curriculum in England: Computing programmes of study. Retrieved September 24, 2016 from https://www.gov.uk/government/publications /national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study.

Gov.UK (2013b). Computing programmes of study: Key stages 3 and 4. Retrieved July, 5, 2018 from https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/239067/SECONDARY_national_curriculum_-_Computing.pdf.

Gov.UK (2013c). Computing programmes of study: Key stages 1 and 2. Retrieved October 2, 2016 from https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/239033/PRIMARY_national_curriculum_-_Computing.pdf.

Guba, E.G., & Lincoln, Y.S. (1994). Competing paradigms in qualitative research. In N.K. Denzin & Y.S. Lincoln (Eds.), *Handbook of qualitative research* (pp. 105-117). Thousand Oaks, California: Sage.

Gunderson, E. A., Ramirez, G., Levine, S. C., & Beilock, S. L. (2012). The role of parents and teachers in the development of gender-related math attitudes. *Sex Roles*, *66*(3-4), 153-166.

Guo, P. J. (2017, May). Older adults learning computer programming: motivations, frustrations, and design opportunities. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (pp. 7070-7083). Denver, Colorado: Association for Computing Machinery.

Gurian, M., Stevens, K., & Daniels, P. (2010). *Successful single-sex classrooms: A practical guide to teaching boys & girls separately*. San Francisco, California: John Wiley & Sons.

Haas, C., & Furman, N. (2011). "Operation Recreation: Adventure Challenge": Teaching Programming through Problem-Based Learning Theory. *SCHOLE*, *23*(1), 97-102.

Halpern, D. F. (2012). *Sex differences in cognitive abilities*. New York, NY: Psychology Press.

Halpern, D. F., Eliot, L., Bigler, R. S., Fabes, R. A., Hanish, L. D., Hyde, J., Liben, L., & Martin, C. L. (2011). The pseudoscience of single-sex schooling. *Science*, *333*(6050), 1706-1707.

Han, D. H., Bolo, N., Daniels, M. A., Arenella, L., Lyoo, I. K., & Renshaw, P. F. (2011). Brain activity and desire for Internet video game play. *Comprehensive Psychiatry*, *52*(1), 88-95.

Hanks, B., Fitzgerald, S., McCauley, R., Murphy, L., & Zander, C. (2011). Pair programming in education: a literature review. *Computer Science Education*, *21*(2), 135-173.

Harcourt, D. (2011). An encounter with children: Seeking meaning and understanding about childhood. *European Early Childhood Education Research Journal*, *19*(3), 331-343.

Harcourt, D., & Einarsdóttir, J. (2011). Introducing children's perspectives and participation in research. *European Early Childhood Education Research Journal*, *19*(3), 301-307.

Harmon-Jones, E., Price, T. F., & Gable, P. A. (2012). The influence of affective states on cognitive broadening/narrowing: considering the importance of motivational intensity. *Social and Personality Psychology Compass*, *6*(4), 314-327.

Harms, K. J., Rowlett, N., & Kelleher, C. (2015, October). Enabling independent learning of programming concepts through programming completion puzzles. In *Visual Languages and Human-Centric Computing (VL/HCC), 2015 IEEE Symposium on* (pp. 271-279).

Hartmann, T., & Klimmt, C. (2006). Gender and computer games: Exploring females' dislikes. *Journal of Computer‐Mediated Communication*, *11*(4), 910-931.

Hasan, B. (2003). The influence of specific computer experiences on computer self-efficacy beliefs. *Computers in Human Behavior*, *19*(4), 443-450.

Hattie, J.A.C. (2003, October). *Teachers make a difference: What is the research evidence?* Keynote presentation at the building teacher quality: The ACER annual conference, Melbourne, Australia. Retrieved January 29, 2017 from http://research.acer.edu.au/research_conference_2003/4/.

Hattie, J. (2013). *Visible learning: A synthesis of over 800 meta-analyses relating to achievement*. London, UK: Routledge.

Haungs, M., Clark, C., Clements, J., & Janzen, D. (2012, February). Improving first-year success and retention through interest-based CS0 courses. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 589-594). Raleigh, North Carolina: Association for Computing Machinery.

Henderson, P. B., De Palma, P., Almstrum, V. L., Hazzan, O., & Kihlstrom, K. P. (2002, February). Women, mathematics and computer science. *ACM SIGCSE Bulletin, 34*(1), 131-132.

Henrich, J., Heine, S. J., & Norenzayan, A. (2010). The weirdest people in the world? *Behavioral and Brain Sciences*, *33*(2-3), 61-83.

Herrmann, S. D., Adelman, R. M., Bodford, J. E., Graudejus, O., Okun, M. A., & Kwan, V. S. (2016). The effects of a female role model on academic performance and persistence of women in STEM courses. *Basic and Applied Social Psychology*, *38*(5), 258-268.

Hertz, M. (2010, March). What do CS1 and CS2 mean?: investigating differences in the early courses. In *Proceedings of the 41st ACM technical symposium on Computer science education* (pp. 199-203). Milwaukee, Wisconsin: Association for Computing Machinery.

Hetland, M. L. (2006). *Beginning Python: From novice to professional*. New York, NY: Apress Media.

Hewlett, S. A., Buck Luce, C., Servon, L. J., Sherbin, L., Shiller, P., Sosnovich, E., & Sumberg, K. (2008). *The Athena Factor: Reversing the brain drain in science, engineering and technology* (Harvard Business Review Research Report). Boston, Massachusetts: Harvard Business Publishing.

Higgins, M. C., & Kram, K. E. (2001). Reconceptualizing mentoring at work: A developmental network perspective. *Academy of Management Review*, *26*(2), 264-288.

Hill, C. (2016). Half of the high-paying jobs in America now require this skill. Retrieved September 24, 2016 from http://www.marketwatch.com/story/half-of-the-high-paying-jobs-in-america-now-require-this-skill-2016-06-21.

Hines, M. (2010). Sex-related variation in human behavior and the brain. *Trends in Cognitive Sciences*, *14*(10), 448-456.

Hitchcock, G., & Hughes, D. (1995). *Research and the teacher: A qualitative introduction to school-based research*. London: Routledge.

Hoare, C. A. R. (1969). An axiomatic basis for computer programming. *Communications of the ACM*, *12*(10), 576-580.

Hooshyar, D., Ahmad, R. B., Yousefi, M., Yusop, F. D., & Horng, S. J. (2015). A flowchart-based intelligent tutoring system for improving problem-solving skills of novice programmers. *Journal of Computer Assisted Learning*, *31*(4), 345-361.

Hopper, R. (2005). What are teenagers reading? Adolescent fiction reading habits and reading choices. *Literacy*, *39*(3), 113-120.

Horzum, M. B., & Cakir, O. (2009). The validity and reliability study of the Turkish version of the online technologies self-efficacy scale. *Educational Sciences: Theory and Practice*, *9*(3), 1343-1356.

Hothorn, T., Bretz, F., & Westfall, P. (2008). Simultaneous inference in general parametric models. *Biometrical journal*, *50*(3), 346-363.

Howard, K. A., Carlstrom, A. H., Katz, A. D., Chew, A. Y., Ray, G. C., Laine, L., & Caulum, D. (2011). Career aspirations of youth: Untangling race/ethnicity, SES, and gender. *Journal of Vocational Behavior*, *79*(1), 98-109.

Howell, K. E. (2013). *An introduction to the philosophy of methodology*. London, UK: Sage Publications.

Hu, H. H. (2008). A summer programming workshop for middle school girls. *Journal of Computing Sciences in Colleges*, *23*(6), 194-202.

Hubwieser, P. (2012). Computer science education in secondary schools - the introduction of a new compulsory subject. *Transactions on Computing Education*. 12(4), 16-41.

Hudson, L. A., & Ozanne, J. L. (1988). Alternative ways of seeking knowledge in consumer research. *Journal of Consumer Research*, *14*(4), 508-521.

Hung, C. M., Hwang, G. J., & Huang, I. (2012). A Project-based Digital Storytelling Approach for Improving Students' Learning Motivation, Problem-Solving Competence and Learning Achievement. *Educational Technology & Society*, *15*(4), 368-379.

Hwang, G. J., Wu, P. H., & Chen, C. C. (2012). An online game approach for improving students' learning performance in web-based problem-solving activities. *Computers & Education*, *59*(4), 1246-1256.

Hyde, J. S. (2005). The gender similarities hypothesis. *American Psychologist*, *60*(6), 581-592.

IBO.org (2014). Computer science in DP | International Baccalaureate®. Retrieved November 12, 2016 from http://www.ibo.org/programmes/diploma-programme/curriculum/sciences/computer-science/.

Ibrahim, R., Yusoff, R. C. M., Omar, H. M., & Jaafar, A. (2010). Students' perceptions of using educational games to learn introductory programming. *Computer and Information Science*, *4*(1), 205-216.

Ingalhalikar, M., Smith, A., Parker, D., Satterthwaite, T. D., Elliott, M. A., Ruparel, K., ... & Verma, R. (2014). Sex differences in the structural connectome of the human brain. *Proceedings of the National Academy of Sciences*, *111*(2), 823-828.

Institute of International Education. (2015). "Top 25 places of origin of international students, 2013/14-2014/15." *Open Doors Report on International Educational Exchange*. Retrieved September 25, 2016 from http://www.iie.org/opendoors.

International Baccalaureate (2017). The conduct of IB Diploma Programme examinations: May and November 2017. Retrieved October 15, 2018 from https://www3.dpcdsb.org/STFXS/Documents/2017_The_conduct_of_IB_Diploma_Programme_examinati.pdf.

International Society for Technology in Education (2016). ISTE standards for students. Retrieved September 24, 2016 from http://www.iste.org/standards/standards/for-students-2016.

Iwamoto, D. H., Hargis, J., Taitano, E. J., & Vuong, K. (2017). Analyzing the Efficacy of the Testing Effect Using Kahoot™ on Student Performance. *Turkish Online Journal of Distance Education*, *18*(2), 80-93.

Jacob, S. R., & Warschauer, M. (2018). Computational Thinking and Literacy. *Journal of Computer Science Integration*, *1*(1), 1-19.

Jang, I. O., & Lew, H. C. (2011). Case studies in thinking processes of mathematically gifted elementary students through Logo programming. *Work*, *4*, 9.

Jayal, A., Lauria, S., Tucker, A., & Swift, S. (2011). Python for teaching introductory programming: A quantitative evaluation. *Innovation in Teaching and Learning in Information and Computer Sciences*, *10*(1), 86-90.

Joint Council for Qualifications (2016a). A, AS and AEA results, Summer 2016. Retrieved May 4, 2017 from https://www.jcq.org.uk/examination-results/a-levels/2016/a-as-and-aea-results

Joint Council for Qualifications (2016b). GCSE and entry level certificate results. Retrieved September 4, 2016 from http://www.jcq.org.uk/examination-results/gcses/2016/gcse-and-entry-level-certificate-results-summer-2016.

Jenkins, T. (1998). A participative approach to teaching programming. In *Proceedings of the 3rd Annual Conference on Innovation and Technology in Computer Science Education*, (pp. 125–129). Leeds, UK: ACM Press.

Jenkins, T. (2001, July). Teaching programming-a journey from teacher to motivator. In *The 2nd Annual Conference of the LSTN Center for Information and Computer Science*. Retrieved December 11, 2016 from https://www.cs.kent.ac.uk/ people/staff/saf/dc/portfolios/tony/doc/other/motivation.pdf.

Johansen, A. (2016). *Python: The ultimate beginner's guide!* London, UK: CreateSpace Independent Publishing Platform.

Johnson, R. B., & Onwuegbuzie, A. J. (2004). Mixed methods research: A research paradigm whose time has come. *Educational Researcher*, *33*(7), 14-26.

Johnson, R. B., Onwuegbuzie, A. J., & Turner, L. A. (2007). Toward a definition of mixed methods research. *Journal of Mixed Methods Research*, *1*(2), 112-133.

Jones, J., & Monaco, J.(n.d.) Problem-based learning/Project-based learning. Retrieved July 5, 2017 from http://hbotoolkit.laureate.net/wp-content/uploads/2015/09/21-PBL_Combined_reviewapproved.engl-EY-QA-xxx.CW-Layout.Engl_.pdf.

Kabeer, N., & Natali, L. (2013). Gender Equality and Economic Growth: Is there a Win-Win? *IDS Working Papers*, *2013*(417), 1-58.

Kafai, Y., Heeter, C., Denner, J., & Sun, J. (2008). Pink, purple, casual, or mainstream games: Moving beyond the gender divide. In Y. Kafai, C. Heeter, J. Denner & J. Sun (Eds.), Beyond Barbie and Mortal Kombat: *New perspectives on gender and gaming* (pp. xi-xxv). Cambridge, Massachusetts: Massachusetts Institute of Technology (MIT) Press

Kahn, R., & Kellner, D. (2005). Reconstructing technoliteracy: A multiple literacies approach. *E-Learning and Digital Media*, *2*(3), 238-251.

Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code. org. *Computers in Human Behavior*, *52*, 200-210.

Kallet, R. H. (2004). How to write the methods section of a research paper. *Respiratory Care*, *49*(10), 1229-1232.

Kaplan, R. M. (2010, October). Choosing a first programming language. In *Proceedings of the 2010 ACM conference on Information technology education* (pp. 163-164). Midland, Michigan: Association for Computing Machinery.

Kapp, K. M. (2012). *The gamification of learning and instruction: game-based methods and strategies for training and education*. San Francisco, California: John Wiley & Sons.

Karpiak, C. P., Buchanan, J. P., Hosey, M., & Smith, A. (2007). University students from single-sex and coeducational high schools: Differences in majors and attitudes at a catholic university. *Psychology of Women Quarterly*, *31*(3), 282-289.

Kaushanskaya, M., Gross, M., & Buac, M. (2013). Gender differences in child word learning. *Learning and Individual Differences*, *27*, 82-89.

Kay, J., Barg, M., Fekete, A., Greening, T., Hollands, O., Kingston, J. H., & Crawford, K. (2000). Problem-based learning for foundation computer science courses. *Computer Science Education*, *10*(2), 109-128.

Kay, J. S., & Lauwers, T. (2013). Robotics in computer science education. *Computer Science Education*, *23*(4), 291-295.

Kay, J. S., & Moss, J. G. (2012, October). Using robots to teach programming to K-12 teachers. In *Frontiers in Education Conference (FIE), 2012* (pp. 1-6). Charlotte, North Carolina: Institute of Electrical and Electronics Engineers.

Kebritchi, M. (2008). Examining the pedagogical foundations of modern educational computer games. *Computers & Education*, *51*(4), 1729-1743.

Kelleher, C. (2006). *Motivating Programming: Using storytelling to make computer programming attractive to middle school girls.* Unpublished Technical Report, Carnegie Mellon University. Retrieved October 12, 2016 from http://www.dtic.mil/get-tr-doc/pdf?AD=ADA492489.

Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR)*, *37*(2), 83-137.

Kelleher, C., & Pausch, R. (2007). Using storytelling to motivate programming. *Communications of the ACM*, *50*(7), 58-64.

Kelleher, C., Pausch, R., & Kiesler, S. (2007, April). Storytelling Alice motivates middle school girls to learn computer programming. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1455-1464). New York, NY: Association for Computing Machinery.

Kelly, K., Dampier, D. A., & Carr, K. (2013). Willing, able, and unwanted: high school girls' potential selves in computing. *Journal of Women and Minorities in Science and Engineering*, *19*(1), 67-85.

Kemeny, J. G., & Kurtz, T. E. (1980). *Basic programming*. New York, NY: John Wiley & Sons.

Keppel, G. & Wickens, T. D., (2004). *Design and analysis: A researcher’s handbook.* Englewood Cliffs, New Jersey: Prentice Hall.

Khayrallah, M. A., & Van Den Meiraker, M. (1987). LOGO programming and the acquisition of cognitive skills. *Journal of Computer-Based Instruction*. 14(4), 133-137.

Kim, B., Lee, M., Kim, K., Choi, H., & Lee, S. M. (2015). Longitudinal analysis of academic burnout in Korean middle school students. *Stress and Health*, *31*(4), 281-289.

Kim, J., Lee, J.W., & Shin, K. (2014). *Gender inequality and economic growth in Korea.* Seoul: Korea University Press. Retrieved January 22, 2017 from http://econ.korea.ac.kr/~jwlee/papers/Gender%20and%20Korea%20KLS.pdf.

Kim, K., & Lowry, D. T. (2005). Television commercials as a lagging social indicator: Gender role stereotypes in Korean television advertising. *Sex roles*, *53*(11-12), 901-910.

Kim, U., & Choi, S. (1994). Individualism, collectivism, and child development: A Korean perspective. In P. M. Greenfield and R. R. Cocking (Eds.), *Cross-cultural roots of minority child development* (pp. 1–37). Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Kim, U., Park, Y. S., Kwon, Y. E., & Koo, J. (2005). Values of children, parent–child relationship, and social change in Korea: Indigenous, cultural, and psychological analysis. *Applied Psychology*, *54*(3), 338-354.

Kim, Y. H., O'Brien, K. M., & Kim, H. (2016). Measuring Career Aspirations Across Cultures: Using the Career Aspiration Scale With Young Korean Women. *Journal of Career Assessment*, *24*(3), 573-585.

King, J., Bond, T., & Blandford, S. (2002). An investigation of computer anxiety by gender and grade. *Computers in Human Behavior*, *18*(1), 69-84.

King, N., & Horrocks, C. (2016). *Interviews in qualitative research*. Los Angeles, California: Sage.

Kinnunen, P., & Simon, B. (2012). My program is ok–am I? Computing freshmen's experiences of doing programming assignments. *Computer Science Education*, *22*(1), 1-28.

Kiwana, L., Kumar, A., & Randerson, N. (2011). An investigation into why the UK has the lowest proportion of female engineers in the EU. *Engineering UK*. Retrieved November 10, 2016 from http://www.engineeringuk.com/_resources/documents/Int_Gender_summary_EngineeringUK_04_11_.pdf.

Klawe, M. (2016, April 02). How can we encourage more women to study computer science? *Newsweek*. Retrieved August 22, 2017 from http://www.newsweek.com/how-can-we-encourage-more-women-study-computer-science-341652.

Kolata, G. B. (1980). Math and sex: Are girls born with less ability? *Science*, *210*(4475), 1234-35.

Kolb, D. A. (2014). *Experiential learning: Experience as the source of learning and development*. London, UK: Pearson Education.

Kolbe, R. H., & Burnett, M. S. (1991). Content-analysis research: An examination of applications with directives for improving research reliability and objectivity. *Journal of Consumer Research*, *18*(2), 243-250.

Korea Herald. (2014, July 27). [Editorial] Coding in schools. Retrieved September 23, 2018 from http://www.koreaherald.com/view.php?ud=20140727000121.

Kothari, C. R., 2009. *Research methodology: Methods and techniques*. Delhi, India: New Age International.

Kulturel-Konak, S., D'Allegro, M. L., & Dickinson, S. (2011). Review of gender differences in learning styles: Suggestions for stem education. *Contemporary Issues in Education Research*, *4*(3), 9-18.

Kvale, S. (1996). *Interviews: An introduction to qualitative research interviewing*. Thousand Oaks, California: Sage Publications.

Lahtinen, E., Ala-Mutka, K., & Järvinen, H. M. (2005, June). A study of the difficulties of novice programmers. *ACM SIGCSE Bulletin, 37*(3), 14-18.

Lang, C. (2010). Happenstance and compromise: a gendered analysis of students' computing degree course selection. *Computer Science Education*, *20*(4), 317-345.

Lang, C., Craig, A., Fisher, J., & Forgasz, H. (2010, June). Creating digital divas: scaffolding perception change through secondary school and university alliances. *In Proceedings of the fifteenth annual conference on Innovation and technology in computer science education* (pp. 38-42). Leeds, UK: Association for Computing Machinery.

Larkin, T. L., & Quinn, C. M. (2010, October). The feminine side of engineering: It's way more than just "Girl Talk!". In *Frontiers in Education Conference (FIE), 2010 IEEE* (pp. F4E-1). Washington, DC: Institute of Electrical and Electronics Engineers.

Lauzen, M. M., Dozier, D. M., & Horan, N. (2008). Constructing gender stereotypes through social roles in prime-time television. *Journal of Broadcasting & Electronic Media*, *52*(2), 200-214.

Laursen, S., Hunter, A. B., Seymour, E., Thiry, H., & Melton, G. (2010). *Undergraduate research in the sciences: Engaging students in real science*. San Francisco, California: John Wiley & Sons.

Leaper, C. (2013). Parents' socialization of gender in children. In C. Martin (Ed.) *Gender: Early socialization*, (pp. 6-9). Retrieved June 18, 2017 from http://www.child-encyclopedia.com/sites/default/files/dossiers-complets/en/gender-early-socialization.pdf#page=6.

Lee, A. (2015). Determining the effects of computer science education at the secondary level on STEM major choices in postsecondary institutions in the United States. *Computers & Education*, *88*, 241-255.

Lee, J., Puig, A., Kim, Y. B., Shin, H., Lee, J. H., & Lee, S. M. (2010). Academic burnout profiles in Korean adolescents. *Stress and Health*, *26*(5), 404-416.

Legewie, J. and DiPrete, T. (2012). *High School Environments, STEM orientations, and the gender gap in science and engineering degrees.* New York, NY: Columbia University. Retrieved December 18, 2017 from http://www.columbia.edu/~tad61/paper_pathway_07172011.pdf.

Lehrer, R., Randle, L., & Sancilio, L. (1989). Learning preproof geometry with LOGO. *Cognition and Instruction*, *6*(2), 159-184.

Leininger, M. M. (1985). Ethnography and ethnonursing: Models and modes of qualitative data analysis. In M. M. Leininger (Ed.), *Qualitative research methods in nursing* (pp. 33-72). Orlando, Florida: Grune & Stratton.

Lenroot, R. K., & Giedd, J. N. (2010). Sex differences in the adolescent brain. *Brain and Cognition*, *72*(1), 46-55.

Lent, R. W., Brown, S. D., & Hackett, G. (1994). Toward a unifying social cognitive theory of career and academic interest, choice, and performance. *Journal of Vocational Behavior*, *45*(1), 79-122.

LePore, P. C., & Warren, J. R. (1997). A comparison of single-sex and coeducational Catholic secondary schooling: Evidence from the National Educational Longitudinal Study of 1988. *American Educational Research Journal*, *34*(3), 485-511.

Letonsaari, M., & Selin, J. (2017). Modeling computational algorithms using nonlinear storytelling methods of computer game design. *Procedia computer science*, *119*, 131-138.

Leutenegger, S., & Edgington, J. (2007). A games first approach to teaching introductory programming. *ACM SIGCSE Bulletin*, *39*(1), 115-118.

Levine, S. C., Foley, A., Lourenco, S., Ehrlich, S., & Ratliff, K. (2016). Sex differences in spatial cognition: Advancing the conversation. *Wiley Interdisciplinary Reviews: Cognitive Science*, *7(2),* 127-155.

Lewis, C. M., Anderson, R. E., & Yasuhara, K. (2016, August). I don't code all day: Fitting in Computer Science when the stereotypes don't fit. In *Proceedings of the 2016 ACM Conference on International Computing Education Research* (pp. 23-32). Leeds, UK: Association for Computing Machinery.

Li, N., & Kirkup, G. (2007). Gender and cultural differences in Internet use: A study of China and the UK. *Computers & Education*, *48*(2), 301-317.

Liao, Y. K. C., & Bright, G. W. (1991). Effects of computer programming on cognitive outcomes: A meta-analysis. *Journal of Educational Computing Research*, *7*(3), 251-266.

Lichtman, M. (2012). *Qualitative research in education: A user's guide*. Los Angeles, California: Sage Publications.

Liebenberg, J., Mentz, E., & Breed, B. (2012). Pair programming and secondary school girls' enjoyment of programming and the subject Information Technology (IT). *Computer Science Education*, *22*(3), 219-236.

Lim, K., & Meier, E. B. (2011). Different but similar: computer use patterns between young Korean males and females. *Educational Technology Research and Development*, *59*(4), 575-592.

Lin, C. H., Liu, E. Z. F., & Huang, Y. Y. (2012). Exploring parents' perceptions towards educational robots: Gender and socio‐economic differences. *British Journal of Educational Technology*, *43*(1), E31-E34.

Lin-Siegler, X., Dweck, C. S., & Cohen, G. L. (2016). Instructional interventions that motivate classroom learning. *Journal of Educational Psychology*, *108*(3), 295-299.

Lockwood, P. (2006). ”Someone like me can be successful”: Do college students need same-gender role models? *Psychology of Women Quarterly*, *30*(1), 36-46.

Logel, C., Walton, G. M., Spencer, S. J., Iserman, E. C., von Hippel, W., & Bell, A. E. (2009). Interacting with sexist men triggers social identity threat among female engineers. *Journal of personality and social psychology*, *96*(6), 1089-1103.

Lombard, M., Snyder, Duch, J., & Bracken, C. C. (2002). Content analysis in mass communication: Assessment and reporting of intercoder reliability. *Human Communication Research*, *28*(4), 587-604.

Long, M., Steinke, J., Applegate, B., Lapinski, M. K., Johnson, M. J., & Ghosh, S. (2010). Portrayals of male and female scientists in television programs popular among middle school-age children. *Science Communication*, *32*(3), 356-382.

Looi, H. C., & Seyal, A. H. (2014). Problem-based Learning: An Analysis of its Application to the Teaching of Programming. In *International Proceedings of Economics Development and Research*, (pp. 68-75). Singapore: IACSIT Press.

Loviglio, L. (1981). Mathematics and the brain: A tale of two hemispheres. *The Massachusetts Reading Teacher, (January/February)*, 8-12.

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12?. *Computers in Human Behavior*, *41*, 51-61.

Maguire, P., Maguire, R., Hyland, P., & Marshall, P. (2014). Enhancing collaborative learning using paired-programming: Who benefits?. *AISHE-J: The All Ireland Journal of Teaching and Learning in Higher Education*, *6*(2), 1411-1425.

Major, B., & O'brien, L. T. (2005). The social psychology of stigma. *Annual Review Psychology*, *56*, 393-421.

Major, L. (2010). Systematic literature review protocol: Teaching novices programming using robots. *Online Submission*. Retrieved September 12, 2016 from https://pdfs.semanticscholar.org/12cd/0f3debb8637b6e80835582223f8a020cfbf9.pdf.

Makuch, E. (2013). Rockstar: More than 1,000 people made GTAV. Retrieved February 08, 2017 from http://www.gamespot.com/articles/rockstar-more-than-1000-people-made-gtav/1100-6415330/.

Malan, D. J., & Leitner, H. H. (2007). Scratch for budding computer scientists. *ACM SIGCSE Bulletin*, *39*(1), 223-227.

Malgwi, C. A., Howe, M. A., & Burnaby, P. A. (2005). Influences on students' choice of college major. *Journal of Education for Business*, *80*(5), 275-282.

Maloney, J., Peppler, K., Kafai, Y. B., Resnick, M., & Rusk, N. (2008). Programming by choice: Urban youth learning programming with Scratch. In *Proceedings of SIGCSE '08*. New York, NY: ACM Press.

Mann, H. B., & Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics, 18*, 50-60.

Margolis, J., & Fisher, A. (2003). *Unlocking the clubhouse: Women in computing*. Cambridge, Massachusetts: MIT press.

Margolis, J., Fisher, A., & Miller, F. (2000). The anatomy of interest: Women in undergraduate computer science. *Women's Studies Quarterly*, *28*(1/2), 104-127.

Margulieux, L. E., Catrambone, R., & Guzdial, M. (2016). Employing subgoals in computer programming education. *Computer Science Education*, *26*(1), 44-67.

Marsh, H. W., & Seaton, M. (2013). Academic self-concept. In J. Hattie & E.M. Anderman, *International Guide to Student Achievement* (pp. 62-63). New York, NY: Routledge.

Marsh, H. W., & Shavelson, R. (1985). Self-concept: It's multifaceted, hierarchical structure. *Educational Psychologist*, *20*(3), 107-123.

Master, A., Cheryan, S., & Meltzoff, A. N. (2016). Computing whether she belongs: Stereotypes undermine girls' interest and sense of belonging in Computer Science. *Journal of Educational Psychology*, *108*(3), 424-437.

May, T. (2011). *Social research*. London: McGraw-Hill Education.

Mayer, R. E., Dyck, J. L., & Vilberg, W. (1986). Learning to program and learning to think: what's the connection? *Communications of the Association for Computing Machinery*, *29*(7), 605-610.

Mayer, R. E., & Fay, A. L. (1987). A chain of cognitive changes with learning to program in Logo. *Journal of Educational Psychology*, *79*(3), 269.

McCall, R. B. (1977). Childhood IQ's as predictors of adult educational and occupational status. *Science*, *197*(4302), 482-483.

McCane, B. (2009). Introductory programming with python. *The Python Papers Monograph*, *1*, 1-18.

McLaughlan, M., Nobert, M., O'Reilly, L., & Thorkelsson, P. (2009). Mixed methods research: An emerging paradigm? University of Victoria. Retrieved July, 26, 2016 from http://admn502awiki.pbworks.com/Mixed-methods+Research.

McNiff, J. (2016). *You and your action research project*. Abingdon, UK: Routledge.

Meece, J. L., Glienke, B. B., & Burg, S. (2006). Gender and motivation. *Journal of School Psychology*, *44*(5), 351-373.

Meeker, M., & Wu, L. (2013). Immigration in America & the growing shortage of high-skilled workers. Retrieved May 11, 2017 from http://www.kpcb.com /blog/immigration-in-america-the-growing-shortage-of-high-skilled-workers.

Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2013). Learning Computer Science concepts with Scratch. *Computer Science Education*, *23*(3), 239-264.

Metz, S. S. (2007). Attracting the engineering of 2020 today. In R. Burke & M. Mattis (Eds.), *Women and minorities in science, technology, engineering and mathematics: Upping the numbers* (pp. 184–209). Northampton, Massachusetts: Edward Elgar Publishing.

Miliszewska, I., & Tan, G. (2007). Befriending computer programming: A proposed approach to teaching introductory programming. *Informing Science: International Journal of an Emerging Transdiscipline*, 4(1), 277-289.

Miller, T., Mauthner, M., Birch, M., & Jessop, J. (Eds.). (2012). *Ethics in Qualitative Research*. London, UK: Sage Publications.

Misa, T. J. (Ed.). (2011). *Gender codes: Why women are leaving computing*. Hoboken, New Jersey: John Wiley & Sons.

Mitchell, R. (2013). Women Computer Science grads: The bump before the decline. Retrieved April 18, 2014 from http://blogs.computerworld.com/it-careers/21993/women-computer-science-visual-trendline.

Miyake, A., Kost-Smith, L. E., Finkelstein, N. D., Pollock, S. J., Cohen, G. L., & Ito, T. A. (2010). Reducing the gender achievement gap in college science: A classroom study of values affirmation. *Science*, *330*(6008), 1234-1237.

Mohsin, M. F. M., Norwawi, N. M., Hibadullah, C. F., & Wahab, M. H. A. (2010, November). Mining the student programming performance using rough set. In *Intelligent Systems and Knowledge Engineering (ISKE), 2010 International Conference* (pp. 478-483). Hangzhou, China: Institute of Electrical and Electronics Engineers.

Montagne, B., Kessels, R. P., Frigerio, E., de Haan, E. H., & Perrett, D. I. (2005). Sex differences in the perception of affective facial expressions: do men really lack emotional sensitivity? *Cognitive Processing*, *6*(2), 136-141.

Morrison, B. B., & DiSalvo, B. (2014, March). Khan academy gamifies computer science. In *Proceedings of the 45th ACM technical symposium on Computer science education* (pp. 39-44). Leeds: ACM.

Mota, M. I. G. (2007, October). Work in progress-using Lego mindstorms and robolab as a mean to lowering dropout and failure rate in programming course. In *Frontiers In*

Education Conference-Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007. FIE'07. 37th Annual (pp. F4A-1). Milwaukee, Wisconsin: Institute of Electrical and Electronics Engineers.

Mow, I. C. (2008). Issues and difficulties in teaching novice computer programming. In M.Iskander (Ed.), *Innovative techniques in instruction technology, e-learning, e-assessment, and education* (pp. 199-204). Dordrecht Netherlands: Springer.

Mullins, P., Whitfield, D., & Conlon, M. (2009). Using Alice 2.0 as a first language. *Journal of Computing Sciences in Colleges*, *24*(3), 136-143.

Multon, K. D., Brown, S. D., & Lent, R. W. (1991). Relation of self-efficacy beliefs to academic outcomes: A meta-analytic investigation. *Journal of Counseling Psychology*, *38*(1), 30.

Muratet, M., Torguet, P., Jessel, J. P., & Viallet, F. (2009). Towards a serious game to help students learn computer programming. *International Journal of Computer Games Technology*, *2009*, 3-14.

Murphy, M. C., Steele, C. M., & Gross, J. J. (2007). Signaling threat: How situational cues affect women in math, science, and engineering settings. *Psychological Science*, *18*(10), 879-885.

Murthy, D. (2008). Digital ethnography: An examination of the use of new technologies for social research. *Sociology*, *42*(5), 837-855.

Nachar, N. (2008). The Mann-Whitney U: A test for assessing whether two independent samples come from the same distribution. *Tutorials in Quantitative Methods for Psychology*, *4*(1), 13-20.

Nager, A., & Atkinson, R. D. (2016). The case for improving U.S. computer science education. *National Consortium of Secondary STEM Schools*, 21 (1), 18-19.

Natale, M. J. (2002). The effect of a male-oriented computer gaming culture on careers in the computer industry. *ACM SIGCAS Computers and Society, 32*(2), 24-31.

National Association for Single Sex Public Education (n.d.). Legal status of single-sex education. Retrieved February 06, 2017 from http://www.singlesexschools.org/policy-legalstatus.htm.

National Science Board, (2016). *Federal Support of Doctoral Researchers in Academia.* Science & Engineering Indicators 2016: Chapter 5 Academic Research and Development. Retrieved August 21, 2018 from https://www.nsf.gov/statistics/2016/nsb20161/#/report/chapter-5/doctoral-scientists-and-engineers-in-academia/federal-support-of-doctoral-researchers-in-academia.

National Science Foundation. (2017). *Survey of Graduate Students and Postdoctorates in Science and Engineering (GSS).* National Center for Science and Engineering Statistics: Integrated Science and Engineering Resources Data System (NCSES Data). Retrieved August 06, 2018 from https://ncsesdata.nsf.gov/ids/gss.

Newby, P. (2014). *Research methods for education*. Abingdon, UK: Routledge.

Newing, H., 2010. *Conducting research in conservation: social science methods and practice*. Abingdon, UK: Routledge.

Njemanze, P. C. (2007). Cerebral lateralisation for facial processing: Gender-related cognitive styles determined using Fourier analysis of mean cerebral blood flow velocity in the middle cerebral arteries. *Laterality*, *12*(1), 31-49.

Nuutila, E., Törmä, S., & Malmi, L. (2005). PBL and computer programming—the seven steps method with adaptations. *Computer Science Education*, *15*(2), 123-142.

Nuzhat, A., Salem, R. O., Hamdan, N. A., & Ashour, N. (2013). Gender differences in learning styles and academic performance of medical students in Saudi Arabia. *Medical Teacher*, *35*(sup1), S78-S82.

Obama, B. (2013). President Obama asks America to learn Computer Science. Retrieved April 18, 2014 from http://www.youtube.com/watch?v=6XvmhE1J9PY.

Obama, B. (2016). Obama outlines $4 billion 'Computer Science for All' education plan. *The Washington Post*. Retrieved February 08, 2016 from https://www.washingtonpost.com/local/education/obama-outlines-4-billion-computer-science-for-all-education-plan/2016/01/29/3ad40da2-c6d9-11e5-9693-933a4d31bcc8_story.html.

OECD (2010). PISA 2012 field trial problem solving framework. Retrieved April 17, 2017 from http://www.oecd.org/pisa/pisaproducts/46962005.pdf.

OECD (2013). OECD skills outlook 2013: First results from the survey of adult skills. Retrieved May 7, 2017 from https://www1.oecd.org/about/publishing/Corrigendum-OECD-skills-outlook-2013.pdf.

OECD (2014). PISA 2012 results: Creative problem solving (volume v): Students' skills in tackling real-life problems. Retrieved May 7, 2017 from https://www.oecd.org/pisa/keyfindings/PISA-2012-results-volume-V.pdf.

OECD (2015). The ABC of gender equality in education: Aptitude, behaviour, confidence. Retrieved May 7, 2017 from https://www.oecd.org/pisa/keyfindings/pisa-2012-results-gender-eng.pdf.

OECD (2016). PISA 2015 results (volume I): Excellence and equity in education. Retrieved May 7, 2017 from http://www.keepeek.com/Digital-Asset-Management/oecd/education/pisa-2015-results-volume-i_9789264266490-en#.WRAISomGP_Q#page1.

OECD (2017). The pursuit of gender equality: An uphill battle. Retrieved on September, 24, 2018 from http:// www.ungei.org/OECD_2017_The_Pursuit_of_Gender_Equality_book_2017.pdf.

O'Kelly, J., & Gibson, J. P. (2006). RoboCode & problem-based learning: a non-prescriptive approach to teaching programming. *ACM SIGCSE Bulletin*, *38*(3), 217-221.

Oleson, A., & Hora, M. T. (2014). Teaching the way they were taught? Revisiting the sources of teaching knowledge and the role of prior experience in shaping faculty teaching practices. *Higher Education*, *68*(1), 29-45.

Olson, S., & Riordan, D. G. (2012). Engage to Excel: Producing One Million Additional College Graduates with Degrees in Science, Technology, Engineering, and Mathematics. Report to the President. Executive Office of the President.

Ortega, F. R., Bolivar, S., Bernal, J., Galvan, A., Tarre, K., Rishe, N., & Barreto, A. (2017, March). Towards a 3D Virtual Programming Language to increase the number of women in computer science education. In *K-12 Embodied Learning through Virtual & Augmented Reality (KELVAR), 2017 IEEE Virtual Reality Workshop on* (pp. 1-6). Los Angeles, California: Institute of Electrical and Electronics Engineers.

Owolabi, J., Olanipekun, P., & Iwerima, J. (2014). Mathematics ability and anxiety, computer and programming anxieties, age and gender as determinants of achievement in basic programming. *Journal on Computing*, *3*(4), 109-114.

Özdemir Uluç , F. (2017). How to approach teaching gender equality to boys and girls. Retrieved September 20, 2017 from https://www.britishcouncil.org/voices-magazine/how-approach-teaching-gender-equality-boys-and-girls.

Pahlke, E., Hyde, J. S., & Mertz, J. E. (2013). The effects of single-sex compared with coeducational schooling on mathematics and science achievement: Data from Korea. *Journal of Educational Psychology*, *105*(2), 444-452.

Pande, V., van der Weide, T. P., & Pande, R. (2016). Computer games and the reinforcement of gender gaps. In J. Steans & D. Tepe-Belfrage (Eds), *Handbook on Gender in World Politics,* (vol 39, pp. 333-342). Cheltenham, UK: Edward Elgar Publishing Limited.

Panitz, T. (1999). Collaborative versus cooperative learning: A comparison of the two concepts which will help us understand the underlying nature of interactive learning. Retrieved February 17, 2017 from http://files.eric.ed.gov/fulltext/ED448443.pdf.

Papastergiou, M. (2009). Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation. *Computers & Education*, *52*(1), 1-12.

Papert, S. (1980). *Mindstorms: Children, computers and powerful ideas.* New York, NY: Basic Books.

Papert, S., & Harel, I. (1991). Situating constructionism. *Constructionism*, *36*, 1-11.

Parcells, N. (2015). What 50,000 students have to say about college recruiting? Retrieved February 8, 2016 from https://www.looksharp.com/blog/what-50000-students-have-to-say-about-college-recruiting.

Pardamean, B., Evelin, E., & Honni, H. (2011, December). The effect of logo programming language for creativity and problem solving. In *Proceedings of the 10th WSEAS international conference on E-Activities* (pp. 151-156). Jakarta, Indonesia: World Scientific and Engineering Academy and Society (WSEAS).

Park, C. J., Hyun, J. S., & Heuilan, J. (2015, October). Effects of gender and abstract thinking factors on adolescents' computer program learning. In *Frontiers in Education Conference (FIE),* 468-473. El Paso, Texas: Institute of Electrical and Electronics Engineers.

Park, J. H. (2014). A Study on Digital Storytelling Based Programming Education. *Journal of the Korea Society of Computer and Information*, *19*(5), 119-128.

Park, H., Behrman, J. R., & Choi, J. (2013). Causal effects of single-sex schools on college entrance exams and college attendance: Random assignment in Seoul high schools. *Demography*, *50*(2), 447-469.

Park, Y. S., & Kim, U. (2006). Family, parent-child relationship, and academic achievement in Korea. In U. Kim, K. Yang, K. Hwang (Eds.). *Indigenous and Cultural Psychology* (pp. 421-443). New York, NY: Springer.

Patton, M. Q. (2002). *Qualitative research and evaluation methods (3rd ed.)*. Thousand Oaks, California: Sage.

Paulson, L. D. (2007). Developers shift to dynamic programming languages. *Computer*, *40*(2), 12-15.

Payne, G., & Payne, J. (2009). *Key concepts in social research*. London, UK: Sage.

Pea, R. D., & Kurland, D. M. (1984). On the cognitive effects of learning computer programming. *New Ideas in Psychology*, *2*(2), 137-168.

Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., Devlin, M., & Paterson, J. (2007, December). A survey of literature on the teaching of introductory programming. *ACM SIGCSE Bulletin*, *39*(4), 204-223.

Pekrun, R. (2014). *Emotions and learning.* International Bureau of Education, Belgium: International Academy of Education. Retrieved December 02, 2015 from http://www.ibe.unesco.org/sites/default/files/resources/edu-practices_24_eng.pdf.

Pekrun, R., & Stephens, E. J. (2012). Academic emotions. In K. R. Harris, S. Graham, & T. Urdan (Eds.), *APA educational psychology handbook* (Vol. 2, pp. 3–31). Washington, D.C.: American Psychological Association.

Peng, W. (2010, September). Practice and experience in the application of problem-based learning in computer programming course. In *Educational and Information Technology (ICEIT), 2010 International Conference,* (vol. 1, pp. 1-170). Chongqing, China: Institute of Electrical and Electronics Engineers.

Peng, R., Sun, D., & Tsai, W. T. (2014, March). Success factors in mobile social networking application development: case study of Instagram. In *Proceedings of the 29th Annual*

ACM Symposium on Applied Computing (pp. 1072-1079). Gyeongju, Republic of Korea: Association for Computing Machinery.

Peper, J. S., & Dahl, R. E. (2013). The teenage brain surging hormones—brain-behavior interactions during puberty. *Current Directions in Psychological Science*, *22*(2), 134-139.

Pera, A. (2014). Neural mechanisms underlying school-based learning. *Contemporary Readings in Law and Social Justice*, *6*(1), 7.

Perkins, D. N., & Salomon, G. (1992). Transfer of learning. *International Encyclopedia of Education*, *2*, 6452-6457.

Perrone, K. M., Zanardelli, G., & Chartrand, J. M. (2002). Role model influence on the career decidedness of college students. *College Student Journal*, *36,* 109-112.

Perry, E., & Francis, B. (2010). The social class gap for educational achievement: a review of the literature. Retrieved on September, 24, 2018 from https://www.thersa.org/globalassets/pdfs/blogs/rsa-social-justice-paper.pdf.

Petronzio, M. (2016, January). 8 ways you can empower girls to learn coding. Retrieved September 09, 2017 from http://mashable.com/2016/01/27/girls-coding-how-to-help/#CB_RRLQ3qSq3.

Petroutsos, E. (2008). *Mastering Microsoft visual basic 2008*. New York, NY: John Wiley & Sons.

Plump, C. M., & LaRosa, J. (2017). Using Kahoot! in the classroom to create engagement and active learning: a game-based technology solution for elearning novices. *Management Teaching Review*, *2*(2), 151-158.

Powers, K., Gross, P., Cooper, S., McNally, M., Goldman, K. J., Proulx, V., & Carlisle, M. (2006, March). Tools for teaching introductory programming: what works? *ACM SIGCSE Bulletin, 38*(1), 560-561.

Prensky, M. (2008, January 13). Programming Is the New Literacy. Retrieved July 13, 2016 from https://www.edutopia.org/literacy-computer-programming.

Prinsen, F. R., Volman, M. L. L., & Terwel, J. (2007). Gender-related differences in computer-mediated communication and computer-supported collaborative learning. *Journal of Computer Assisted Learning*, *23*(5), 393-409.

Ramayah, M., Sivanandan, P., Nasrijal, N. H., Letchumanan, T., & Leong, L. C. (2009). Preferred learning style: Gender influence on preferred learning style among business students. *Journal of US-China Public Administration*, *6*(4), 65-78.

Ramsey, L. R., Betz, D. E., & Sekaquaptewa, D. (2013). The effects of an academic environment intervention on science identification among women in STEM. *Social Psychology of Education*, *16*(3), 377-397.

Raven, J. (2000). Psychometrics, cognitive ability, and occupational performance. *Review of Psychology*, *7*(1-2), 51-74.

Reichardt, C. S. (2009). Quasi-experimental design. In R. E. Millsap & A. Maydeu-Olivares (Eds.), *The SAGE handbook of quantitative methods in psychology (pp. 46-71).* Thousand Oaks, California: Sage.

Reilly, D., Neumann, D. L., & Andrews, G. (2016). Gender differences in spatial ability: Implications for STEM education and approaches to reducing the gender gap for parents and educators. In M. S. Khine (Ed.), *Visual-spatial ability in STEM: Transforming research into practice*. Basel, Switzerland: Springer.

Rétornaz, P., Riedo, F., Magnenat, S., Vaussard, F., Bonani, M., & Mondada, F. (2013, August). Seamless multi-robot programming for the people: ASEBA and the wireless Thymio II robot. In *Information and Automation (ICIA), 2013 IEEE International Conference,* (pp. 337-343). Yinchuan, China: Institute of Electrical and Electronics Engineers.

Rice, L., Barth, J. M., Guadagno, R. E., Smith, G. P., & McCallum, D. M. (2013). The role of social support in students' perceived abilities and attitudes toward math and science. *Journal of Youth and Adolescence*, *42*(7), 1028-1040.

Richards, L. (2009). *Handling qualitative data: A practical guide*. London, UK: Sage Publications.

Rieber, L. (2005). Multimedia learning with games, simulations, and microworlds. In R. E. Mayer (Ed.), *Cambridge handbook of multimedia learning* (pp. 549–567). New York, NY: Cambridge University Press.

RisingStars-uk. (n.d.). Switched on computing. Retrieved November 02, 2016, from http://www.risingstars-uk.com/Series/Switched-On-Computing.

Ritchie, J., Lewis, J., Nicholls, C. M., & Ormston, R. (2014). *Qualitative research practice: a guide for social science students and researchers*. Los Angeles, California: Sage Publications.

Rizvi, M., Humphries, T., Major, D., Jones, M., & Lauzun, H. (2011). A CS0 course using scratch. *Journal of Computing Sciences in Colleges*, *26*(3), 19-27.

Robertson, J. (2013). The influence of a game-making project on male and female learners' attitudes to computing. *Computer Science Education*, *23*(1), 58-83.

Robertson, J., & Howells, C. (2008). Computer game design: Opportunities for successful learning. *Computers & Education*, 50(2), 559-578.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, *13*(2), 137-172.

Robnett, R. D., & Leaper, C. (2013). Friendship groups, personal motivation, and gender in relation to high school students' STEM career interest. *Journal of Research on Adolescence*, *23*(4), 652-664.

Robson, C., & McCartan, K. (2016). *Real world research*. Hoboken, New Jersey: John Wiley & Sons, Inc.

Rocks, S. (2015). Next steps for the BBC Micro: Bit. Retrieved February 8, 2016 from http://www.bbc.co.uk/blogs/aboutthebbc/entries/4678b923-29e6-4912-a643-b637cf5c9f03.

Rodrigo, M. M. T., Baker, R. S., Jadud, M. C., Amarra, A. C. M., Dy, T., Espejo-Lahoz, M. B. V., Lim, S.A.L., Pascua, S.A., Sugay, J.O., & Tabanao, E. S. (2009). Affective and behavioral predictors of novice programmer achievement. *ACM SIGCSE Bulletin*, *41*(3)*,* 156-160.

Roorda, D. L., Koomen, H. M., Spilt, J. L., & Oort, F. J. (2011). The influence of affective teacher–student relationships on students' school engagement and achievement: A meta-analytic approach. *Review of Educational Research*, *81*(4), 493-529.

Rosenberg, H., Grad, H. A., & Matear, D. W. (2003). The effectiveness of computer-aided, self-instructional programs in dental education: a systematic review of the literature. *Journal of Dental Education*, *67*(5), 524-532.

Rossi, J. S. (2013). Statistical power analysis. In J. A. Schinka & W. F. Velicer (Eds.), Handbook of psychology: Volume 2. Research methods in psychology (2nd ed., pp. 71-108). New York, NY: Wiley.

Rothwell, J. (2013). *The hidden STEM economy.* Washington D.C.: Brookings Institution.

Rowell, G. H., Perhac, D. G., Hankins, J. A., Parker, B. C., Pettey, C. C., & Iriarte-Gross, J. M. (2003, February). Computer-related gender differences. *ACM SIGCSE Bulletin*, *35(1)*, 54-58.

Rubio, M. A., Romero-Zaliz, R., Mañoso, C., & Angel, P. (2015). Closing the gender gap in an introductory programming course. *Computers & Education, 82,* 409-420.

Ruble, D. N., & Martin, C. L., & Berenbaum, S. A. (1998). Gender development. In W. Damon & N. Eisenberg (Eds.), *Handbook of child psychology: Vol. 3. Social, emotional, and personality development* (pp. 933-1016). New York, NY: Wiley

Ruigrok, A. N., Salimi-Khorshidi, G., Lai, M. C., Baron-Cohen, S., Lombardo, M. V., Tait, R. J., & Suckling, J. (2014). A meta-analysis of sex differences in human brain structure. *Neuroscience & Biobehavioral Reviews*, *39*, 34-50.

Rushkoff, D., & Purvis, L. (2011). *Program or be programmed: ten commands for a digital age*. Berkeley, California: Soft Skull Press.

Saadé, R. G., & Kira, D. (2009). Computer anxiety in e-learning: The effect of computer self-efficacy. *Journal of Information Technology Education*, *8*(1), 177-191.

Sáez-López, J. M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education*, *97*, 129-141.

Sale, J. E., Lohfeld, L. H., & Brazil, K. (2002). Revisiting the quantitative-qualitative debate: Implications for mixed-methods research. *Quality and Quantity*, *36*(1), 43-53.

Salleh, N., Mendes, E., Grundy, J., & Burch, G. S. J. (2010, May). An empirical study of the effects of conscientiousness in pair programming using the five-factor personality model. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1* (pp. 577-586). Cape Town, South Africa: Association for Computing Machinery.

Sangwin, C. J., & O'Toole, C. (2017). Computer programming in the UK undergraduate mathematics curriculum. *International Journal of Mathematical Education in Science and Technology*, *48*, 1-20.

Saunders, M., Lewis, P., & Thornhill, A. (2012). *Research methods for business students*. Harlow, UK: Pearson.

Savage, M., Devine, F., Cunningham, N., Taylor, M., Li, Y., Hjellbrekke, J., Le Roux, B., Friendman, S. & Miles, A. (2013). A new model of social class? Findings from the BBC's Great British Class Survey experiment. *Sociology*, *47*(2), 219-250.

Savery, J. R., & Duffy, T. M. (1995). Problem based learning: An instructional model and its constructivist framework. *Educational Technology*, *35*(5), 31-38.

Sax, L. J., Arms, E., Woodruff, M., Riggers, T., & Eagan, K. (2009). *Women graduates of single-sex and coeducational high schools: Differences in their characteristics and the transition to college*. Los Angeles, CA: Sudikoff Family Institute for Education and New Media.

Sax, L. J., Lehman, K. J., Jacobs, J. A., Kanny, M. A., Lim, G., Monje-Paulson, L., & Zimmerman, H. B. (2017). Anatomy of an enduring gender gap: The evolution of women's participation in computer science. *The Journal of Higher Education*, *88*(2), 258-293.

Sax, L. J., Riggers, T. A., & Eagan, M. K. (2013). The Role of Single-Sex Education in the Academic Engagement of College-Bound Women: A Multilevel Analysis. *Teachers College Record*, *115*(1), 1-27.

Sax, L. J., Shapiro, C. A., & Eagan, M. K. (2011). Promoting Mathematical and Computer Self-Concept Among Female College Students: Is There a Role of Single-Sex Secondary Education? *Journal of Women and Minorities in Science and Engineering*, *17*(4), 325-355.

Schäfer, A., Holz, J., Leonhardt, T., Schroeder, U., Brauner, P., & Ziefle, M. (2013). From boring to scoring–a collaborative serious game for learning and practicing mathematical logic for Computer Science education. *Computer Science Education*, *23*(2), 87-111.

Schanzenbach, D. W. (2014). *Does class size matter?* Boulder, CO: National Education Policy Center.

Scharkow, M., Festl, R., Vogelgesang, J., & Quandt, T. (2015). Beyond the “core-gamer”: Genre preferences and gratifications in computer games. *Computers in Human Behavior*, 44, 293-298.

Schmidt, E. (2011). Full Transcript: Eric Schmidt’s Edinburgh Festival Keynote MacTaggart Lecture. Retrieved March 31, 2014 from http://www.theguardian.com/media/interactive/2011/aug/26/eric-schmidt-mactaggart-lecture-full-text.

Scott, M. and Ghinea, G. 2013. Educating programmers: a reflection on barriers to deliberate practice. In *HEA Conference on Learning and Teaching in STEM Disciplines,* Birmingham, UK: The Higher Education Academy.

Scott, A., Watkins, M., & McPhee, D. (2007). A Step Back From Coding-An Online Environment and Pedagogy for Novice Programmers. In *Proceedings of the 11th Java in the Internet Curriculum Conference* (pp. 35-41). Leeds, UK: The Higher Education Academy.

Searle, J. R. (2007). *The construction of social reality*. New York, NY: Free Press.

Sentance, S., Humphreys, S., & Dorling, M. (2014, November). The network of teaching excellence in computer science and master teachers. In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education* (pp. 80-88). Berlin, Germany: Association for Computing Machinery.

Shapiro, E. C., Haseltine, F. P., & Rowe, M. P. (1978). Moving up: Role models, mentors, and the" Patron System". *Sloan Management Review*, *19*(3), 51.

Shapiro, J. R., & Williams, A. M. (2012). The role of stereotype threats in undermining girls’ and women’s performance and interest in STEM fields. *Sex Roles*, *66*(3-4), 175-183.

Shaughnessy, J. J., Zechmeister, E. B., & Zechmeister, J. S. (2015). *Research methods in psychology*. Dubuque, Iowa: McGraw-Hill Education.

Shin, J., Lee, H., & Kim, Y. (2009). Student and school factors affecting mathematics achievement: International comparisons between Korea, Japan and the USA. *School Psychology International*, *30*(5), 520-537.

Shin, J., Lee, H., McCarthy-Donovan, A., Hwang, H., Yim, S., & Seo, E. (2015). Home and motivational factors related to science-career pursuit: Gender differences and gender similarities. *International Journal of Science Education*, *37*(9), 1478-1503.

Shneiderman, B., Mayer, R., McKay, D., & Heller, P. (1977). Experimental investigations of the utility of detailed flowcharts in programming. *Communications of the ACM*, *20*(6), 373-381.

Shortland, S. (2014). Role models: expatriate gender diversity pipeline or pipe-dream? *Career Development International*, *19*(5), 572-594.

Shucard, J. L., & Shucard, D. W. (1990). Auditory evoked potentials and hand preference in 6-month-old infants: Possible gender-related differences in cerebral organization. *Developmental Psychology*, *26*(6), 923.

Siemens, G. (2014). Connectivism: A learning theory for the digital age. Retrieved Jan 05, 2016 from http://t.edtechpolicy.org/AAASGW/Session2/siemens_article.pdf.

Sikes, P 2004, Methodology, procedures and ethical concerns. In C. Opie (Ed) *Doing educational research: A guide for first time researchers*, (pp.15–33). London, UK: Sage Publishers.

Silim, A., & Crosse, C. (2014). Women in Engineering: Fixing the talent pipeline. Retrieved June 18, 2016 from http://www.ippr.org/files/publications/pdf/women-in-engineering_Sept2014.pdf?noredirect=1.

Simsek, A. (2011). The relationship between computer anxiety and computer self-efficacy. *Online Submission*, *2*(3), 177-187.

Singaravelu, H. D., White, L. J., & Bringaze, T. B. (2005). Factors influencing international students' career choice: A comparative study. *Journal of Career Development*, *32*(1), 46-59.

Singh, K., Allen, K. R., Scheckler, R., & Darlington, L. (2007). Women in computer-related majors: A critical synthesis of research and theory from 1994 to 2005. *Review of Educational Research*, *77*(4), 500-533.

Skaalvik, S., & Skaalvik, E. M. (2004). Gender differences in math and verbal self-concept, performance expectations, and motivation. *Sex Roles*, *50*(3-4), 241-252.

Sloan, R. H., & Troy, P. (2008, March). CS 0.5: A better approach to introductory computer science for majors. *ACM SIGCSE Bulletin, 40*(1), 271-275.

Smith, K. N., & Gayles, J. G. (2018). "Girl Power": Gendered Academic and Workplace Experiences of College Women in Engineering. *Social Sciences*, *7*(1), 11.

Smith, N., Allsop, Y., Caldwell, H., Hill, D., Dimitriadi, Y., & Csizmadia, A. P. (2015, November). Master Teachers in Computing: What have we achieved?. In *Proceedings of the Workshop in Primary and Secondary Computing Education*(pp. 21-24). ACM.

Soares, A., Fonseca, F., & Martin, N. L. (2015). Teaching introductory programming with game design and problem-based learning. *Issues in Information Systems*, *16*(3), 128-137.

Soloway, E., Bonar, J., & Ehrlich, K. (1983). Cognitive strategies and looping constructs: An empirical study. *Communications of the ACM*, *26*(11), 853-860.

Son, J. H. (2013). Career experiences of highly educated professional women: Focusing on women with 5-10 career years. *Korean Journal of Counseling*, *14*, 501-522.

Spielhagen, F. R. (2013). *Debating single-sex education: Separate and equal?* Lanham, Maryland: Rowman & Littlefield Education.

Stahl, G., Koschmann, T., & Suthers, D. (2006). Computer-supported collaborative learning: An historical perspective. In R. K. Sawyer (Ed.), *Cambridge handbook of the learning sciences*, (pp. 409–426). Cambridge, UK: Cambridge University Press.

Statista (2018). U.S. video gamer gender statistics 2018. Retrieved September 6, 2018, from https://www.statista.com/statistics/232383/gender-split-of-us-computer-and-video-gamers/.

Statistic Korea. (2013). Employment data. Retrieved April, 20, 2016 from http://kostat.go.kr /portal/korea/index.action.

Steele, C. M., Spencer, S. J., & Aronson, J. (2002). Contending with group image: The psychology of stereotype and social identity threat. In M. P. Zanna (Ed.), *Advances in experimental social psychology* (Vol. 34, pp. 379-440). San Diego, California: Academic Press.

Steinmayr, R., & Spinath, B. (2009). What explains boys' stronger confidence in their intelligence? *Sex Roles*, *61*(9-10), 736-749.

Stoet, G., & Geary, D. C. (2012). Can stereotype threat explain the gender gap in mathematics performance and achievement? *Review of General Psychology*, *16*(1), 93-102.

Stoet G, Geary DC (2013) Sex Differences in Mathematics and Reading Achievement Are Inversely Related: Within- and Across-Nation Assessment of 10 Years of PISA Data. PLoS ONE 8(3): e57988. Retrieved Mar 15, 2016 from https://doi.org/10.1371/journal.pone.0057988.

Stolee, K. T., & Fristoe, T. (2011, March). Expressing computer science concepts through Kodu game lab. In *Proceedings of the 42nd ACM technical symposium on Computer science education,* 99-104. Dallas, Texas: Association for Computing Machinery.

Stout, J. G., Dasgupta, N., Hunsinger, M., & McManus, M. A. (2011). STEMing the tide: using ingroup experts to inoculate women's self-concept in science, technology, engineering, and mathematics (STEM). *Journal of personality and social psychology*, *100*(2), 255-270.

Sullivan, A., & Bers, M. U. (2013). Gender differences in kindergarteners' robotics and programming achievement. *International Journal of Technology and Design Education*, *23*(3), 691-702.

Summers, L. (2005). Why women are poor at science, by Harvard president. Retrieved January 20, 2016, from https://www.theguardian.com/science/2005/jan/18/educationsgendergap.genderissues.

Tabet, N., Gedawy, H., Alshikhabobakr, H., & Razak, S. (2016, July). From alice to python. Introducing text-based programming in middle schools. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 124-129). Association for Computing Machinery.

Tale, S. (2016). *SQL the ultimate beginners guide: Learn SQL today.* North Charleston, South Carolina: CreateSpace Independent Publishing Platform.

Tashakkori, A., & Creswell, J. W. (2008). Editorial: Mixed methodology across disciplines. *Journal of Mixed Methods Research*, *2*(1), 3-6.

Tatroe, K., MacIntyre, P., & Lerdorf, R. (2014). *Programming PHP*. Sebastopol, California: O'Reilly Media.

Teddlie, C., & Tashakkori, A. (Eds.). (2009). *Foundations of mixed methods research: Integrating quantitative and qualitative approaches in the social and behavioral sciences*. Los Angeles, California: Sage Publications Inc.

Tekinarslan, E. (2008). Computer anxiety: A cross-cultural comparative study of Dutch and Turkish university students. *Computers in Human Behavior*, *24*(4), 1572-1584.

Terrell, J., Kofink, A., Middleton, J., Rainear, C., Murphy-Hill, E., & Parnin, C. (2016). *Gender bias in open source: Pull request acceptance of women versus men.* Technical Report. PeerJ PrePrints.

Teo, T (2006) Attitudes toward computers: A study of post-secondary students in Singapore. *Interactive Learning Environments 14*(1), 17-24.

Tew, A. E., Fowler, C., & Guzdial, M. (2005, February). Tracking an innovation in introductory CS education from a research university to a two-year college. *ACM SIGCSE Bulletin*, *37*(1), 416-420.

Thomas, D., Fowler, C., & Hunt, A. (2013). *Programming Ruby 1.9 and 2.0: The pragmatic programmer's guide (Vol. 2)*. Raleigh, North Carolina: The Pragmatic Programmers, LLC.

Tinsley, H. E. A., & Weiss, D. J. (2000). Interrater reliability and agreement. In H. E. A. Tinsley & S. D. Brown (Eds.), *Handbook of applied multivariate statistics and mathematical modeling* (pp. 95–124). San Diego, California: Academic Press.

Todman, J., & Day, K. (2006). Computer anxiety: The role of psychological gender. *Computers in Human Behavior*, *22*(5), 856-869.

Tsai, M. J. (2002). Do male students often perform better than female students when learning computers?: a study of Taiwanese eighth graders' computer education through strategic and cooperative learning. *Journal of Educational Computing Research*, *26*(1), 67-85.

Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., & Verno, A. (2003). A model curriculum for K-12 Computer Science. *Final Report of the ACM K-12 Task Force Curriculum Committee, CSTA*.

Turkle, S., & Papert, S. (1990). Epistemological pluralism: Styles and voices within the computer culture. *Signs*, *16*(1), 128-157.

United Nations (2009). Department of Economic, & United Nations. Department of Public Information. *The millennium development goals report 2009*. United Nations Publications.

United Nations Educational, Scientific and Cultural Organization (2017). Women still a minority in engineering and computer science. Retrieved July 01, 2018 from http://www.unesco.org/new/en/media-services/single-view/news/women_still_a_minority_in_engineering_and_computer_science/.

United Nations Education, Scientific and Cultural Organization (UNESCO) Bangkok (2017). New education policies and practices in South Korea. Retrieved July 02, 2018 from https://bangkok.unesco.org/content/new-education-policies-and-practices-south-korea.

Unger, R. K. (1979). Toward a redefinition of sex and gender. *American Psychologist*, 34(11), 1085-1094.

U.S. Bureau of Labor Statistics (n.d.). Retrieved August 25, 2017 from https://www.bls.gov/ooh/computer-and-information-technology/computer-programmers.html.

Utting, I., Cooper, S., Kölling, M., Maloney, J., & Resnick, M. (2010). Alice, greenfoot, and scratch--a discussion. *ACM Transactions on Computing Education (TOCE)*, *10*(4), 17:1-11.

Valcke, M., Bonte, S., De Wever, B., & Rots, I. (2010). Internet parenting styles and the impact on Internet use of primary school children. *Computers & Education*, *55*(2), 454-464.

Visser, P. S. Krosnick. JA, & Lavrakas, P.(2000). Survey research. In Reis, H.T. & Judd, C. M. *(Eds.), Handbook of research methods in social psychology*. New York, NY: Cambridge University Press.

Wai, J., Lubinski, D., Benbow, C. P., & Steiger, J. H. (2010). Accomplishment in science, technology, engineering, and mathematics (STEM) and its relation to STEM educational dose: A 25-year longitudinal study. *Journal of Educational Psychology*, *102*(4), 860.

Walker, A., & Dimmock, C. A. (Eds.). (2002). *School leadership and administration: adopting a cultural perspective*. New York, NY: RoutledgeFarmer.

Walliman, N. (2011). *Research methods: The basics*. London, UK: Routledge.

Wang, H. Y., Huang, I., & Hwang, G. J. (2016). Comparison of the effects of project-based computer programming activities between mathematics-gifted students and average students. *Journal of Computers in Education*, *3*(1), 33-45.

Warrington, M., Younger, M., & Williams, J. (2000). Student attitudes, image and the gender gap. *British Educational Research Journal*, *26*(3), 393-407.

Watson, D., & Williams, H. (2015). *Cambridge IGCSE® Computer Science Revision Guide*. Cambridge: Cambridge University Press.

Watt, H. M. (2005). Explaining gendered math enrollments for NSW Australian secondary school students. *New Directions for Child and Adolescent Development*, *2005*(110), 15-29.

Wechsler, D. (2003). *Wechsler intelligence scale for children–Fourth Edition* (WISC-IV). San Antonio, TX: The Psychological Corporation.

Wedekind, J., Amavasai, B. P., Dutton, K., & Boissenin, M. (2008, June). A machine vision extension for the Ruby programming language. In *Information and Automation, 2008. ICIA 2008. International Conference on* (pp. 991-996). Changsha, China: Institute of Electrical and Electronics Engineers.

Wehrwein, E. A., Lujan, H. L., & DiCarlo, S. E. (2007). Gender differences in learning style preferences among undergraduate physiology students. *Advances in Physiology Education*, *31*(2), 153-157.

Werner, L., Denner, J., Bliesner, M., & Rex, P. (2009, April). Can middle-schoolers use Storytelling Alice to make games?: results of a pilot study. In *Proceedings of the 4th International Conference on foundations of digital games* (pp. 207-214). New York, NY: Association for Computing Machinery.

Werner, L. L., Hanks, B., & McDowell, C. (2004). Pair-programming helps female computer science students. *Journal on Educational Resources in Computing*, *4*(1), 1-8.

Wharton, A. S. (2009). *The sociology of gender: An introduction to theory and research*. New York, NY: John Wiley & Sons.

White, S., Carter, J., Jamieson, S., Efford, N., & Jenkins, T. (2007, October). TOPS-Collaboration and competition to stretch our most able programming novices. In *Frontiers In Education Conference-Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007. FIE'07. 37th Annual* (pp. S2J-20). Milwaukee, Wisconsin: Institute of Electrical and Electronics Engineers.

Wilcox, R. R. (2005). New methods for comparing groups: Strategies for increasing the probability of detecting true differences. *Current Directions in Psychological Science*, *14*(5), 272-275.

Williams, D. A., & King, P. (1980). Do males have a math gene? *Newsweek*, *96*, 73-75.

Williams, O. (2014, June 02). Apple Announces Swift, A New Programming Language for iOS and OS X. Retrieved September 01, 2017 from https://thenextweb.com/apple/2014/06/02/apple-announces-swift-new-programming-language-ios/#.tnw_2Buh41Mx.

Willoughby, T. (2008). A short-term longitudinal study of Internet and computer game use by adolescent boys and girls: prevalence, frequency of use, and psychosocial predictors. *Developmental Psychology*, *44*(1), 195-204.

Wilson, B. C. (2002). A study of factors promoting success in computer science including gender differences. *Computer Science Education*, *12*(1-2), 141-164.

Wing, J. M. (2006). Computational thinking. *Communications of the Association for Computing Machinery*, *49*(3), 33-35.

Wise, A.F., & O'Neill, K. (2009). Beyond more versus less. In S. Tobias and T.M. Duffy (Eds.) *Constructivist theory applied to instruction: Success or failure?* (pp. 82-105). New York, NY: Taylor and Francis.

Wistedt, I. (2001). Five gender-inclusive projects revisited. A follow-up study of the Swedish Government's initiative to recruit more women to higher education in mathematics, science and technology. Stockholm, Sweden: Høgskoleverket.

Wolz, U., Leitner, H. H., Malan, D. J., & Maloney, J. (2009, March). Starting with scratch in CS 1. *ACM SIGCSE Bulletin*, *41(1),* 2-3.

Women in Science and Engineering (WISE) Campaign (n.d.). Core STEM graduates 2017. Retrieved July 01, 2018 from https://www.wisecampaign.org.uk/statistics/core-stem-graduates-2017/.

Woodside, A. G. (2010). *Case study research: Theory, methods and practice*. Bingley: Emerald Group Publishing.

Woolley, A. W., Chabris, C. F., Pentland, A., Hashmi, N., & Malone, T. W. (2010). Evidence for a collective intelligence factor in the performance of human groups. *Science*, *330*(6004), 686-688.

Wright, K. B. (2005). Researching Internet-based populations: Advantages and disadvantages of online survey research, online questionnaire authoring software packages, and web survey services. *Journal of computer-mediated communication*, *10*(3), Retrieved September 12, 2018 from http://jcmc.indiana.edu/vol10/issue3/wright.html.

Wuchy, S., Jones, B. F., & Uzzi, B. (2007). The increasing dominance of team in production of knowledge. *Science*, *316*(5827), 1036–1039.

Wyatt-Nichol, H., Brown, S., & Haynes, W. (2011). Social class and socioeconomic status: Relevance and inclusion in MPA-MPP programs. *Journal of Public Affairs Education*, *17*(2), 187-208.

Xinogalos, S. (2013, March). Using flowchart-based programming environments for simplifying programming and software engineering processes. *Global Engineering Education Conference (EDUCON),* 1313-1322.

Xu, Y. J. (2008). Gender disparity in STEM disciplines: A study of faculty attrition and turnover intentions. *Research in Higher Education*, *49*(7), 607-624.

Yadin, A. (2011). Reducing the dropout rate in an introductory programming course. *ACM Inroads*, *2*(4), 71-76.

Yan, B., Becker, D., & Hecker, C. (2011). An effective way of introducing iPhone application development to undergraduate students. *Journal of Computing Sciences in Colleges*, *26*(5), 166-173.

Yin, R. K. (2014). *Case study research: design and methods*. London, UK: Sage Publication.

Young, A. (2017, August 07). Google employee's viral anti-diversity manifesto confirms your worst fears about tech-bro culture. Retrieved August 20, 2017 from http://www.salon.com/2017/08/07/google-employees-viral-anti-diversity-manifesto-confirms-your-worst-fears-about-tech-bro-culture/.

Young, D. M., Rudman, L. A., Buettner, H. M., & McLean, M. C. (2013). The influence of female role models on women's implicit science cognitions. *Psychology of Women Quarterly*, *37*(3), 283-292.

Yuan, Z., Pan, H., & Zhang, L. (2008). A novel pen-based flowchart recognition system for programming teaching. In E. Leung, F. Wang, L. Miao, J. Zhao, & J. He (Eds.), *Advances in Blended Learning* (pp. 55-64). New York, NY: Springer.

Yusupova, N. (2014). Nelly Yusupova: CTO. Entrepreneur. Risk-taker. Movement-starter. Retrieved July 30, 2014 from http://www.techrepublic.com/blog/smb-technologist/nelly-yusupova-cto-entrepreneur-risk-taker-movement-starter/.

Zaidi, R., Freihofer, I., & Townsend, G. C. (2017, March). Using Scratch and Female Role Models while Storytelling Improves Fifth-Grade Students' Attitudes toward Computing. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 791-792). Seattle, Washington: Association for Computing Machinery.

Zehetmeier, D., Böttcher, A., Brüggemann-Klein, A., & Thurner, V. (2015). Development of a classification scheme for errors observed in the process of computer programming education. In J. Domenech, J. Lloret, M.C. Vincent-Vela, E. de la Poza, E. Zuriaga (Eds.). *Advances in Higher Education*, (pp. 127-148). Valencia, Spain: Universitat Politecnica de Valencia.

Appendix A: Programming task (Microsoft Visual Basic 2010)

Creating a Hangman game using a combination of string manipulation we can create a very simple hangman game:

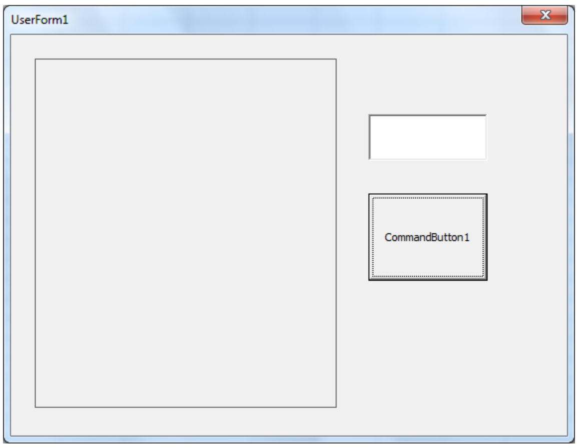
Concatenating
- Used to combine two strings to form one string
- e.g MSGBOX "You need " & total & " tiles"

Locate or **Find** function (**Instr** function in Visual Basic)
- Use to find a string within a string and return its location
- e.g. INSTR("ell","Hello") returns **2**

Len function
- Used to return the length of a string
- e.g. variable = LEN("Hello") returns **5**

LoadPicture function
- Used to load an image into an imagebox
- e.g. image1.Picture = LoadPicture("N:/My Pictures/Hangman.jpg")
- **You will need to save a jpg picture called Hangman in your pictures folder**

1. Load Visual Basic and add a TextBox, an image and a Button to the form:



2. Add the code on the next page to the button, test your program and then complete the following tasks.

Tasks:

1. Validate your code so only one letter can be input at a time.
2. Load an image of a developing hangman for each incorrect answer.
3. Try to prevent the same letter being guessed more than once

'global variables used to keep track of correct

```
'and incorrect answers
Dim intIncorrect As Integer
Dim intCorrect As Integer

Private Sub Button1_Click()
  'local variables used to store the word and letter
  Dim strLetter As String
  Dim strWord As String
  Dim intLength, intLocation As Integer
  'store the word
  strWord = TextBox1.Text
  'count how long the word is
  intLength = Len(strWord)
  'loop until 3 incorrect guesses or correct guesses equal word length
  Do
    'Store a letter
    strLetter = InputBox("Please enter a letter")
    'check the location of that letter
    intLocation = InStr(strWord, strLetter)
    'If letter is incorrect then 0 is returned
    If intLocation <> 0 Then
      MsgBox ("You guessed correctly with: " & strLetter & " in position: " &
intLocation)
      'add one to the correct total
      intCorrect = intCorrect + 1
    Else
      MsgBox ("you guessed incorectly with: " & strLetter)
      'add one to the incorrect total
      intIncorrect = intIncorrect + 1
    End If
  Loop Until intIncorrect = 3 Or intCorrect = intLength

    'If correct equals the word length then you won
  If intCorrect = intWordLength Then
    MsgBox ("Game Over - you Won!")
  Else
    MsgBox ("Game Over - you lose!")
    'Load the image of a hangman
    Me.PictureBox1.ImageLocation = ("n:/My Pictures/Hangman.jpg")
  End If

End Sub
```

Appendix B: **Ethics documentation**

The ethics documentation is composed of six components: an ethics form; a risk assessment; student consent form; research assistant consent form; parental consent form; and a school consent form. The University of Reading's Ethical Committee gave ethical approval on 05/04/2015.

University of Reading IoE Approval Form A (version September 2013)

Tick one:

Staff project: ___ PhD: ____ **EdD**: ✓

Name of applicant (s): Mr Terence Norman McAdams

Title of project: Gender and Computer Programming: Teaching and learning strategies designed to increase the engagement of girls

Name of supervisor (for student projects): Dr. Vincent Trakulphadekrai,
Dr. Daisy Powell

Please complete the form below including relevant sections overleaf.

	YES	NO
Have you prepared an Information Sheet for participants and/or their parents/carers that:		
a) explains the purpose(s) of the project	✓	
b) explains how they have been selected as potential participants	✓	
c) gives a full, fair and clear account of what will be asked of them and how the information that they provide will be used	✓	
d) makes clear that participation in the project is voluntary	✓	

e) explains the arrangements to allow participants to withdraw at any stage if they wish	✓	
f) explains the arrangements to ensure the confidentiality of any material collected during the project, including secure arrangements for its storage, retention and disposal	✓	
g) explains the arrangements for publishing the research results and, if confidentiality might be affected, for obtaining written consent for this	✓	
h) explains the arrangements for providing participants with the research results if they wish to have them	✓	
i) gives the name and designation of the member of staff with responsibility for the project together with contact details, including email . If any of the project investigators are students at the IoE, then this information must be included and their name provided	✓	
k) explains, where applicable, the arrangements for expenses and other payments to be made to the participants	N/A	
j) includes a standard statement indicating the process of ethical review at the University undergone by the project, as follows: 'This project has been reviewed following the procedures of the University Research Ethics Committee and has been given a favourable ethical opinion for conduct'.	✓	
k) includes a standard statement regarding insurance: "The University has the appropriate insurances in place. Full details are available on request".	✓	
Please answer the following questions		
1) Will you provide participants involved in your research with all the information necessary to ensure that they are fully informed and not in any way deceived or misled as to the purpose(s) and nature of the research? (Please use the subheadings used in the example information sheets on blackboard to ensure this).	✓	

2) Will you seek written or other formal consent from all participants, if they are able to provide it, in addition to (1)?	✓		
3) Is there any risk that participants may experience physical or psychological distress in taking part in your research?		✓	
4) Have you taken the online training modules in data protection and information security (which can be found here: http://www.reading.ac.uk/internal/imps/Staffpages/imps-training.aspx)?	✓		
5) Have you read the Health and Safety booklet (available on Blackboard) and completed a Risk Assessment Form to be included with this ethics application?	✓		
6) Does your research comply with the University's Code of Good Practice in Research?	✓		
	Yes	No	N/A
7) If your research is taking place in a school, have you prepared an information sheet and consent form to gain the permission in writing of the head teacher or other relevant supervisory professional?	✓		
8) Has the data collector obtained satisfactory DBS clearance?	✓		
9) If your research involves working with children under the age of 16 (or those whose special educational needs mean they are unable to give informed consent), have you prepared an information sheet and consent form for parents/carers to seek permission in writing, or to give parents/carers the opportunity to decline consent?	✓		
10) If your research involves processing sensitive personal data, or if it involves audio/video recordings, have you obtained the explicit consent of participants/parents?			✓
11) If you are using a data processor to subcontract any part of your research, have you got a written contract with that contractor which (a) specifies that the contractor is required to act only on your instructions, and (b) provides			✓

for appropriate technical and organisational security measures to protect the data?			
12a) Does your research involve data collection outside the UK?	✓		
12b) If the answer to question 11a is "yes", does your research comply with the legal and ethical requirements for doing research in that country?			✓
13a. Does the proposed research involve children under the age of 5?		✓	
13b. If the answer to question 12a is "yes": My Head of School (or authorised Head of Department) has given details of the proposed research to the University's insurance officer, and the research will not proceed until I have confirmation that insurance cover is in place.		✓	
If you have answered YES to Question 3, please complete Section B below			

PLEASE COMPLETE **EITHER SECTION A OR B** AND PROVIDE THE DETAILS REQUIRED IN SUPPORT OF YOUR APPLICATION, THEN SIGN THE FORM (SECTION C)

A: My research goes beyond the 'accepted custom and practice of teaching' but I consider that this project has **no** significant ethical implications.	✓
Give a brief description of the aims and the methods (participants, instruments and procedures) of the project in up to 200 words. Attach any consent form, information sheet and research instruments to be used in the project (e.g. tests, questionnaires, interview schedules). Please state how many participants will be involved in the project: *This form and any attachments should now be submitted to the Institute's Ethics Committee for consideration. Any missing information will result in the form being returned to you.*	

The purpose of this thesis is to investigate the programming aspirations of secondary school South Korean students (years 8 & 9) and explain what factors might influence their decision to choose IGCSE Computer Science.

All students are taught computer programming using a variety of techniques: through the use of flowcharts, problem-based learning (graphical and text-based), and storytelling. All students will be asked to complete two short, practical, programming assessments, one in term 2 and one in term 3. Both assessments will have three tasks, each one progressively more difficult, that requires the students to solve a problem by writing the solution using a programming language.

Students will complete three short, secure, anonymous online questionnaires: the first will ask questions assessing student confidence in programming, and whether they have the necessary skills to complete the first assessment prior to taking it. They will also be asked whether they believe boys will outperform girls or visa versa; the second will ask questions about the effectiveness of the learning methodologies (flowcharts, problem-solving and narrative) and elaborate on what aspects of computer programming they like/dislike, find difficult/easy; and the third will ask questions about their perception of the computing industry, how often they play computer games, and their desire to continue learning computer programming.

Finally, a small group of student volunteers will be asked to take part in a 20 minute group interview/discussion to help identify which teaching methods were the most interesting and effective. The interview questions will be largely formed from the results of the questionnaires.

Sample: 194 South Korean nationals (81 girls, 113 boys) from Year 8 and Year 9.

B: I consider that this project **may** have ethical implications that should be brought before the Institute's Ethics Committee.	

Please provide all the further information listed below in a separate attachment.

1. title of project
2. purpose of project and its academic rationale
3. brief description of methods and measurements
4. participants: recruitment methods, number, age, gender, exclusion/inclusion criteria
5. consent and participant information arrangements, debriefing (attach forms where necessary)
6. a clear and concise statement of the ethical considerations raised by the project and how you intend to deal with then.
7. estimated start date and duration of project

This form and any attachments should now be submitted to the Institute's Ethics Committee for consideration. Any missing information will result in the form being returned to you.

C: SIGNATURE OF APPLICANT:

I have declared all relevant information regarding my proposed project and confirm that ethical good practice will be followed within the project.

Signed: **Print Name:** Terence Norman McAdams **Date:** 04/01/2015

STATEMENT OF ETHICAL APPROVAL FOR PROPOSALS SUBMITTED TO THE INSTITUTE ETHICS COMMITTEE

This project has been considered using agreed Institute procedures and is now approved.

Signed: **Print Name** Andy Kempe **Date 5th April 2015**
(IoE Research Ethics Committee representative)*

* A decision to allow a project to proceed is not an expert assessment of its content or of the possible risks involved in the investigation, nor does it detract in any way from the ultimate responsibility which students/investigators must themselves

* A decision to allow a project to proceed is not an expert assessment of its content or of the possible risks involved in the investigation, nor does it detract in any way from the ultimate responsibility which students/investigators must themselves have for these matters. Approval is granted on the basis of the information declared by the applicant.

Risk Assessment Form

Select one:

Staff project: ☐ **PGR project:** ☑ MA/UG project: ☐

Name of applicant (s): Mr Terence Norman McAdams

Title of project: Gender and Computer Programming: Teaching and learning strategies designed to increase the engagement of girls

Name of supervisor (for student projects): Dr. Vincent Trakulphadetkrai, Dr. Daisy Powell

A: Please complete the form below

Brief outline of Work/activity:	I plan to gather student data (Years 7, 8, and 9) at various stages until June 2015. Initially, I would like to investigate if there is any gender difference with computer programming anxiety and confidence before determining if there is any gender difference between the preferred styles of learning. Finally, I will look at gender performances differences and desire/intention to continue studying Computer Science at IGCSE (Year 10). Programming in Python is new to all of the students, which is why 8 and 9 will be given the same assessments. **Assessment 1 (30 minutes) and the CPCA Questionnaire – April 2015:** Male and female students will complete an online computing confidence anxiety programming questionnaire and comment on whether they will be able to complete the three programming tasks. Students will then complete programming tasks using Python and a comparison between perceived achievement and actual achievement will be conducted. We will use Google Form, computers and Python. Years 8 and 9 will all take the Python turtle.

	Preferred Computer Programming Learning Style Questionnaire (Appendix O) **– May 2015:** Half of the students will complete a 7/8-week programme of collaborative and/or cooperative storytelling in Python (years 8 and 9) and Microsoft Visual Basic 2010 (years 7). They will then be asked to complete an online questionnaire on their perceptions of learning in this way. Male and females students are taught separately. During this period, the other half of students will be learning through problem solving. **Assessment 2 – June 2015:** End of year online programming assessment. **Student Perceptions on Computer Science Questionnaire (30 minutes) – June 2015**: Students will be given a final questionnaire on their perceptions of computer programming and the industry itself. Two small groups of 20 students in total, one group of 10 females and one group of 10 males, will be asked to attend a group discussion session. These 20 students will be selected at random from those that have volunteered.

Where will data be collected?	One international secondary school

Significant hazards:	None identified. The school has recently undergone FOBISIA and CIS accreditation and passed the health and safety requirements. There are no trailing cables and surge protectors are installed. In the past month, the network manager has recently completed a safety check on all computer equipment.

Who might be	N/A

exposed to hazards?	

Existing control measures:	The computer rooms comply with the school's Health & Safety policy, which has been approved by the Council of International Schools.

Are risks adequately controlled:	Yes ☑ No ☐

If NO, list additional controls and actions required:	Additional controls	Action by:

B: SIGNATURE OF APPLICANT:

I have read the Heath and Safety booklet posted on Blackboard, and the guidelines overleaf. I have declared all relevant information regarding my proposed project and confirm risks have been adequately assessed and will be minimized as far as possible during the course of the project.

Signed: **Print Name** Mr Terence Norman McAdams **Date**: 04 January 2014

STATEMENT OF APPROVAL TO BE COMPLETED BY SUPERVISOR (FOR UG AND MA STUDENTS) **OR** BY IOE ETHICS COMMITTEE REPRESENTATIVE (FOR PGR AND STAFF RESEARCH).

This project has been considered using agreed Institute procedures and is now approved.

* A decision to allow a project to proceed is not an expert assessment of its content or of the possible risks involved in the investigation, nor does it detract in any way from the ultimate responsibility which students/investigators must themselves have for these matters. Approval is granted on the basis of the information declared by the applicant.

Student Consent Form

What happens next?

Your parents have been sent a letter asking for their permission for you to take part in this project.

We will check with you before we ask you to complete questionnaires and ensure you are happy to help us with our project.

If you have any questions please speak to your tutor.

Or

contact

Mr Terry McAdams

B211

tmcadams@nlcsjeju.kr

The research team is:

Researcher

Mr Terry McAdams

University Supervisors

Dr. Vincent Trakulphadetkrai

Dr Daisy Powell

This project has been reviewed following the procedures of the University of Reading Research Ethics Committee and has been given a favourable ethical opinion for conduct.

University of Reading

Institute of Education
London Road Campus
RG1 5EX

Research Project:

Finding the best ways for you to **learn computer programming**

Information Sheet

We are doing a project to understand what are the best ways for you to learn computer program. We would like you to help us with the project. We have already asked your parents, and the Principal if they are happy for you to help us.

Why have I been invited to take part?

You have been invited to take part because you are in Year 7, 8 and 9, because you learn computer programming, and because the School is interested in knowing how we can help you learn better.

What will I have to do if I agree to take part?

1. *We will use the data from the two tests that you will take in Computer Science.*
2. *We will ask you to complete three anonymous, online questionnaires on how you learn computer programming and whether you want to continue when you are older.*
3. *We will ask 20 of you to volunteer to do a group discussion on what you think about computer science and programming.*

These will all take place during your Computer Science lessons. Thank you very much to the ICT & English departments, tutors, and the School.

Will it affect my school levels?

Taking part is separate to your lessons and will not change your grades/levels at all.

Will anyone know about my answers?

- Your answers will be anonymous; no one will be able to identify you. Each student in Year 7, 8, and 9 will be given a different ID and answers will be put next to that ID.
- We will tell the school, you, and your parents the general results (how everyone has done), not individual results.

Will it help me if I take part?

Your answers will help your teachers to know the best ways to teach you and hopefully it will make learning to computer programming easier and more enjoyable.

Do I have to take part?

You will need to do the two tests, but if you wish, we won't include your results. Also, you can stop helping us with our project at any time, without giving a reason. Just email me or ask your tutor or parents to email me if you want to stop.

Thank you.

Research Assistant Information Sheet

Research Project: Gender and Computer Programming: Teaching and learning strategies designed to increase the engagement of girls

Researcher: Mr. Terence McAdams

Supervisors: Dr. Vincent Trakulphadetkrai
Dr. Daisy Powell

Dear colleague,

I would like to invite you to take part in a research study to explore the process of how boys and girls learn computing programming, in order to identify strategies to improve student learning. You are being asked to assist in one of two roles: 1) Supervise questionnaire competition and programming assessment; 2) Conduct a group interview.

What is the study?

The study is being conducted as a research project towards an Educational Doctorate. There are comparatively few females, relative to males, studying computer programming at university and employed in the software engineering industry so this study is investigating if this lack of interest in computer programming exists in secondary school.

There have been a number of reasons cited for this imbalance, including: a male dominated environment; a lack of female role models; teaching strategies that suit male learners; and a lack of socialisation in computer programming. Very little research has looked at the perceptions of school children and last year only 5 girls, compared to 34 boys, chose to continue their programming education into year 10. As the Head of Computer Science at NLCS Jeju, I feel that it is my obligation to investigate this phenomenon and try and to address this imbalance.

Why have you been chosen to take part?

There are two distinct roles in the assistance of this research: 1) Questionnaire and assessment supervision; and, 2) Group interview supervision.

1) Questionnaire and assessment supervision

You have been selected because you teach Computer Science boys' and girls' classes at Key Stage 3 or are covering a lesson during one of the assessments.

2) Group interview supervision

You do not teach Computer Science and therefore the students will be more comfortable discussing their learning and opinions of computer programming with you.

Do I have to take part?

It is entirely up to you whether you agree to assist in the gathering of data. You may also withdraw your consent to participation at any time during the project, without any repercussions to you. If you wish to do this, simply contact the researcher or supervisors, via the emails above.

What will happen if I take part?

All students will be taught computer programming using a variety of techniques: through the use of flowcharts, problem-based learning, and storytelling. All students will be asked to complete two short, practical, programming assessments, one in term 2 and one in term 3. Both assessments will have three tasks, each one progressively more difficult, that requires the students to practically solve a problem by writing the solution using a programming language. Each of the two assessments will last 30 minutes and will take place during a timetabled lesson. These assessments are used to assess learning and will influence the end of term grades.

Only the students, who are taking part in the research, will be asked to complete three anonymous and secure online questionnaires. The first questionnaire will include questions assessing student confidence in completing the first assessment prior to taking the assessment itself. It will also include questions on whether the students believe boys will outperform girls or vice-versa. The second questionnaire will ask questions about the effectiveness of the learning methodologies (flowcharts, problem-solving and narrative) and elaborate on what aspects of computer programming they like/dislike, find difficult/easy. The third and final questionnaire will ask questions about their perception of the computing industry and their desire to continue learning computer programming.

Finally, a small group of student volunteers will be asked to take part in a 20-minute group interview/discussion to help identify which teaching methods were the most interesting and effective. An impartial colleague will conduct this interview to ensure that the students are free to express their opinions without consequences. These twenty students will be selected at random from those who have volunteered. The discussion topics will include learning strategies and gender differences in computer programming. The topics will be informed by the questionnaire data and will be recorded using the recording app on a school-owned smart phone. This recording will be kept secure and deleted once it has been transcribed by the impartial colleague.

Assessment will be administered during Computer Science timetabled lessons, except for the group interview, which will take place one lunchtime. You must ensure that all participants in the group interviews are anonymous to the researcher by recording yourself repeating the students' comments. Once this is done you will be asked to delete the original recordings. All other data (questionnaire and assessment data) are stored electronically and you are not required to do anything with this data.

If you agree to participate, I will explain your role in detail prior to the assessments.

What are the risks and benefits of assisting?

Your role in this study will remain confidential but you will need to ensure that all of the data is kept securely. Participants in similar studies have found it interesting to take part and you may decide to conduct your own research at a later date.

What will happen to the data?

Any data collected will be held in strict confidence and no real names will be used in this study or in any subsequent publications. The records of this study will be kept private. No identifiers linking you, the children or the school to the study will be included in any sort of report that might be published.

Participants will be assigned a number and referenced to that number in all records. Research records will be stored securely in a locked filing cabinet and on a password-protected computer. Only the research team will have access to the records. After five years, the data will be destroyed securely once the findings of the study are written up. The results of the study may be presented at national and international conferences, and in written reports and articles. You can be sent electronic copies of these publications, if you wish.

What happens if I change my mind?

You can change your mind at any time without any repercussions.

What happens if something goes wrong?

In the unlikely case of concern or complaint, you can contact: Dr. Vincent Trakulphadetkrai, University of Reading

Where can I get more information?

If you would like more information, please contact either the researcher, Mr T. McAdams, or Dr Vincent Trakulphadetkrai. Thank you for agreeing to take part in the study. To make it official, please complete the enclosed consent form and return it to Mr T. McAdams.

This project has been reviewed following the procedures of the University Research Ethics Committee and has been given a favourable ethical opinion for conduct. The University has the appropriate insurances in place. Full details are available on request.

Thank you for your time.

Yours sincerely

Terry McAdams

RESEARCH ASSISTANT CONSENT FORM – NORTH LONDON COLLEGIATE SCHOOL JEJU

- I have read the Information Sheet about the project and received a copy of it.

- I understand what the purpose of the project is and what is required of me. All my questions have been answered.

Name of colleague:

Name of school: North London Collegiate School Jeju

Please tick as appropriate:

☐ I consent to my involvement in the research project as outlined in the Information Sheet

Signed: ____________________________

Date: ____________________________

Information Sheet For Parents

Research Project: Gender and Computer Programming: Teaching and learning strategies designed to increase the engagement of girls

Researcher: Mr. Terence McAdams

Supervisors: Dr. Vincent Trakulphadetkrai
Dr. Daisy Powell

I would like to invite your child to take part in a study to explore the process of how boys and girls learn computing programming, in order to identify strategies to improve student learning.

What is the study?

The study is being conducted as a research project towards an Educational Doctorate. The rationale behind the study is that there are comparatively few females, relative to males, studying computer programming at university and employed in the software engineering industry, so this investigation is trying to determine if this lack of interest in computer programming begins in secondary school.

There have been a number of reasons cited for an imbalance, including: biological and psychological differences, a male dominated environment; a lack of female role models; teaching strategies that suit male learners; and a lack of socialisation in computer programming. Very little research has looked at the perceptions of school children and last year it was disappointing to see only 5 girls, compared to 34 boys, chose to continue with Computer Science in year 10. As the Head of Computer Science at NLCS Jeju, I feel that it is my obligation to investigate this phenomenon and try and to address this imbalance.

Why has my child been invited to take part?

Your child has been invited to take part in the project because he/she is in Year 8 or 9. All learners in KS3 are taught by Mr. N. Scarlett or Mr. T. McAdams and are being invited to take part.

Do I have to let my child take part?

Taking part in the research is entirely voluntary. You may also withdraw your consent to participation at any time during the project, without any repercussions to you or your child, by contacting the researcher, Mr. T. McAdams.

What will happen if my child takes part?

Regardless of whether your child takes part in the study, all students will be taught computer programming using a variety of techniques: through the use of flowcharts, problem-based learning, and storytelling. As part of the curriculum, all students are required to complete two short, practical, programming assessments, one in term 2 and one in term 3. Both assessments will have three tasks, each one progressively more difficult, that requires the students to practically solve a problem by writing the solution using a programming language. Each of the two assessments will last 30 minutes and will take place during a timetabled lesson. These assessments are used to assess learning and will influence the end of term grades. Only the assessment data of those students, who are taking part in the research, will be used as part of the study. The data from those that opt-out will not be included in the research.

Only the students, who are taking part in the research, will be asked to complete three anonymous and secure online questionnaires. The first questionnaire will include questions assessing student confidence in completing the first assessment prior to taking the assessment itself. It will also include questions on whether the students believe boys will outperform girls or vice-versa. The second questionnaire will ask questions about the effectiveness of the learning methodologies (flowcharts, problem-solving and narrative) and elaborate on what aspects of computer programming they like/dislike, find difficult/easy. The third and final questionnaire will ask questions about their perception of the computing industry and their desire to continue learning computer programming.

Finally, a small group of student volunteers will be asked to take part in a 20-minute group interview/discussion to help identify which teaching methods were the most interesting and effective. An impartial colleague will conduct this interview to ensure that the students are free to express their opinions without consequences. These twenty students will be selected at random from those who have volunteered. The discussion topics will include learning strategies and gender differences in computer programming. The topics will be informed by the questionnaire data and will be recorded using the recording app on a school-owned smart phone. This recording will be kept secure and deleted once it has been transcribed by the impartial colleague.

What are the risks and benefits of taking part?

The information your child gives will remain confidential. Only the teachers conducting the assessments and the research team identified at the top of this letter will see the data. Neither you, your child or the school will be identifiable in any published report resulting from the study. Taking part, or refusing to take part, will in no way influence the grades your child receives at school. Furthermore, information about individuals will not be shared with the school.

Participants in similar studies have found it interesting to take part. We anticipate that the findings of the study will be useful for educators teaching computer programming in schools. An electronic copy of the published findings of the study can be made available to you through the school.

What will happen to the data?

Any data collected will be held in strict confidence and no real names will be used in this study or in any subsequent publications. The records of this study will be kept private. No identifiers linking your child to the study will be included in any sort of report that might be published.

You child's assessment data will be assigned a number and will be referred to by that number in all records. Questionnaire data is completely anonymous and so it will not be possible to identify your child's answers from any other. Should your child be one of the twenty volunteers for the group interview, their name will not be recorded on any documentation to ensure anonymity. Research records will be stored securely in a locked filing cabinet and on a password-protected computer and only the research team will have access to the records. The data will be destroyed securely once the findings of the study are written up, after five years. If you would like a brief summary of the findings, you can let me know directly using the contact details above.

What happens if I change my mind?

You can change your mind at any time without any repercussions. If you change your mind after data collection has ended, we will discard your child's data.

Who has reviewed the study?

This project has been reviewed following the procedures of the University Research Ethics Committee and has been given a favourable ethical opinion for conduct. The University has the appropriate insurances in place. Full details are available on request.

What happens if something goes wrong?

In the unlikely case of concern or complaint, you can contact: Dr. Vincent Trakulphadetkrai, University of Reading.

Where can I get more information?

If you would like more information, please contact Mr. Terence McAdams.

What do I do next?

I do hope that you will agree to take part in the study. If not, please complete the attached opt-out form. If you have any queries or wish to clarify anything about the study, please feel free to contact me, or my supervisor, using the email addresses given above.

Thank you for your time.

Yours sincerely

Terry McAdams

OPT-OUT FORM FOR PARENTS

Research Project: Gender and Computer Programming: Teaching and learning strategies designed to increase the engagement of girls

Researcher: Mr. Terence McAdams

Supervisors: Dr. Vincent Trakulphadetkrai
Dr. Daisy Powell

I have received a copy of it the Research Information Sheet and have read the contents. I understand the purpose of the project and what is required of my child. All my questions have been answered.

I have had explained to me the purposes of the project and what will be required of my child, and any questions have been answered to my satisfaction. I agree to the arrangements described in the Information Sheet in so far as they relate to my child's participation.

I understand that my child will be taking Computer Science assessments anyway and I give permission for their results to be included in the research project.

I understand that allowing my child's work to be included in the project is entirely voluntary and that I have the right to withdraw my child from the project any time, without giving a reason and without repercussions.

I have received a copy of this Opt-Out Form and the accompanying Information Sheet.

I understand that if I do not wish my child's computer programming assessment data to be used in the project, I must return the lower part of this sheet, otherwise my child's computer programming assessment data will be used in the project.

I understand that if I do not wish my child to complete the three anonymous questionnaires (1. Computer programming confidence; 2. Preferred learning style; 3. Desire to continue learning), I must return the lower part of this sheet, otherwise my child will be asked to complete three anonymous, secure, online questionnaires and their answers will be used in the research project.

Only to be returned if permission is NOT given:

I DO NOT consent to my child's work being used in the project as detailed in the Information Sheet.

Name of Parent:	
Name of Child:	
Signed:	
Date:	

Information Sheet for the School Principal

Research Project: Gender and Computer Programming: Teaching and learning strategies designed to increase the engagement of girls

Researcher: Mr. Terence McAdams

Supervisors: Dr. Vincent Trakulphadetkrai
Dr. Daisy Powell

Dear Sir

I would like to invite your school to take part in a research study to explore the process of how boys and girls learn computing programming, in order to identify strategies to improve student learning.

What is the study?

The study is being conducted as a research project towards an Educational Doctorate. There are comparatively few females, relative to males, studying computer programming at university and employed in the software engineering industry so this study is investigating if this lack of interest in computer programming begins in secondary school.

There have been a number of reasons cited for this imbalance, including: a male dominated environment; a lack of female role models; teaching strategies that suit male learners; and a lack of socialisation in computer programming. Very little research has looked at the perceptions of school children and last year only 5 girls, compared to 34 boys, chose to continue their programming education into year 10. As the Head of Computer Science at NLCS Jeju, I feel that it is my obligation to investigate this phenomenon and try and to address this imbalance.

Why has this school been chosen to take part?

This school has been selected because it has separate boys' and girls' classes at Key

Stage 3 and Computer Science is a compulsory subject at this age group.

Does the school have to take part?

It is entirely up to you whether you give permission for the school to participate. You may also withdraw your consent to participation at any time during the project, without any repercussions to you. If you wish to do this, simply contact the researcher or supervisors, via the emails above.

What will happen if the school takes part?

With your agreement, all students will be taught computer programming using a variety of techniques: through the use of flowcharts, problem-based learning, and storytelling. All students will be asked to complete two short, practical, programming assessments, one in term 2 and one in term 3. Both assessments will have three tasks, each one progressively more difficult, that requires the students to practically solve a problem by writing the solution using a programming language. Each of the two assessments will last 30 minutes and will take place during a timetabled lesson. These assessments are used to assess learning and will influence the end of term grades.

Only the students, who are taking part in the research, will be asked to complete three anonymous and secure online questionnaires. The first questionnaire will include questions assessing student confidence in completing the first assessment prior to taking the assessment itself. It will also include questions on whether the students believe boys will outperform girls or vice-versa. The second questionnaire will ask questions about the effectiveness of the learning methodologies (flowcharts, problem-solving and narrative) and elaborate on what aspects of computer programming they like/dislike, find difficult/easy. The third and final questionnaire will ask questions about their perception of the computing industry and their desire to continue learning computer programming.

Finally, a small group of student volunteers will be asked to take part in a 20-minute group interview/discussion to help identify which teaching methods were the most interesting and effective. An impartial colleague will conduct this interview to ensure that the students are free to express their opinions without consequences. These twenty students will be selected at random from those who have volunteered. The discussion topics will include learning strategies and gender differences in computer programming. The topics will be informed by the questionnaire data and will be recorded using the recording app on a smart phone. This recording will be kept secure and deleted once it has been transcribed by the impartial colleague.

Assessment would be administered during Computer Science timetabled lessons, and overseen by Mr. Terry McAdams. Any additional staff administering assessment will be given careful guidance on how to conduct the data collection. Additionally, it would be helpful to have access to student entrance assessment data (e.g. Wechsler intelligence scores and English as an Additional Language data).

This study may lead on to further study to identify the longitudinal effects of learning computer programming.

If you agree to the school's participation, I will seek further consent from parents/carers and the children themselves, as well as from two colleagues who will act as research assistants and will be administering assessment and conducting the focus group interview.

What are the risks and benefits of taking part?

The information given by participants in this study will remain confidential. Only the teachers conducting the assessments and the research team identified at the top of this letter will see the data. Neither you, the children or the school will be identifiable in any published report resulting from the study. Information about individuals will not be shared with the school.

Participants in similar studies have found it interesting to take part. It is anticipated that the findings of the study will be useful for teachers of programming in Secondary Schools.

What will happen to the data?

Any data collected will be held in strict confidence and no real names will be used in this study or in any subsequent publications. The records of this study will be kept private. No identifiers linking you, the children or the school to the study will be included in any sort of report that might be published.

Participants will be assigned a number and referenced to that number in all records. Research records will be stored securely in a locked filing cabinet and on a password-protected computer. Only the research team will have access to the records. After five years, the data will be destroyed securely once the findings of the study are written up. The results of the study may be presented at national and international conferences, and in written reports and articles. You can be sent electronic copies of these publications, if you wish.

What happens if I change my mind?

You can change your mind at any time without any repercussions. If you change your mind after data collection has ended, we will discard the school's data.

What happens if something goes wrong?

In the unlikely case of concern or complaint, you can contact: Dr. Vincent Trakulphadetkrai, University of Reading.

Where can I get more information?

If you would like more information, please contact either the researcher, Mr. T. McAdams, or Dr. V. Trakulphadetkrai. Thank you for agreeing to take part in the study. To make it official, please complete the enclosed consent form and return it to Mr. T. McAdams.

This project has been reviewed following the procedures of the University Research Ethics Committee and has been given a favourable ethical opinion for conduct. The University has the appropriate insurances in place. Full details are available on request.

Thank you for your time.

Yours faithfully

Mr. T. McAdams

HEAD TEACHER CONSENT FORM – NORTH LONDON COLLEGIATE SCHOOL JEJU

- I have read the Information Sheet about the project and received a copy of it.

- I understand what the purpose of the project is and what is required of me. All my questions have been answered.

Name of Principal:

Name of school:

Please tick as appropriate:

☐ I consent to the involvement of my school in the project as outlined in the Information Sheet

Signed: ______________________

Date: ______________________

Appendix C: Tutorial for learning JavaScript programming through short tasks

Tutorial Learning Objectives

- Using selection (if…elif…else)
- Using loops (for loop)
- Using substring to remove parts of a string
- Using concatenation to combine strings

The substring function reduces a string into a smaller size. The first number is the start position beginning at 0. The second number is the finish position:

- “Hello”.substring(0,1) will return “H”
- “Hello”.substring(2,4) will return “ll”

1. Open Text Wrangler or Atom.io and enter the following:

```
<!DOCTYPE html>
<html>
<body>

<h1>Removing * from a string</h1>

<p>Enter some text into the textbox:</p>
<input id="demo" type="text" value="com*put**er*">
<button type="button" onclick="getRid()">Submit</button>

<p id="output"></p>

<script>
// a function called remove
function getRid() {
   // create a variable to store the new sentence
   var newsent = "";
   // Get the value of the input field with id="demo"
   sent = document.getElementById("demo").value;
   //loop from 0 to the length of the string
   for(var i=0; i<sent.length;i++){
      letter = sent.substring(i,(i+1))
      // if the letter is not equal to "*" add the letter to newsent
      if(letter !== "*"){
          newsent += letter;
      }
    }
    document.getElementById("output").innerHTML = newsent;
}
</script>

</body>
</html>
```

2. Save the program as substring.html and run it.

3. Test the program in your web browser software.

Task

1. Change the sentence from the default value to something else and test that it removes the * symbol
2. Adapt the program to remove a different character (e.g. &) in addition to the * symbol.

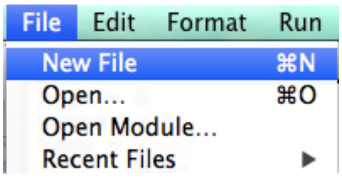
Appendix D: Tutorial for learning Python through short tasks

Learning Objectives

- Using selection (if…elif…else)
- Using Boolean logic: and, or
- Using random numbers through the random library
- Converting binary to ASCII character codes and visa versa

1. Launch Python Idle by double-clicking on it:

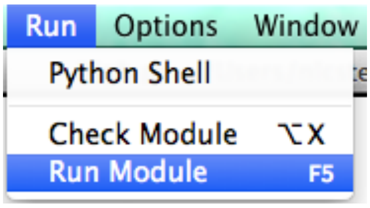
2. Select the **File | New File** menu:



3. Enter the following commands into the Window **(Note: You do not need to enter the code in red because these are comments):**

```
import random #imports the random library so randint can be used

#Ask the user to enter their choice
user = input("Enter R, P or S for Rock, Paper or Scissors: ")

comp_int = random.randint(1,3)
if comp_int == 1:
    comp = "R"
elif comp_int == 2:
    comp = "P"
else:
    comp = "S"
print ("Computer chose: " + comp)

if (user == "R" and comp == "S") or (user == "P" and comp == "R"):
    print("User wins")
elif (user == "S" and comp == "R") or (user == "R" and comp == "P"):
    print("Computer wins")
else:
    print("It is a draw")
```

4. Now run the program by selecting the **Run | Run Module** menu:



5. The program should ask you to enter a choice and will tell you what the computer chose and who won:

```
Enter R, P or S for Rock, Paper or Scissors: R
Computer chose: S
User wins
```

Task 1
What happens if the user doesn't enter R, P or S? Try entering **Rock** instead of **R**

Use the following statement to extract the first character and then test to check if the user enters Rock it will still work:

user = user[0:1]

Task 2
What happens if the user enters a lowercase letter instead of an uppercase letter. Try entering this code to see if the user entered a lowercase letter and if so convert to uppercase e.g. ord('r') = 114 and 114 - 32 = 82 and chr(82) = R

```
if ord(user) > 90:
    user = chr(ord(user) - 32)
```

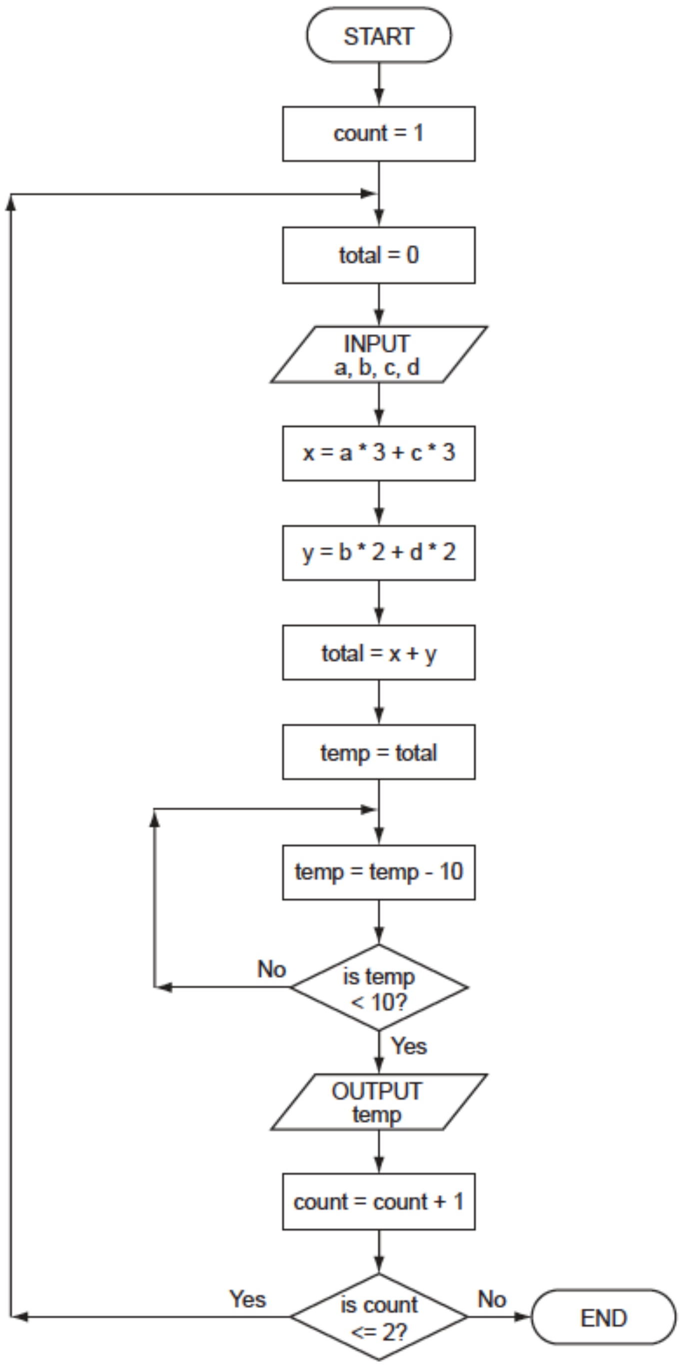
Task 3
There is a **logic error** in the code. This means that the program works but not perfectly. If the user enters P (Paper) and the computer chooses S (Scissors) then it informs you that it is a draw. Adapt the code to ensure that Scissors beats Paper.

Advanced Extension Task 3
Watch this clip from 'The Big Bang Theory'
http://www.youtube.com/watch?v=iapcKVn7DdY

Adapt the code so that the program works with R, P, S, L (Lizard), K (Spock)

Appendix E: Task for learning computer programming through flowcharts

Task: Trace the following flowchart using the headings shown below:



Complete the trace table for the following two sets of data:

(i) a = 5, b = 4, c = 1, d = 9

(ii) a = 5, b = 9, c = 4, d = 1

count	total	a	b	c	d	x	y	temp	OUTPUT

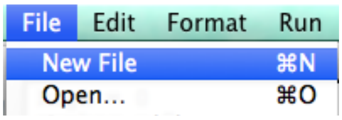
Appendix F: Tutorial for learning Python programming through storytelling

Learning Objectives

- Local and global variables
- Random numbers
- Iteration (while loops)
- Combining strings (concatenation)

1. Launch Python Idle by double-clicking on it:

2. Select the **File | New File** menu:



3. Enter the following commands into the Window to set your characters statistics and objects (e.g. glass, potion) for the game *(comments in red do not need to be entered)*:

```
# main program
import random
health = int(random.randint(10,20)) #Your health will be between 10 and 20
strength = int(random.randint(10,20))
glass = False #Variable to identify if you have collected the glass or not
loc1() #run the first location subroutine
```

4. We now need to enter a description for the first location so we will create a subroutine. Enter the following code:

```
def loc1(): # cliff edge subroutine called loc1
  global glass # let the subroutine know the glass and potion are public variables
  global potion
  repeat = True
  str1 = "You find yourself at the top of a Mountain, surrounded by a mist that makes
    visibility quite difficult. "
  str1 += "You look down and jump back realising that there is nothing between you and
    plummeting to the earth far below. "
  str1 += "In your sudden movement you notice that have stepped on a piece of thick glass.
    Pretty lucky you didn't cut yourself! "
  str1 += "Turning fully to make sense of your surroundings, you notice that there is a
    winding path behind you in a southerly direction."
  print(str1) # print the description
  print("\n")
```

```
while (repeat == True):
  response = input("? ")
  response = response.lower() #convert to lower case
  if (response.find("glass") != -1): #user entered a sentence with the word glass
    print("You bend down, pick up the thick glass and put it in your pocket for safe keeping. \n")
    glass = True
  elif (response.find("jump") != -1):
    print("Foolishly, you decide that playing this game is not worthwhile and leap to your doom. Game Over!")
    repeat = False
  elif(response.find("south") != -1) or (response.find("path") !=-1): #south or path takes you to loc2
    loc2()
  else:
    print("I don't see how that will help you....")
```

4. Run the program and test it a few times.

Tasks

1. In your groups plan your story by:
   - Choose the theme and overall idea for your story
   - Decide on the number and description of the locations e.g cliff edge
   - Identify the objects with which your character will interact e.g. glass
   - Identify the way your character interacts e.g. pick up glass, jump, south, path

2. Once you have your plan outlined then adapt the code above to fit your story.

Appendix G: Tutorial for learning JavaScript programming through storytelling

Learning Objectives

- Local and global variables
- Save data to local storage
- Calling Functions
- Selection (if statements)

1. Launch Text Wrangler or Atom.io.

2. Create a basic page and save it as **index.html**:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Text Adventure Project</title>
</head>
<body>
  <h1>The Jeju Adventure</h1>
  <section>
  <p id="story"> insert the story here </p>
  </section>
<hr>
<section>
  <p>Enter your instruction</p>
  <input id="command">
  <button type="button" onclick="action1()">Submit</button>
  <hr>
  <p id="output" style="color: red;"></p>
</section>
</body>
</html>
```

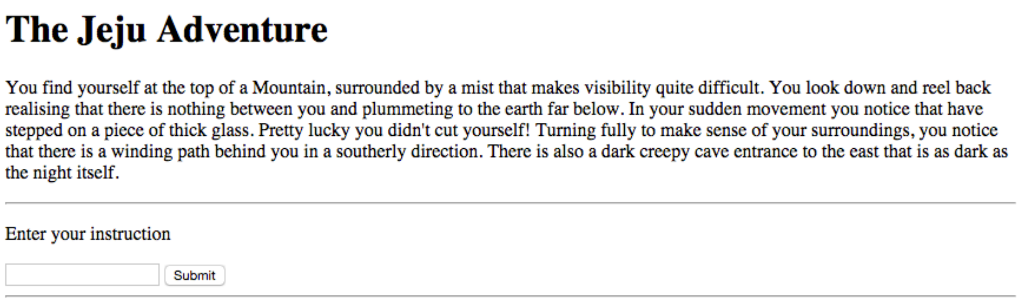
Explanation:

- The id attribute allows the individual element to be identified
- When the Submit button is clicked the onclick command runs the javascript action1() function (not yet created)

3. Replace the text ‘insert the story here’ by copying and pasting the following text into the paragraph element with the id of story:

   You find yourself at the top of a Mountain, surrounded by a mist that makes visibility quite difficult. You look down and reel back realising that there is nothing between you and plummeting to the earth far below. In your sudden movement you notice that have stepped on a piece of thick glass. Pretty lucky you didn't cut yourself! Turning fully to make sense of your surroundings, you notice that there is a winding path behind you in a southerly direction. There is also a dark creepy cave entrance to the east that is as dark as the night itself.

4. View the webpage in the browser window:



Cascading Style Sheets (CSS)

CSS enables the developer to add style to their web pages. There are three ways of linking CSS:

External File

<link rel="stylesheet" type="text/css" href="mystyle.css">

Internal

```
<style>
  p { font-family: Arial; }
</style>
```

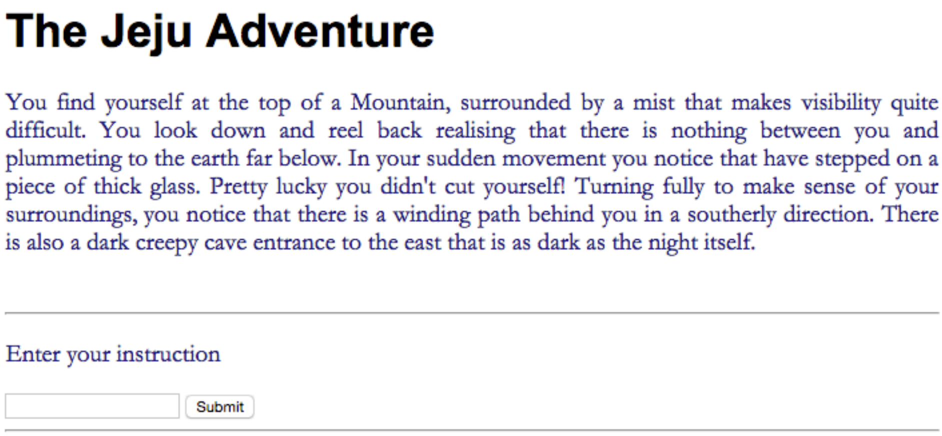
Inline

<p style="color:blue;">This is a paragraph.</p>

5. Add the following internal cascading style sheet inside the <HEAD> </HEAD> element just below the <TITLE </TITLE>:

```
<style>
  h1 {
    font-family: Arial;
    font-size: 26pt;
  }
  p {
    text-align: justify;
    font-family: garamond;
    font-size: 14pt;
    color: #222280;
  }
  body{
    width: 80%;
    margin-left: 10%;
    margin-right: 10%;
  }
  section {
    height: 160px;
  }
</style>
```

6. View the webpage in the browser window:



Task:

Add an image that matches the scenario or a background image using CSS:
http://www.w3schools.com/html/html_images.asp
http://www.w3schools.com/css/css_background.asp

JavaScript

This scripting language is the default client scripting language. It is interpreted by the web browser e.g. Safari, Google Chrome. See the following link for more details: http://www.w3schools.com/js/

7. Add the following script inside the <HEAD> </HEAD> element just after the <STYLE> </STYLE>:

```
<script>
// create the global variables
var health = 20;
var strength = 20;
var glass = false;
var potion = false;

  window.onload = function() {
    // store items to a file
    localStorage.setItem("health",health);
    localStorage.setItem("strength",strength);
    localStorage.setItem("glass","F"); // Item not obtained
    localStorage.setItem("potion","F"); // Item not obtained
  };
</script>
```

Explanation:

- The var command is creating variables that will be recognised within the subroutines (i.e. function). See http://www.w3schools.com/js/js_variables.asp
- The window.onload = function() automatically runs when the page is loaded. See http://www.w3schools.com/jsref/event_onload.asp
- The localStorage.setItem command stores the variable data to a file so that it can be carried over to other web pages. See http://www.w3schools.com/html/html5_webstorage.asp

8. Add the following javascript above the </script> element and below the };

```
function action1() {
  var user;
  var ans1 = "You bend down and pick up the glass!";
  var ans2 = "You walk towards the cave!";
  var ans3 = "You walk towards the path!";
  var ans4 = "I am not sure how that will help you!";

  // Get the value of the input field with id="numb"
  user = document.getElementById("command").value;
  // Look for keywords
  if (user.search("glass") != -1 && glass == false) {
     document.getElementById("output").innerHTML = ans1;
     glass = true;
     localStorage.setItem("glass","T"); //store glass : True
     document.getElementById("command").value = ""; //reset input field
     document.getElementById("command").focus(); //cursor in input field
  }
  else if (user.search("north") != -1){
     document.getElementById("output").innerHTML = "You can't go that way!";
     document.getElementById("command").value = ""; //reset input field
     document.getElementById("command").focus(); //cursor in input field
  }
  else if (user.search("cave") != -1 || user.search("east") != -1) {
     document.getElementById("output").innerHTML = ans2;
     location3()
  }
  else if (user.search("path") != -1 || user.search("south") != -1) {
     document.getElementById("output").innerHTML = ans3;
     location2()
  }
  else {
     document.getElementById("output").innerHTML = ans4;
  }
}
function location2(){
  window.open("location2.html", "_self");
}

function location3(){
  window.open("location3.html", "_self");
}
```

Explanation:

- The .search function looks for a word and returns -1 if it doesn't find it. See http://www.w3schools.com/jsref/jsref_search.asp
- The innerHTML command prints to an element. See http://www.w3schools.com/jsref/prop_html_innerhtml.asp
- The window.open command opens a webpage in the same way as a <a href=file.html>. The _self part refers to the same window rather than a new window. See http://www.w3schools.com/jsref/met_win_open.asp
- The location2.html webpage and location3.html webpage have not been created yet.

9. Run the web page and test it.

Task:

Add additional commands that require responses. For example, if the user types in west.

Advanced Task

Add a section with buttons for north, south, east and west. These will do the same as typing in these commands.

Appendix H: Tutorial for learning Python Turtle programming with visual design

Learning objectives

- Be able to create a subroutine that draws a triangle
- Be able to use random numbers and apply to shapes
- Be able to call a subroutine from another one to create a star
- Combine loops and subroutine calls to create a night sky

1. Create a New Window in Turtle and enter the following code (comments in read do not need to be entered):

```
import turtle #import the turtle library
import random #import the random numbers library
t = turtle.Turtle() #set t as the turtle
t.speed(5) #draw at a medium pace

def triangle(size):
    t.color("yellow") #set line colour
    t.fillcolor("yellow") #set fill colour
    t.begin_fill() #use before shape is drawn
    for a in range(3): #draw a triangle
        t.forward(size)
        t.right(120)
    t.end_fill() #use after the shape is drawn

size = random.randint(20,150) #random size for the shape
triangle(size) #draw triangle
```

2. Save your program as triangle.py and run it to see a randomly-sized solid yellow triangle:

3. Enter the following code to create a star subroutine that calls the triangle subroutine on two occasions to create a star:

```
def star(size):
    t.left(60) #choose the direction bearing 30 degrees
    triangle(size) #draw triangle
    t.left(30) #choose the direction bearing 0 degrees
    t.penup()
    t.forward(size - (size * 0.4)) #move forward 60% of size
    t.pendown()
    t.right(90) #choose the direction bearing 90 degrees
    triangle(size) #draw triangle
```

4. Replace the final statement **triangle(size)** of the original program with **star(size)** in order to call the star subroutine instead of the triangle subroutine:

```
size = random.randint(20,150) #random size for the shape
triangle(size) #draw triangle

star(size)
```

5. Save your program as **star.py** and run it (You should get a randomly-sized solid yellow star):

6. Place the final two statements inside a for loop so that fifty stars are created (notice that the range for the variable size has been reduced to 2, 15):

```
for b in range(50): #create 50 stars
    x = random.randint(-370,370) #random location x
    y = random.randint(100,280) #random location y
    t.penup()
    t.setposition(x,y) #position the star x, y
    t.pendown()
    size = random.randint(2,15) #random size of star
    star(size) #draw a star of random size
```

7. Save your program as **night.py** and run it (You should get 50 random stars):

Task:
Try improving on the drawing by adding a satellite and/or changing the background to a darker colour. Use the commands s = turtle.Screen() and s.bgcolor("black") for the background. Alternatively, try using hexadecimal colours to get a very dark blue as the night sky.

Appendix I: Tutorial for learning HTML5 / JavaScript programming with visual design

Learning objectives

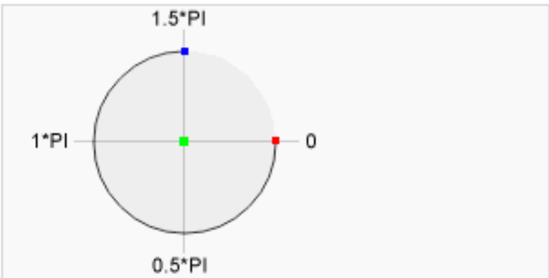
- Be able to create a webpage that uses the canvas API
- How to draw lines in the canvas
- How to draw shapes: rectangles, circles
- How to use colours and fill shapes

The Arc tool

context.arc(xCo-ord, yCo-ord ,radius, sAngle, eAngle, clockwise);

The arc tool has several parameters includes an x co-ordinate, y co-ordinate, arc radius, start angle in radians, ending angle in radians, true (anti-clockwise) or false (clockwise).

To draw a circle set the start angle at 0 and the end angle at 2 * Math.PI.



Center	arc(**100,75**,50,0*Math.PI,1.5*Math.PI)
Start angle	arc(100,75,50,**0**,1.5*Math.PI)
End angle	arc(100,75,50,0*Math.PI,**1.5*Math.PI**)

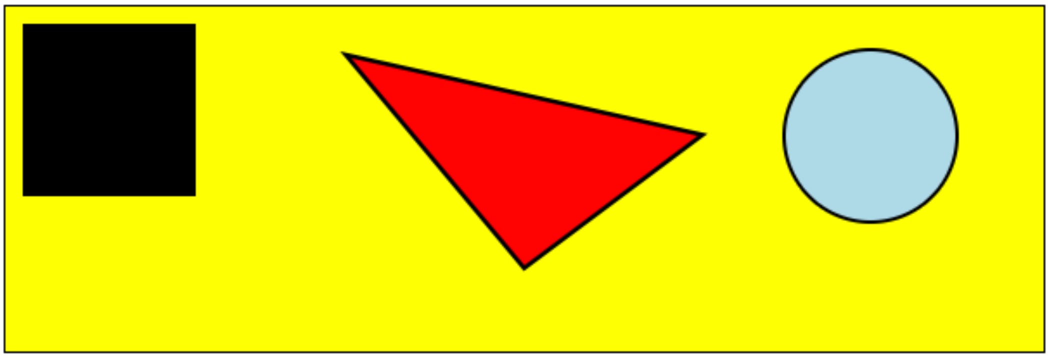
The following script will draw a circle inside the canvas API:

```
<script>
  var c = document.getElementById("testCanvas");
  var ctx = c.getContext("2d");
  ctx.beginPath();
  ctx.arc(200, 150, 50, 0, 2 * Math.PI, true);
  ctx.stroke();
</script>

<body>
  <canvas id="testCanvas" width="300" height="150" style="border:1px"> </canvas>
</body>
```

1. Open a new document in Text Wrangler or Text Edit and save it as **drawing.html**

```
<html>
<head>
 <style>
  canvas {border: 1px solid black; background-color: yellow;}
 </style>

 <script>
  window.onload = function() {
  var canvas = document.getElementById("test");
  var ctx = canvas.getContext("2d");

  // draw a rectangle
  ctx.fillRect(10, 10, 100, 100);

  // draw a triangle
  ctx.beginPath();
  ctx.moveTo(300, 150);
  ctx.lineTo(400, 75);
  ctx.lineTo(200, 30);
  ctx.closePath();
  ctx.lineWidth = 5;
  ctx.stroke();
  ctx.fillStyle = "red";
  ctx.fill();

  // draw a circle
  ctx.beginPath();
  ctx.arc(500, 75, 50, 0, 2 * Math.PI);
  ctx.fillStyle = "lightblue";
  ctx.fill();
  ctx.lineWidth = 2;
  ctx.stroke();
 };
 </script>
</head>
<body>
  <canvas width="600" height="200" id="test"></canvas>
</body>
</html>
```

2. Save the page and preview it in the web browser to check that it looks something like the one shown below:



3. Add the following code inside the script tags to add text to the canvas:

```
// add text
ctx.font = "2em Arial, sans-serif";
```

```
ctx.fillStyle = "black";
ctx.fillText ("Draw a picture of a house", 100, 180);
```

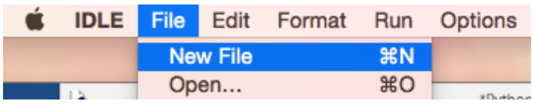
4. Save the page and preview it in the web browser to check that it your text has appeared.

Task: Save this page and then try to draw something by using the commands you have learned.

Advanced task: Include an image (jpg or png)

Appendix J - Using Tkinter to add a Graphical User Interface (GUI)

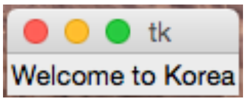
1. Select a **New File** from the **File** menu:



2. Now add the following code to add a label to the root window:

```
from tkinter import * # import the tkinter library
from tkinter import ttk # import the tkinter themed widgets
root = Tk() # set the parent or root
myLabel = ttk.Label(root, text = 'Welcome to Korea')
myLabel.pack()
```

3. Run the program and notice that the program includes the text 'Welcome to Korea' in the label:



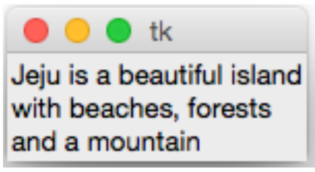
Wrapping Text in a label

You can change the contents of a label with the **config** command.

4. Add the following text underneath the existing program:

```
# change the text to something longer
myLabel.config(text = 'Jeju is a beautiful island with beaches, forests and a mountain')
myLabel.config(wraplength = 150)
```

5. Run the program and notice that the text wraps onto new lines:

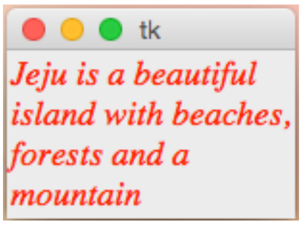


Changing colours, fonts and styles

6. Now enter the following directly under the existing code:

```
myLabel.config(foreground = 'red')
myLabel.config(font = ('Times', 18, 'italic'))
```

7. Run the program and notice that the text is red, 18pt and in italics:



Adding an image

8. Change your program so that it looks like the one below ***(Your gif image must be located in the same folder as your program***):

```
from tkinter import * # import the tkinter library
from tkinter import ttk # import the tkinter themed widgets
root = Tk() # set the parent or root
myLabel = ttk.Label(root, text = 'Welcome to Korea')
myLabel.pack()
# change the text to something longer
logo = PhotoImage(file="Jeju.gif")
myLabel.config(image = logo)
```

9. Run the program and notice that the logo is displayed:

Task: Add some images to your text adventure and put the story to a label.

Advanced Task: Add an Entry box for the user to type in responses (look on the Internet to see how to add an Entry box in Tkinter).

Appendix K: Programming test (HTML5/CSS and Python)

Programming Test Instructions

You have **75 minutes** to complete tasks for three areas:

1. HTML5 / CSS
2. Python turtle
3. Python and flowcharts

Extension activity (attempt only if you have completed all of the three sections and still have time left).

....

You may use the official Python documentation website (https://docs.python.org/3/) and the HTML/CSS section of the W3schools website (http://www.w3schools.com) during the exam.

....

Prior to start of the exam

Step 1: Look at all three sections of the annual test. You may ignore the extension activity until after you have completed the three sections.

Step 2: Complete the online questionnaire sent to your school Gmail account (if you haven't received it then let your teacher know).

Step 3: Create a folder in your Google Drive. Call it **Programming Test**

Exam Starts at this point – you have 75 minutes

Step 4: Complete each task and save it in your Google Drive. You may attempt the tasks in any order. Only attempt the extension task if you have finished the three sections.

Step 5: Share your Programming Test folder with both Computer Science teachers, Mr. McAdams (tmcadams@nlcsjeju.kr) and Mr. Scarlett (nscarlett@nlcsjeju.kr). This can be done after the exam has finished!

Note: Even if your program doesn't work correct send the files anyway because you can still gain marks for the attempt.

HTML / CSS Section [22 marks]

An activity company called Jeju Adventure Centre has hired your company to create a simple website for them. Your manager has delegated the task to you and suggested that you use www.w3schools.com to assist you if you are confused by their requests.

Follow the Instructions carefully

1 Open a new document in Text Wrangler and save the html file using your login name and the extension .html (e.g. **yjkim19.html**). [1]

2 Create a basic html page using the following html (note: replace the comment **yjkim19** with your login name):

```
<!-- yjkim19 --!>
<!doctype html>
<html>
<head>

</head>

<body>

</body>
</html>
```

[2]

3 Within the head tags <head> …. </head> add:

- a **title tag** containing the title: **Jeju Adventure Centre** [2]

(Help on this topic can be found: http://www.w3schools.com/tags/tag_title.asp)

4 Within the body tags add: [1]

- a header tag with a h1 tag containing the text: **Jeju Adventures**

(Help on this topic can be found: http://www.w3schools.com/tags/tag_header.asp)

5 Make sure the text inside the h1 tag is: **centre aligned**, **Impact**, **30pt**. [2]

6 Add a **horizontal rule** above and below the header and ensure that there is a gap

around the text and that your page is similar to the one shown below: *[2]*

7 Add an image of <u>any</u> **Jeju logo** to the right of the Jeju Adventures text and set its **width** to **100px**. (Search for a Jeju logo or use this image: http://upload.wikimedia.org/wikipedia/en/thumb/c/cc/Seal_of_Jeju.svg/1024px-Seal_of_Jeju.svg.png)

8 Change the background colour of the header to a **light grey** colour so that your header resembles the following one (Note: it doesn't matter if you have chosen a different Jeju image or your background colour is different): *[2]*

9 Below this header add an **article tag** and copy and paste the following text into it (Note: text location: http://www.jejunature.com/emain/e_index.html)

Jeju Special Self-Governing Province is an isolated island south-east of the mainland of South Korea. It is located 154km from Mokpo, 304km from Busan, and 255.1km from Tsushima of Japan. To the east it is facing Tsushima and Janggi prefecture of Japan with the south sea and East China Sea in between. To the west, Jeju faces Shanghai, China with the East China Sea in between. To the south with the South China sea and to the north is the mainland of South Korea with the South Sea in between. *[1]* *[2]*

[2]

Jeju Special Self-Governing Province's location is southeast of South Korea. Being placed in the center of Northeast Asia has given it a very important geopolitical location in the past. In 1275 (Empire of Goryeo), the Tamna general headquarters of Won was established here. For more than one century it was the headquarters of the conquering Japanese. In the last Pacific war, many military facilities were established here. During the Korean War, the first army training camp was set up here. This is the historical *[2]* *[1]* evidence of Jeju Special Self-Governing Province's importance in location.

[2]

Note: *An article tag does the same thing as a div tag. It sets a contained area on the webpage. (http://www.w3schools.com/tags/tag_article.asp)*

---

20 Change the background colour of the article tag to a **light grey** colour.

21 Add a second image of a Jeju map and set its height to 140px (Note: You can find this image here: http://english.jeju.go.kr/files/image/imgsub01/no13.gif)

22 Position the image to the right of your text by using float

(Help on this is located here: http://www.w3schools.com/cssref/pr_class_float.asp)

23 Add a grey **footer tag** containing a **hyperlink** to the school's website so that is says sponsored by North London Collegiate School Jeju with the school name linking to the school site (www.nlcsjeju.co.kr).

24 Add a **horizontal rule** below your footer.

25 Using CSS, add a 50px left and right margin for the header, article and footer.

(Help on this topic can be found: http://www.w3schools.com/css/css_margin.asp)

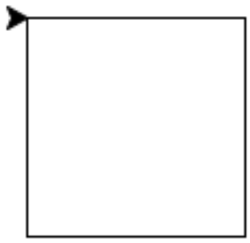
Your final page should resemble the one shown below (Note: if it doesn't go back and check your work)

End of HTML / CSS Section of the Programming Test

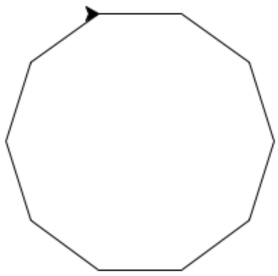
Python Section [12 marks Turtle, 5 marks Flowchart]

The following program draws a square using Python turtle:

```
import turtle
t = turtle.Turtle()

for i in range(4):
    t.forward(100)
    t.right(90)
```



1. Create the following shape (length = 50, sides = 10) using Python turtle:

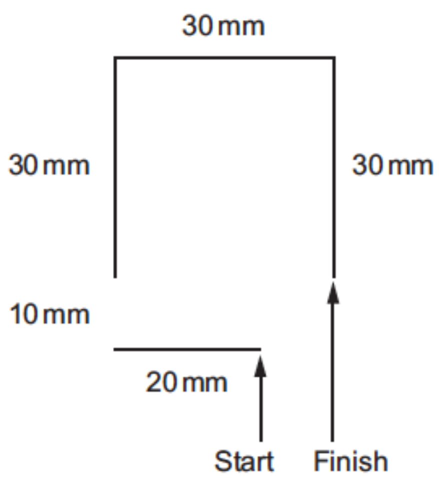


[2]

2. Create the following shape (line thickness is 5, line length is 40)

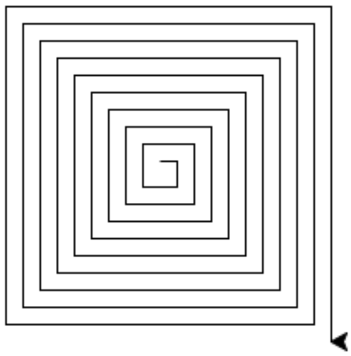
[2]

3. Create the following shape (Note: don't write in text e.g. 30 mm)



[4]

4. Create the following shape (the shortest line is 10, the longest is 195, the length of the line increases by 5 each time, and there are 38 lines in total):



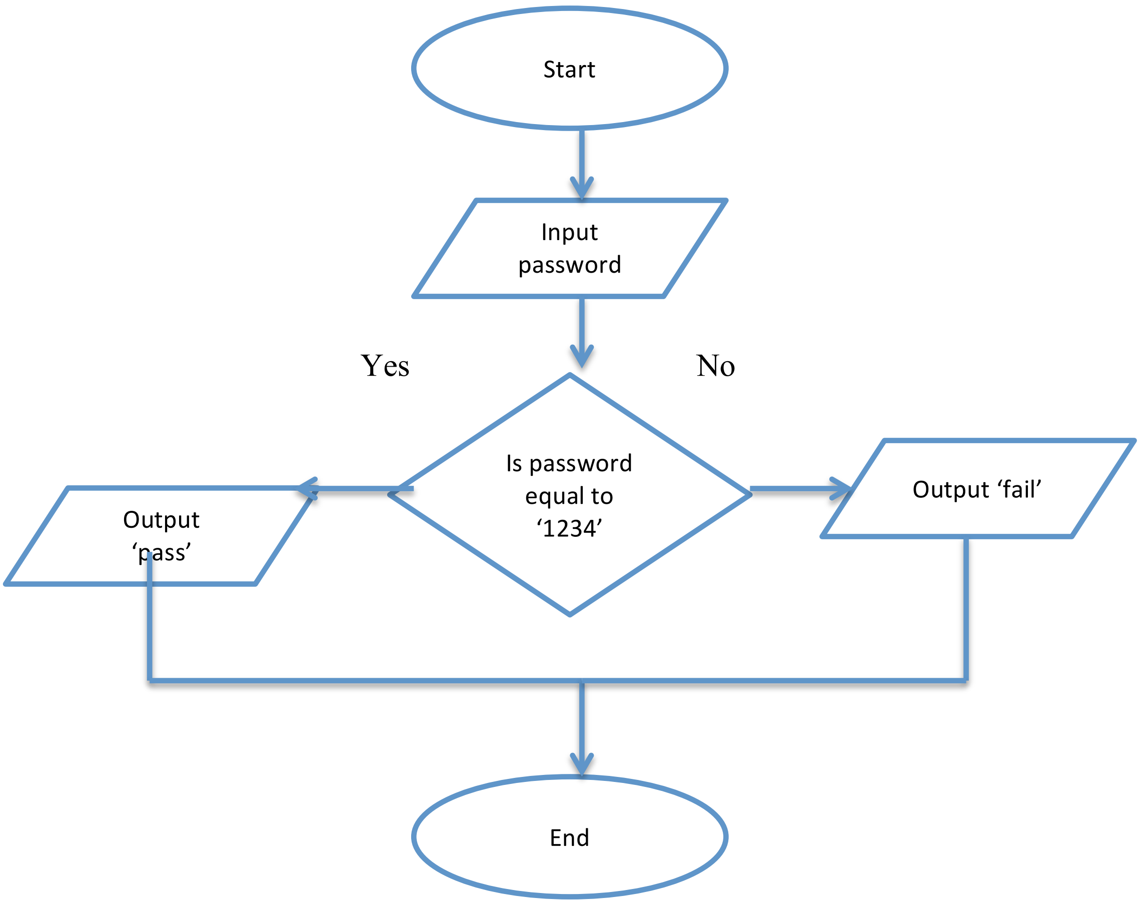
Note: 2 of the 4 marks available will only be awarded if a loop is used. [4]

Python programming and Flowchart

Question 1 – [2 marks]

1. Open Python Idle and create a new file. Save it as **Q1.py**

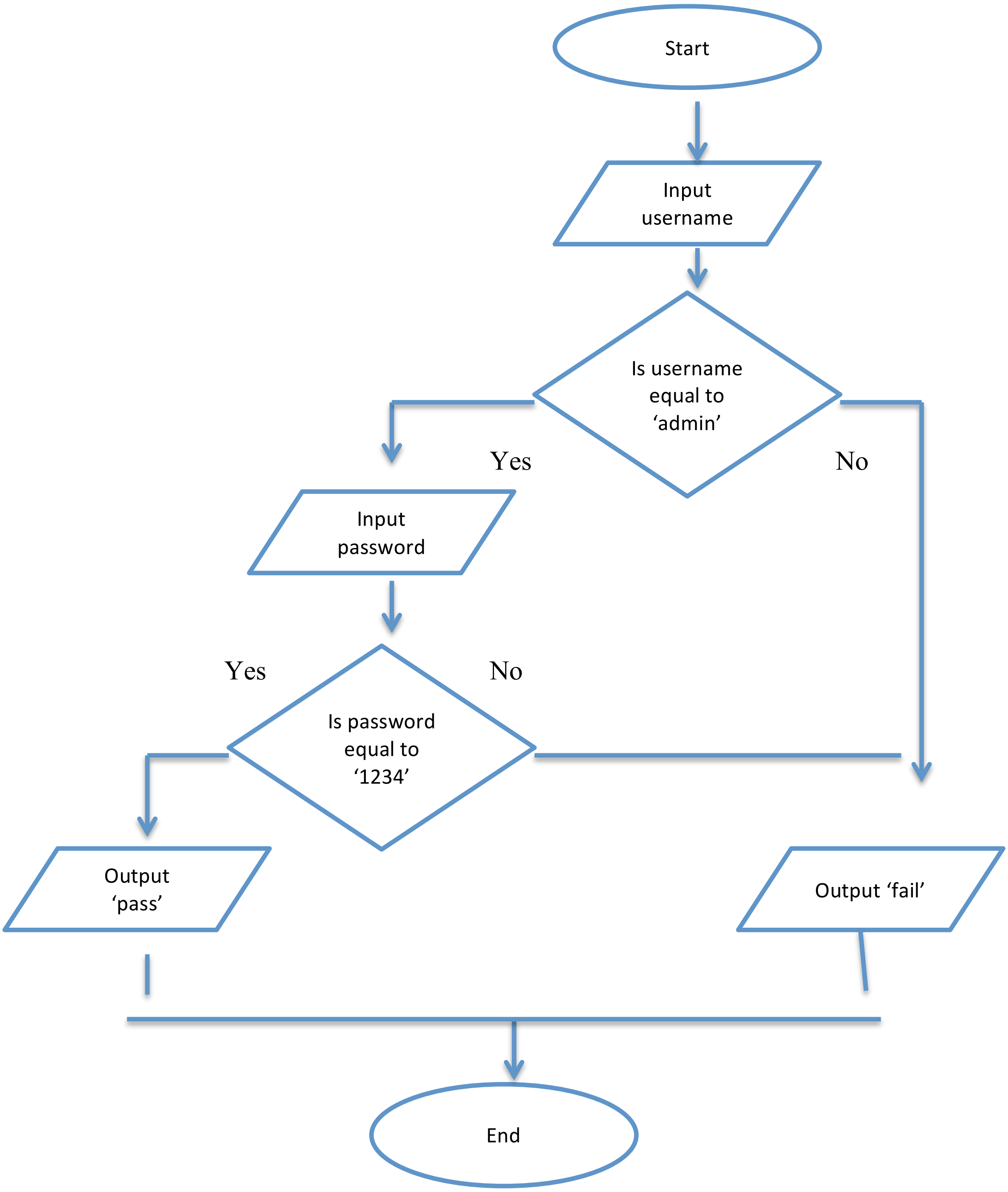
2. Write a program that asks the user to enter a password and then check to see if the password is '1234'. If it is output 'Pass' otherwise output 'Fail'. The flowchart below demonstrates the program:



Question 2 - [1 mark]

1. Create a new file in Python Idle and save it as **Q2.py**

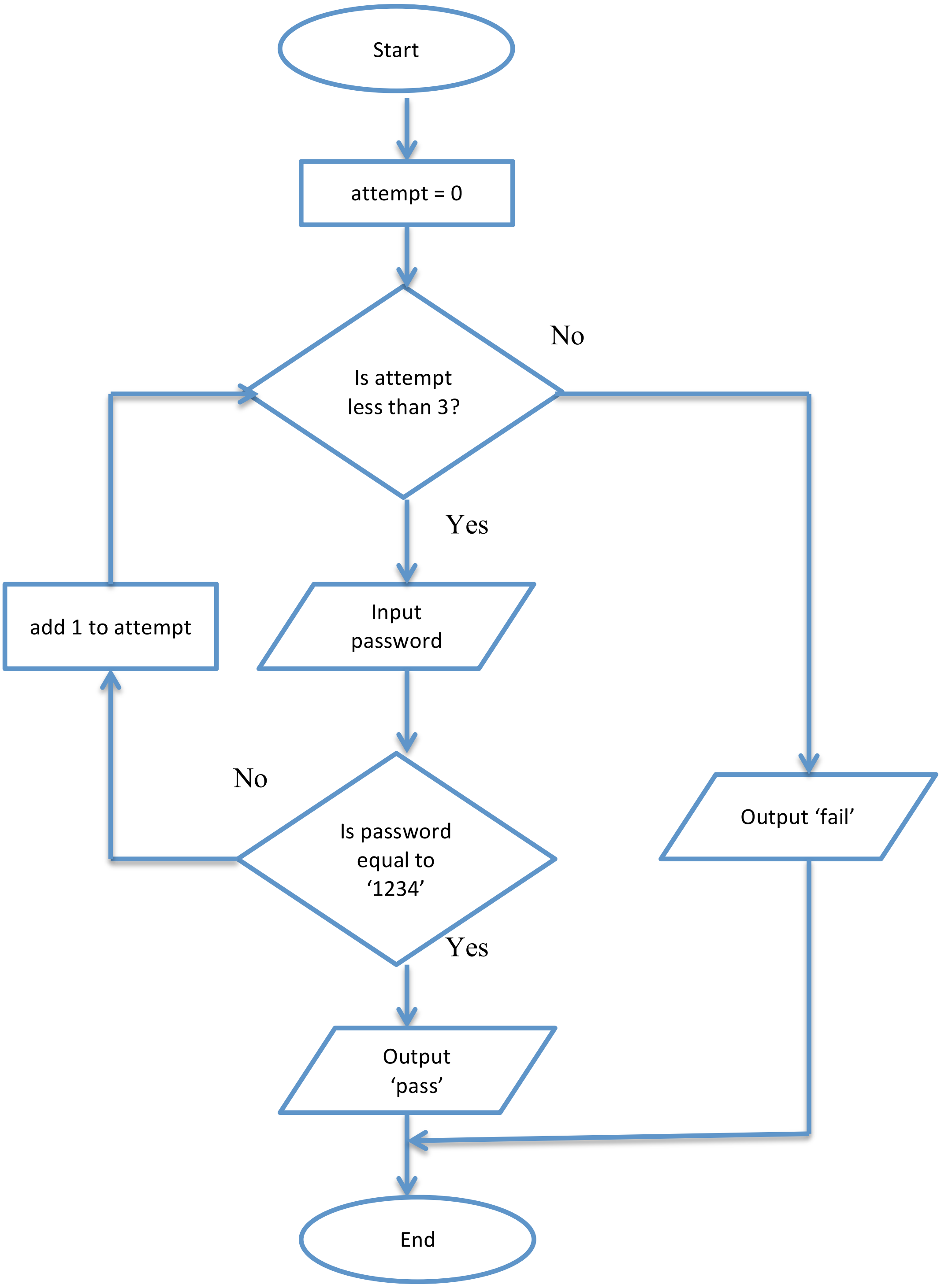
2. Write a program that meets the requirement of the flowchart:



Question 3 - [2 marks]

1. Create a new file in Python idle and save it as **Q3.py**

2. Write a program that meets the requirement of the flowchart and when complete save it and **email** it to your teacher:



Extension Question: This is for extra credit only

Only attempt if you complete the first 3 sections within 75 minutes.

"Write a program that repeatedly asks the user to input integers, until the user enters the integer 99999. After seeing 99999 it should print out the correct average. That is, should not count the final 99999."

For example, if the following 4 integers were entered the output would be 6.

5, 6, 7, 99999

Average: 6

Programming test solution

Solution to HTML / CSS

The Excel sheet contains the marks for each question.

Solution to Python Turtle

Solution to Flowchart

Rationale

Marks may still be awarded if the logic is correct but the program doesn't run. However, if the logic is incorrect or text simply copied from the flowchart, do not award a mark.

1.

```
password = input("Enter your password: ")
if password == "1234":
    print('pass')
else:
    print('fail')
```

[2 marks]

1 mark for each point

- input password
- if statement

Examples of where the mark can and can not be awarded:

Forgotten or incorrect speech marks, or uppercase instead of lower case, is still correct:

password = Input(Enter Password)

Placing the variable and input in an incorrect order is not a mark – too similar to question:

Input password

2. **[1 marks]**

1 mark for each point

- if statement inside the if statement (nested if statement)

3. [**2 marks]**

1 mark for each point

- attempts increases by one if password is wrong
- while loop checks <3

Solution to the ‘Rainfall Problem’ extension question

Appendix L: Computer Programming Confidence and Anxiety (CPCA) Questionnaire

What year group are you in?

- ○ 8
- ○ 9

What is your gender?

- ○ Female
- ○ Male

1. The challenge of learning computer programming is exciting?

Choose the appropriate answer below

- ○ 1 - strongly agree
- ○ 2 - agree
- ○ 3 - somewhat agree
- ○ 4 - neither agree nor disagree
- ○ 5 - somewhat disagree
- ○ 6 - disagree
- ○ 7 - strongly disagree

2. I am confident that I can learn to write programs to solve the kinds of problems we encounter in Computer Science classes.

Choose the appropriate answer below

- ○ 1 - strongly agree
- ○ 2 - agree
- ○ 3 - somewhat agree
- ○ 4 - neither agree nor disagree
- ○ 5 - somewhat disagree
- ○ 6 - disagree
- ○ 7 - strongly disagree

3. I get frustrated/fed up when the computer program I have written doesn't work.

Choose the appropriate answer below

- ○ 1 - strongly agree
- ○ 2 - agree
- ○ 3 - somewhat agree
- ○ 4 - neither agree nor disagree
- ○ 5 - somewhat disagree
- ○ 6 - disagree
- ○ 7 - strongly disagree

4. I am confident that I can find and correct errors in my computer programs.

Choose the appropriate answer below

- ◯ 1 - strongly agree
- ◯ 2 - agree
- ◯ 3 - somewhat agree
- ◯ 4 - neither agree nor disagree
- ◯ 5 - somewhat disagree
- ◯ 6 - disagree
- ◯ 7 - strongly disagree

5. I don't like to make changes to a working computer program in case it goes wrong and stops working.

Choose the appropriate answer below

- ◯ 1 - strongly agree
- ◯ 2 - agree
- ◯ 3 - somewhat agree
- ◯ 4 - neither agree nor disagree
- ◯ 5 - somewhat disagree
- ◯ 6 - disagree
- ◯ 7 - strongly disagree

6. I am confident that when I have a programming task, I can work independently by using the Internet to help me find a solution.

Choose the appropriate answer below

- ◯ 1 - strongly agree
- ◯ 2 - agree
- ◯ 3 - somewhat agree
- ◯ 4 - neither agree nor disagree
- ◯ 5 - somewhat disagree
- ◯ 6 - disagree
- ◯ 7 - strongly disagree

7. I think that, in the future, I would have the capability to successfully complete a University degree that includes modules in computer programming.

Note: This is not asking if you want to study programming at University, only if you think you have the capability

- ◯ 1 - strongly agree
- ◯ 2 - agree
- ◯ 3 - somewhat agree
- ◯ 4 - neither agree nor disagree
- ◯ 5 - somewhat disagree
- ◯ 6 - disagree
- ◯ 7 - strongly disagree

8. I will be able to correctly complete the HTML / CSS task during the exam?

Look at the HTML / CSS question and choose the appropriate answer below

- 1 - strongly agree
- 2 - agree
- 3 - somewhat agree
- 4 - neither agree nor disagree
- 5 - somewhat disagree
- 6 - disagree
- 7 - strongly disagree

9. I will be able to correctly complete the python turtle tasks (not including the extension question) during the exam?

Look at the python turtle questions and choose the appropriate answer below

- 1 - strongly agree
- 2 - agree
- 3 - somewhat agree
- 4 - neither agree nor disagree
- 5 - somewhat disagree
- 6 - disagree
- 7 - strongly disagree

10. I will be able to correctly complete the python flowchart tasks (not including the extension question) during the exam?

Look at the python flowchart questions and choose the appropriate answer below

- 1 - strongly agree
- 2 - agree
- 3 - somewhat agree
- 4 - neither agree nor disagree
- 5 - somewhat disagree
- 6 - disagree
- 7 - strongly disagree

11. On average, do you think that there is any difference in academic abilities between boys and girls in your year group?

Please answer honestly from the options below

- 1 - Boys are much more academic than Girls
- 2 - Boys are more academic that Girls
- 3 - Boys are slightly more academic than Girls
- 4 - No difference between Boys and Girls
- 5 - Girls are slightly more academic than Boys
- 6 - Girls are more academic that Boys
- 7 - Girls are much more academic than Boys

12. On average, do you think that there will be any difference, between boys and girls in your year group, in the results of this test?

Please answer honestly from the options below

- ◯ 1 - Boys will scores much higher than Girls
- ◯ 2 - Boys will score higher than Girls
- ◯ 3 - Boys will score slightly higher than Girls
- ◯ 4 - No difference between Boys and Girls
- ◯ 5 - Girls will score slightly higher than Boys
- ◯ 6 - Girls will score higher than Boys
- ◯ 7 - Girls will scores much higher than Boys

Appendix M: Student Perceptions of Computer Science (SPCS) Questionnaire

What is your gender?
Choose Female or Male

- ☐ Female
- ☐ Male

What year group are you in?
Choose 8 or 9

- ◯ Year 8
- ◯ Year 9

1. On average, and compared with your other classes, how difficult / easy is computer science?

- ◯ 1 - much more difficult
- ◯ 2 - more difficult
- ◯ 3 - a little more difficult
- ◯ 4 - about the same
- ◯ 5 - a little easier
- ◯ 6 - easier
- ◯ 7 - much easier

2. On average, and compared with your other classes, how enjoyable is computer science?

- ◯ 1 - much more enjoyable
- ◯ 2 - more enjoyable
- ◯ 3 - a little more enjoyable
- ◯ 4 - about the same
- ◯ 5 - a little less enjoyable
- ◯ 6 - less enjoyable
- ◯ 7 - much less enjoyable

3. Do you think boys or girls are more suited to a career in computer programming?

- ◯ 1 - boys are much more suited
- ◯ 2 - boys are more suited
- ◯ 3 - boys are a little more suited
- ◯ 4 - no gender difference
- ◯ 5 - girls are a little more suited
- ◯ 6 - girls are more suited
- ◯ 7 - girls are much more suited

4. Describe the abilities or skills that you think are most useful for computer programming.

5. Do you think learning computer programming is an important activity to do or not?

Please explain why you do or don't

6. Do you think that you will choose to study computer programming in the future?

This might be IGCSE Computer Science, IB Computer Science, a module at University, or self-study!

- ◯ 1 - Really want to program in the future
- ◯ 2 - Want to program in the future
- ◯ 3 - Somewhat want to program in the future
- ◯ 4 - No idea at this time
- ◯ 5 - Somewhat don't want to program in the future.
- ◯ 6 - Don't want to program in the future
- ◯ 7 - Really don't want to program in the future

7. Do you know what career / job that you would like to pursue and will this influence your choices at IGCSE or IB?

8. How much do you enjoy playing computer games?

- ◯ 1 - I love computer games
- ◯ 2 - I like computer games
- ◯ 3 - I somewhat like computer games
- ◯ 4 - I neither like or dislike computer games
- ◯ 5 - I somewhat dislike computer games
- ◯ 6 - I dislike computer games
- ◯ 7 - I hate computer games

9. In general , do you think computer games are a useful activity or a distraction from doing useful things?

- ◯ 1 - Really useful activity
- ◯ 2 - Useful activity
- ◯ 3 - Slightly useful activity
- ◯ 4 - Neither useful nor distracting
- ◯ 5 - Slightly distracting from doing useful activities
- ◯ 6 - Distracting from doing useful activities
- ◯ 7 - Really distracting from doing useful activities

10. What type of computer games do you play the most?

Choose the one type of game you play the most often.

- ◯ Online mulitplayer
- ◯ Simulation
- ◯ Adventure
- ◯ Real time strategy
- ◯ Puzzle
- ◯ First person shooter
- ◯ Stealth shooter
- ◯ Sports
- ◯ Role playing games (RPG)
- ◯ Educational

11. On average, how much time per week do you spend playing computer games?

- ◯ 0 - 20 minutes per week
- ◯ 21 - 40 minutes per week
- ◯ 41 - 60 minutes per week
- ◯ 61 - 80 minutes per week
- ◯ 81 - 100 minutes per week
- ◯ 101 - 120 minutes per week
- ◯ 121 - 140 minutes per week
- ◯ 141 - 160 minutes per week
- ◯ 161 - 180 minutes per week
- ◯ More than three hours per week

Submit

Appendix N: Influences on IGCSE Choices (IIC) Questionnaire

Influences on IGCSE Choices

Anonymous Research into what influences students IGCSE choices.

What is your current year group?

- ○ Year 9
- ○ Year 10

What is your gender?

- ○ Male
- ○ Female

1. Which subjects do you think you will choose (Year 9) or did you choose (Year 10) for IGCSE?

- ☐ Art
- ☐ Computer Science
- ☐ Dance
- ☐ Drama
- ☐ France
- ☐ Geography
- ☐ History
- ☐ Latin
- ☐ Mandarin
- ☐ Music
- ☐ Spanish

2. How much did your parent(s) influence your decision? (Scale 1 - 7 in order of influence)

- 1. Exceptionally influential
- 2. Very influential
- 3. Reasonably influential
- 4. Influential
- 5. Slightly influential
- 6. Very slightly influential
- 7. No influence at all

3. How much did your teacher(s) influence your decision? (Scale 1 - 7 in order of influence)

- 1. Exceptionally influential
- 2. Very influential
- 3. Reasonably influential
- 4. Influential
- 5. Slightly influential
- 6. Very slightly influential
- 7. No influence at all

4. How much did your friend(s) influence your decision? (Scale 1 - 7 in order of influence)

- 1. Exceptionally influential
- 2. Very influential
- 3. Reasonably influential
- 4. Influential
- 5. Slightly influential
- 6. Very slightly influential
- 7. No influence at all

5. How much did useful life skills to be learned influenced your decision? (Scale 1 - 7 in order of influence)

- ◯ 1. Exceptionally influential
- ◯ 2. Very influential
- ◯ 3. Reasonably influential
- ◯ 4. Influential
- ◯ 5. Slightly influential
- ◯ 6. Very slightly influential
- ◯ 7. No influence at all

6. How much did your university plans or career plans influence your decision? (Scale 1 - 7 in order of influence)

- ◯ 1. Exceptionally influential
- ◯ 2. Very influential
- ◯ 3. Reasonably influential
- ◯ 4. Influential
- ◯ 5. Slightly influential
- ◯ 6. Very slightly influential
- ◯ 7. No influence at all

7. How much has lesson enjoyment influenced your decision? (Scale 1 - 7 in order of influence)

- ◯ 1. Exceptionally influential
- ◯ 2. Very influential
- ◯ 3. Reasonably influential
- ◯ 4. Influential
- ◯ 5. Slightly influential
- ◯ 6. Very slightly influential
- ◯ 7. No influence at all

8. How much has subject difficulty influenced your decision? (Scale 1 - 7 in order of influence)

- ○ 1. Exceptionally influential
- ○ 2. Very influential
- ○ 3. Reasonably influential
- ○ 4. Influential
- ○ 5. Slightly influential
- ○ 6. Very slightly influential
- ○ 7. No influence at all

9. How much has a role model (not a parent or teacher) inspired you and influenced your decision? (Scale 1 - 7 in order of influence)

- ○ 1. Exceptionally influential
- ○ 2. Very influential
- ○ 3. Reasonably influential
- ○ 4. Influential
- ○ 5. Slightly influential
- ○ 6. Very slightly influential
- ○ 7. No influence at all

10. If you were influenced by a role model who was it or were they?

Long answer text

11. Were there any other influencing factors and if so what were they?

Long answer text

Appendix O: Preferred Computer Programming Learning Style (PCPLS) Questionnaire

What is your gender?
Choose Female or Male

- ☐ Female
- ☐ Male

What year group are you in?
Choose 8 or 9

- ◯ Year 8
- ◯ Year 9

1. I really enjoy learning to program in python by writing short programs which solve small problems
1 – 7 (1 – strongly agree, 7 strongly disagree)

- ◯ 1 - Strongly agree
- ◯ 2 - Agree
- ◯ 3 - Somewhat agree
- ◯ 4 - Neither agree nor disagree
- ◯ 5 - Somewhat disagree
- ◯ 6 - Disagree
- ◯ 7 - Strongly disagree

2. I find completing short problems in code difficult and often need help!
1 – 7 (1 – strongly agree, 7 strongly disagree)

- ◯ 1 - Strongly agree
- ◯ 2 - Agree
- ◯ 3 - Somewhat agree
- ◯ 4 - Neither agree nor disagree
- ◯ 5 - Somewhat disagree
- ◯ 6 - Disagree
- ◯ 7 - Strongly disagree

3. When solving a problem it really helps if I can see the program as a flowchart.
1 – 7 (1 – strongly agree, 7 strongly disagree)

- ◯ 1 - Strongly agree
- ◯ 2 - Agree
- ◯ 3 - Somewhat agree
- ◯ 4 - Neither agree nor disagree
- ◯ 5 - Somewhat disagree
- ◯ 6 - Disagree
- ◯ 7 - Strongly disagree

4. I find completing short problems develops my programming skills much more than working on a project.
1 – 7 (1 – strongly agree, 7 strongly disagree)

- 1 - Strongly agree
- 2 - Agree
- 3 - Somewhat agree
- 4 - Neither agree nor disagree
- 5 - Somewhat disagree
- 6 - Disagree
- 7 - Strongly disagree

5. When learning programming concepts, (e.g. loops, subroutines) in python, do you find it more interesting / enjoyable to learn it graphically (i.e turtle or tkinter) rather than text-based?
1 – 7 (1 – strongly agree, 7 strongly disagree)

- 1 - Strongly agree
- 2 - Agree
- 3 - Somewhat agree
- 4 - Neither agree nor disagree
- 5 - Somewhat disagree
- 6 - Disagree
- 7 - Strongly disagree

6. When learning programming, I really enjoy working independently on my own.
1 – 7 (1 – strongly agree, 7 strongly disagree)

- 1 - Strongly agree
- 2 - Agree
- 3 - Somewhat agree
- 4 - Neither agree nor disagree
- 5 - Somewhat disagree
- 6 - Disagree
- 7 - Strongly disagree

7. When learning programming, I really enjoy working collaboratively, where a friend and I, write the program together.
1 – 7 (1 – strongly agree, 7 strongly disagree)

- 1 - Strongly agree
- 2 - Agree
- 3 - Somewhat agree
- 4 - Neither agree nor disagree
- 5 - Somewhat disagree
- 6 - Disagree
- 7 - Strongly disagree

8. When learning programming, I really enjoy working cooperatively, where I work on one task and my friend works on another, before we combine them to complete the project.
1 – 7 (1 – strongly agree, 7 strongly disagree)

- 1 - Strongly agree
- 2 - Agree
- 3 - Somewhat agree
- 4 - Neither agree nor disagree
- 5 - Somewhat disagree
- 6 - Disagree
- 7 - Strongly disagree

9. I find working on a project improves my programming skills much more than completing short exercises.
1 – 7 (1 – strongly agree, 7 strongly disagree)

- 1 - Strongly agree
- 2 - Agree
- 3 - Somewhat agree
- 4 - Neither agree nor disagree
- 5 - Somewhat disagree
- 6 - Disagree
- 7 - Strongly disagree

10. When learning to program I find having a storyline (e.g. Text adventure, Sherlock) really helps my learning?
1 – 7 (1 – strongly agree, 7 strongly disagree)

- 1 - Strongly agree
- 2 - Agree
- 3 - Somewhat agree
- 4 - Neither agree nor disagree
- 5 - Somewhat disagree
- 6 - Disagree
- 7 - Strongly disagree

11. Look at the following list of different ways we have learned computer programming, and explain which was the most effective method and why it was effective?
(1. Group work 2. Independent work 3. Self-instructional tutorials 4. Storytelling 5. Project / Problem-based 6. Visual Design (turtle, tkinter)

12. Are there any other ways of learning that are used in other subjects, which might improve the way computing programming is taught?
i.e. How can the teaching of computer programming be improved?

Appendix P: Semi-structured questions for the group interviews

Start with asking what options they picked at IGCSE and the reasons why.

Chose IGCSE Computer Science

Q1. What were the main reasons that you choose IGCSE Computer Science?

Did not choose IGCSE Computer Science

Q1. What were the main reasons that you didn't choose IGCSE Computer Science?

Q2. Do your parents have much influence over your IGCSE choices? Are they any other influencing factors?

Q3 Are you more likely to take a subject like Computer Science if the curriculum was written to incorporate other subjects you enjoy?

Q4. Do you think that the decisions you make in Year 9 impact upon your future careers?

Q5. Do you think Year 9 is too soon to make decisions that might affect your future career?

Q6. Why do you think that more boys take Computer Science than girls?

Q7. Why do you think that Computer Science at university and industry is male-dominated?

Q8. Do you think playing computer games impacts upon the interesting in learning about Computer Science?

Q9. Boys' favourite type of game was online multiplayer! What is it about this genre that is so appealing?

Q10. Girls' favourite type of game was a tie between puzzle and simulation. What is appealing about these genres?

Q11. Do you think that an interest in computer games has any influence over whether you choose Computer Science or not?

Q12. How much do you enjoy learning when playing a Quiz game on Kahoot?
Likert 1 - 7

Q13. Do role models matter to you?
Likert 1 – 7

Q14. Who are you role models and their profession?

Appendix Q: Lichtman's Stage 1 - Initial comments

Janice and Bella (Female non-IGCSE Computer Science students)

Interviewer: So when you took your option choices what did you choose?

Janice: Art, History and Mandarin

Interviewer: Ok, so why did you choose Art, History and Mandarin?

Janice: First of all, I chose History because I have to at least choose one of the Humanities subjects. So I prefer History than Geography and I thought History is very interesting subject so I chose History and for Mandarin, it is very important to speak another language instead of English and Korean, which is my mother tongue. And China is like increasingly getting more developed and so parents and I thought speaking Chinese will help my future career and help me to be more successful in the future so I chose Mandarin. And for Art, I had thought I had to balance my life and so if I only focus on academic stuff I won't be able to um manage a lot of stuffs and so my brain might just explode (small laugh). I like Art and I like Art History, it helps to calm down myself and so I chose Art.

Interviewer: You said that you think Mandarin will be useful career-wise in the future. What do you see yourself doing?

Janice: Cause um as you know China is the biggest population. That means that they the people the economy is very productive so their companies and their economics are getting bigger and they are becoming more modernising industrialised and you know Korea is near to China and so if I work in Korea and work global if I work in Korea there will be a lot of stuffs that I have to cooperate with Chinese companies and a lot of economies predict that China will be develop as much as US economy in the future. So, yeah! A lot as I said before, China is the biggest economy in the world and so a lot that means that means most of world population speaks China so if I speak Chinese I will be able to communicate with a lot of most people.

Interviewer: So what job do you see yourself doing?

Appendix R: Group interview narrative and emerging themes

Janice and Bella (Female non-IGCSE Computer Science students)

Transcript	Theme
Interviewer: So when you took your option choices what did you choose?	
Janice: Art, History and Mandarin	Creative / language
Interviewer: Ok, so why did you choose Art, History and Mandarin?	
Janice: First of all, I chose History because I have to at least choose one of the Humanities subjects. So I prefer History than Geography and I thought History is very interesting subject so I chose History and for Mandarin, it is very important to speak another language instead of English and Korean, which is my mother tongue. And China is like increasingly getting more developed and so parents and I thought speaking Chinese will help my future career and help me to be more successful in the future so I chose Mandarin. And for Art, I had thought I had to balance my life and so if I only focus on academic stuff I won't be able to um manage a lot of stuffs and so my brain might just explode (small laugh). I like Art and I like Art History, it helps to calm down myself and so I chose Art.	Required Enjoy Interest Language Parents Future / career / Success / creative Balance / academics Cope Enjoy Relax
Interviewer: You said that you think Mandarin will be useful career-wise in the future. What do you see yourself doing?	
Janice: Cause um as you know China is the biggest population. That means that they the people the economy is very productive so their companies and their economics are getting bigger and they are becoming more modernising industrialised and you know Korea is near to China and so if I work in Korea and work global if I work in Korea there will be a lot of stuffs that I have to cooperate with	

Chinese companies and a lot of economies predict that China will be develop as much as US economy in the future. So, yeah! A lot as I said before, China is the biggest economy in the world and so a lot that means that means most of world population speaks China so if I speak Chinese I will be able to communicate with a lot of most people.	Economy Career Global Future
Interviewer: So what job do you see yourself doing?	
Janice: I want to work in company but I didn't decide what company what company should I work.	Economy Language Communicate
Interviewer: So how much influence does your parents have in your IGCSE choices?	Career
Janice: Hmmmm. My parents support my decision. They really did not influence my options choice because they support my decision, but because I was struggling between Art and Geography and I asked my mum whether I should take Art or Geography and she said that its she wants me to do art, in her opinion, but she didn't force me to do art. She showed her opinion to me.	Parents / decision
Interviewer: So, do you think these choices at this age group influence your career or university choices? For example, if you didn't do Drama, or say, Computer Science, at IGCSE do you think you are likely to take Drama, or Computer Science, at IB level or later in life. How much do you think the decision making the decision you made at this age will affect whether you go back to these subjects?	Support decision Parents
Janice: So, do you mean how much influence does the options choice have on my future? Because, the IB choices made on IGCSE options choice, that that means if I don't I didn't take for options for example for Computer Science I can't take in IB therefore I can't take in	

University, I can't major, it is hard to major in university. That means it is hard to get a job in that subject so I think IGCSE options choice was important start my life. Yeah!	
Interviewer: You can choose subjects at IB that you didn't study at IGCSE. For example, you can choose Economics at IB and you don't study Economics at IGCSE. So, you could potentially choose Computer Science at IB even if you didn't study it at IGCSE.	Future Can't
Janice: But there will be a lot of Computer Science, IGCSE Computer Science students who will took IB Computer Science for their choice and I if I take Computer Science for IB I will it will be hard to follow their skills so they will be much more skilful in Computer Science than me and I heard that I have to start with beginner level while other can be standard level and higher level.	Career Important
Interviewer: Why do you think so many more boys than girls choose to study Computer Science?	Competition
Janice: Umm	
Interviewer: Either of you can answer this it doesn't have to be just Janice.	Difficulty Beginner
Janice: Cause um	
Bella: Um	
Interviewer: If you don't know it is fine.	
Janice: Ah, we it's a stereotype but we tend to think that boys have more developed left-brain.	

Bella: Yes, and I think also mostly boys watch robotic animation, like cartoon. Which is about technology. Um, since girls watch girlish thing, and pretty thing kind of related to art. I think boys as they grow up tend to interested in computing and other technology and girls are more interested in art or drama.	
Interviewer: So Bella, you think that boys are more interested in learning computer programming?	Gender difference
Bella: Yes!	
Interviewer: and you Janice, what did you mean by boys have more developed left-brain?	Gender difference
Janice: Well err, just that left brain people tend to be better at Math and Computer Science and stuff like that, but I don't know.	Creative Interest / Creative
Interviewer: So, Bella what were your option choices?	
Bella: I chose History, Geography and Drama.	
Interviewer: So, why did you choose those subjects?	Interest
Bella: Err, even though I knew doing two humanities is really difficult, I am interested in History but also Geography. I don't want to gave up, give up Geography or History so I chose both.	
Interviewer: And the other subject?	Gender difference
Bella: Drama?	Genetic / maths
Interviewer: Yeah	Gender difference

Bella: Um, from I was young I was interested in drama and I was always watch drama and study how to act like professionally. So I always went to musical thing and theatre to um study drama separately and then I saw that in this school there is subject drama and so yes I chose drama when I became year 10. **Interviewer:** So, do you think your parents had influence on your decisions or not? **Bella:** Err, like her, like her, my father and mother support my, um support my, choices and also I think I need to balance my academic thing cause I am doing History and Geography which is very academic thing. And um yes my mother also said I think you are talented in Drama since she saw me from when I was young. **Interviewer:** Do you consider Computer Science to an academic subject? **Bella**: Um **Interviewer**: Well, do you think it is a difficult subject? **Bella:** I think Computer Science um, in the beginning it's quite easy but later on is quite difficult but I really wanted to do Computer Science now. **Interviewer**: Why? **Bella**: Cause, um since the world, since the world and company kind of develop and want to develop their technology so they tend to find more people who did major of Computer Science and also when I am doing IGCSE Computer Science it could be better for my future career.	Difficulty Interest Creative Interest Support / Parents Balance Parents Ability

Interviewer: Do you see it an issue that you are female?	
	Difficulty
Bella: No	
Interviewer: So you think that girls are just as good as boys in Computer Science?	
Bella: Yes	Global
Interviewer: Janice?	
	Future / Career
Janice: Yes, of course but girls are also interested in other stuff.	
Interviewer: So, how much influence do your teachers have on your option choices?	Interest
Janice: For me, teachers did not have a lot of influence but not teachers but the grades the teachers have given me before have influenced me because The grades prove that, no the grades doesn't prove a lot of things but grades still tell me that I am good at this subject so I consider what grades I've got for a subject.	Teachers Grades
	Ability
Interviewer: What about you Bella?	
Bella: I completely agree with Janice.	
Interviewer: So you weren't influenced by whom your teachers were?	
Bella and Janice: No	
Interviewer: Did any of your teachers try to persuade you to choose	

their subject? **Janice**: Some teachers persuaded but I knew why those teachers were persuading **Interviewer**: Did your friends influence your IGCSE choices in any way? **Bella**: No **Janice**: No **Interviewer**: Okay, thank you for your time.	Teachers Friends

Eunice and Yuna (Female non-IGCSE Computer Science students)

Transcript	Theme
Interviewer: So when you took your option choices what did you choose?	
Eunice: I chose Art, and both the Humanities Geography and History	Creative
Interviewer: Why did you choose those subjects?	
Eunice: First of all I wanted to do a creative subject so I was wondering to and I quite like art so I chose art, and I quite like Mandarin but I kind of hated it so I just chose the humanities.	Creative Enjoyment

Interviewer: You wanted to do a creative subject? **Eunice:** Yes **Interviewer:** Why? **Eunice:** To kind of like not only like just do studies but kind of relaxing and doing something fun.	Balance
Interviewer: and are you happy with your choices? **Eunice:** Yes **Interviewer:** Okay, so what do you want to do at university or job-wise? **Eunice:** Um, I want to do something related to Biology.	Future
Interviewer: So were you choices at IGCSE related to this or not? **Eunice**: Umm, not really sure but there was um beacon of um. Actually, I have no idea. **Interviewer**: That's ok! You can have no idea. How about if you don't choose a subject, say, Computer Science, are you likely to choose it at IB or study it later in life? **Eunice**: Um, I don't think so because I will be really behind the other students if I haven't chosen that subject compared to what other students have gone through in the Computer Science course.	Future
Interviewer: Thank you Eunice! Yuna, what were your choices?	

Yuna: Um, Geography, Mandarin and Dance. **Interviewer**: So why did you choose those? **Yuna:** I enjoy Mandarin and I was really like I really thought about choosing History or Geography but personally I hate History so I chose Geography. I thought it will be will make my life much easier if I choose something creative things, so I chose Dance. **Interviewer**: Okay! So, this question is for both of you. How much influence did your parents have on your IGCSE choices? **Yuna**: None *(and very quickly afterwards, almost in unison)* **Eunice**: None (both girls laugh) **Interviewer:** What about generically; Within the school, do parents influence IGCSE choices? **Yuna**: Some *(and at the same time)* **Eunice**: Some parents **Interviewer:** Okay, so obviously you didn't choose Computer Science, but 19 girls did and 58 boys also chose Computer Science *(both girls laughed).* Now that trend is true at University, in fact more so, with only 20% of the Computer Science students in America being female. And in Industry the ratio is similar. So why do you think more boys, males, go into Computer Science than females? *(Long pause as the girls appear to be thinking)* **Yuna**: I heard men's brain are much developed in Maths or Science thing than girls and that influence them, I think.	Creative Enjoyment Creative Parents Parents

Interviewer: What about you Eunice? **Eunice**: Ummm, I am no sure but just in my opinion, I am just not interested into computers and stuff. I think that boys are more interested in playing computer games and living with their phones than girls I think. Maybe? **Interviewer**: Yuna, you said that you heard that boys have the right time of brain for Maths or Computers! Where do you think that you heard that? **Yuna**: Umm, I saw it in Newspaper, as kind of a research paper comparing boys and girls, physically or mentally, yeah. **Interviewer**: Thank you so much for your time. Your answers have been really useful and will be kept confidential.	Gender difference / genetic Interest Interest Gender difference

Kay, Sarah, and Gina (Female IGCSE Computer Science students)

Transcript	Theme
Interviewer: Okay we'll start with Sarah. What were your IGCSE options? **Sarah:** Um. My IGCSE options were Computer Science, History, Mandarin and Korean *(Korean isn't really an option since it is compulsory for Korean students i.e. all but the teachers' children)* **Interviewer:** Gina? **Gina**: Um the same as Sarah. I choose Computer Science, History,	

Mandarin and Korean **Interviewer**: *(looks at Kay)* **Kay**: I am the same with them. I chose Computer Science, History, Mandarin and Korean. **Interviewer:** That's interesting! Did you choose the same because you're friends? **All**: No! **Interviewer:** Why did you choose the options you chose? **Sarah**: Because firstly Korean was my native language and I, we had no choice but to choose Korean. Computer Science I always got good grades in Computer Science and it was really fun so I definitely chose it. Um and Mandarin, I was doing Mandarin last year and I couldn't like give up because I was learning for one year or so. And lastly History, I also like History better than Geography so I chose History. **Interviewer:** Gina? **Gina**: Um firstly, I choose Korean because it is my native language and secondly, um History, I choose History because I didn't like Geography. Well I didn't really like History, but I didn't ah, I hate Geography so I chose History. And Mandarin, I like Mandarin I was in Miss Jeong's class and I heard that when we get to year 10, I am, I can go to Mrs Lu's class, which um I wanted to go, and I heard that she teaches very well and I also didn't want to give up Mandarin and I also like Mandarin so I choose Mandarin, and I choose Computer Science because um I want to work in Computer thing in the future, so I thought it would be helpful to choose Computer Science, well I	

like Computer Science as well, but I thought it would be helpful to learn GCSE and IB as well. **Interviewer:** Okay! Kay. **Kay**: Um, same as them, I chose Korean because it is my native language and for History, compared to Geography its more um thought-provoking subject, which allows me to think rather than actually collect um data and analysing them. For Mandarin I spent a long time doing it so I kind of just let it go. And Computer Science, I tried to change it to dance but um I thought Computer Science was more complimentary with me so I chose it. **Interviewer**: Okay! So, you mentioned specific teachers teaching Mandarin, do the teachers influence your decisions or not? **Gina**: I think kind of. Because like, not other subjects but I think Mandarin it actually had an effect like because if they said that I am going to be in Mrs Jeong's class in year 10 as well and then I think I will have gave up Mandarin, given up Mandarin and I think I would have chosen something like drama but because I heard that I can go into, I am going into, Mrs Lu's class I just choose to do Mandarin because, I, Mrs Lu class she teaches very well and she gives out extended vocabularies and like um compared to Mrs Lu, Mrs Jeong teaches oh kind of um so *(awkward laugh)* **Interviewer**: What about your Sarah? **Sarah**: Yes I think teachers hold great sway over our choice. Because, like when I was choosing History over Geography the fact that our like last year teacher was Mr.. (**Gina**: Probert) No Geography (**Gina**: Ah Geogrpahy) (**Kay**: Mr Carter-Stead) Mr Carter Stead influenced my decision because um we didn't really like his teaching and I	

thought and like some teachers certain teachers make some subjects boring so I think it really holds great sway over our choice.

Interviewer: Kay?

Kay: Um I think for me the subject choices matter more than teachers um and I think there that are new teachers I can't actually choose so I thought that um yeah choosing what I want to do is more important than the teacher.

Interviewer: I have only a couple more questions now. How influential were your parents in your decision-making?

Sarah: None! I basically chose it by myself. And I after choosing I just told my parents and they said yes ok it's fine and so it was um 100% my choice.

Gina: Um, same as Sarah. They didn't really care about my choosing. I mean they let me choose it alone and I told my parents like I talked with them like because I was confused with my decision to choose Computer Science or mandarin or like that but they just um they talked but didn't force me to something, they just let me to choose something. I choose it.

Kay: For me, I did consult my parents before I did the choices. Um, they wanted me to do some more challenging stuff rather than that I could easily do. Um, so we chose some advanced type subjects.

Interviewer: There are 19 girls that do Computer Science at IGCSE and 58 boys. Why do think there are more boys choosing Computer Science than girls?

Kay: I think the fact that they are more um close with the computer

since they are young, because of gaming or watching sports games compared to girls who actually do stuff um slightly different than that. Um, I think the fact they are more familiar with computing and the fact that their brain is kind of um the different way that acts more um acts better with the computer so I think that's why they find it easy. I think so. **Interviewer**: Okay. **Gina**: I have similar reason with Kay but to add I am not sure. Computer Science compared to other subjects it is kind of more, it is easier to achieve a good grade, a like A* I think some of them just choose it choose um Computer Science because they don't have like other things like special areas to do and also well it is also mixed with her reason. They are more relevant to computer and they think they choose Computer Science, well not most of them, but some of them choose Computer Science because they got a good grade from Computer Science more than others. **Sarah**: Yes I basically got the same reasons with them. So I think that are already existing social stereotypes like boys to computer and girls do dance its like subliminally somehow control like their thinking and umm that's because girls tend to choose dance and drama more than boys. **Interviewer:** Thank you for your candid responses. I have no more questions for you.	

Seo Yon, Dain, ChaeEun (Female IGCSE Computer Science students) – New Year-10 students to the school

Transcript	Theme
Interviewer: What were your IGCSE options?	

Seo Yon: I choose Computer Science, and History and Mandarin.	
Interviewer: Dain?	
Dain: Um History, Geography and Computer Science.	
ChaeEun: Same with her (indicating Dain)	
Interviewer: So why did you choose those subjects?	
Dain: Shall I go first? So firstly, I wasn't very good at creative things, like drama, art. That was out of the question. So, I chose stuff that I could, I actually chose French first so I got Computer Science instead. History and Geography were sort of they were things that I liked to study. French was something I wanted to go for English literature in future so. So other languages were pretty good to study.	Creative Language Enjoyment
Interviewer: So you were pretty much forced to choose Computer Science.	
Dain: Yeah but, I thought, it is a lot different from what I was expecting and so I like it.	Enjoyment
Interviewer: So what were you expecting?	
Dain: It was something more. I don't what I was expecting but it wasn't like this.	
Interviewer: Okay. That's interesting! Either of you two?	
ChaeEun: For me, I chose Computer Science as a fourth choice because in my previous school we like mainly did digital technology,	

as like design parts so we created things in Photoshop and printed things with 3D printers but this is mostly programming so is different from what I expected. But I chose History and Geography because I enjoy Geography so I choose Geography. But I didn't manage to choose History, but then I wanted to do Spanish for the same reason as her [Dain] but I was a new student that I cannot join in so I choose History instead.	Enjoyment Language
Seo Yon: Actually, I choose Computer Science subject first because before I came to this school I did not know they did a subject like Computer Science but actually I was already interested in coding and programming and so at first when I see Computer Science was in the subject and I can learn I was happy to see and happy learn about it so I choose it first. Then for the Mandarin, my mum, my parents wanted me a lot to do the language for my future and university or something so I also choosed and also the History actually the Korean History before I was studying in the normal school so I just choose History and all then all three combine.	Interest Happy Parents Language
Interviewer: You mentioned that your parents wanted you to study Mandarin. How did they feel about you choosing Computer Science?	
Seo Yon: Ah, they also were curious about it. Actually I did never say I wanted to learn about the coding and programming because actually I was interesting in medical or chemistry and not engineering or the computer things. But I also wanted to learn about future work, maybe not I am going to do the future work in Computer Science but I want to learn and it is a really good opportunity to choose it so my parents also agreed and think it is good for me.	Interest Future Parents
Interviewer: and you parents (indicating Dain)	

Dain: About my subjects? **Interviewer**: Yeah **Dain**: So, History and Geography they…. It was mostly my dad I talked to when I was doing my subjects and I wanted to do French or Spanish, one of the two and he was supportive because he knows I want to do literature and then History and Geography he thought they were my best bet because I don't have creative things to do and when I told him it had changed to Computer Science he was oh that is interesting but um because I don't know he just wants me to do what I want to do so it was nothing much. It was mostly my choices really.	Parents Language Support
Chae Eun: For me, my parents didn't think that I am good at creative subject, mostly art or drama because I wasn't confident so I didn't choose any of them. So when I turned to Computer Science. I was interested in this part. I really wanted to do Mathematic or engineering parts and my Maths and Physics were bad. (laugh) so I couldn't challenge for it. My parents totally agree with my choosing Computer Science and other subjects too.	Parents Creative
Interviewer: Since you are all new to the school, your teachers would have had little bearing on your choices. None of that applies to you. So know that you are studying Computer Science what are your impressions? How does it compare to your expectations?	Expectations
Chae Eun: With designing and creating websites. It is more creative than I thought it would be but it in a good way because it is not physically creative, cause like I am not good at drawing or acting or something	Creative
Seo Yeon: It is thinking creating.	Creative

Dain: I am not good at Math or creative. I am more of a literature kind of girl. It is a language but a language is difficult to learn so I am getting used to it but it is not easy. **Interviewer**: That is great! Thank you so much.	Creative Language

Senna (Female IGCSE Computer Science students)

Transcript	Theme
Interviewer: What were your IGCSE options?	
Senna: At first, I chose Geography, Mandarin and Dance and then I changed to Computer Science instead of Mandarin.	Switch
Interviewer: Why was that?	
Senna: Um, because first of all the reason I chose Mandarin was mainly because of my parents because they wanted me to. Ah, I think I am more like Math Mathsy or Sciencey type person and I didn't really like Mandarin because I had to memorise all of those characters, so I persuaded my parents and I changed.	Parents STEM Dislike Memorise Switch
Interviewer: So, why did they want you to do Mandarin? Do you know?	
Senna: They think that it will be useful but I don't know. (awkward laugh)	Future use
Interviewer: Do you think that in general students' parents influence their IGCSE choices, or not?	
Senna: I think they are because the parents are the ones who have to	

sign the forms at the last date and if they are not happy they will change it. **Interviewer**: So why did you choose Geography? **Senna**: Because I thought it was better than History (laughing) **Interviewer**: better in what way? **Senna:** More interesting, more enjoyable. **Interviewer**: and your other option? **Senna**: Dance! **Interviewer**: why did you choose dance? **Senna**: There were, there was not, none of the other, I wasn't confident with any of the other arts subjects. **Interviewer**: Really? **Senna**: Yes **Interviewer**: But you are very, very good at dance anyway. **Senna**: Not very good! **Interviewer**: Yes you are, I've seen you dance. I was really impressed. So do the teachers try and influence you over your choices.	Parents decide Superior Interest Enjoyment Creative subject Limited options Confidence Modest

Senna: I don't think so. **Interviewer**: So does the teacher influence your choice of subject? **Senna**: No **Interviewer**: So, these options that you choose at this age group, so you think they influence was you do at IB, although there are some IB subjects that you don't need to study at IGCSE, like economics? **Senna**: I think that they are important somehow because it gives us the broad way that we will do in IB or in the future. **Interviewer**: There are 58 boys that do Computer Science and 20 girls and also if you look at universities only 20% of Computer Science graduates are female. Why do you think there are more boys than girls do computer Science? **Senna**: I don't know if other countries think like this but in Korea, parents, adults tend to think like that boys are better at Maths and Sciences and girls are better at other subjects but I think the boys are encouraged more to study Maths and computer stuff than girls or they might think like that themselves because they were taught like that. **Interviewer**: So do you think that it is the parents we have to convince? **Senna**: Partly. **Interviewer**: So do think it is important to try to make it more balanced in terms of males and females?	No Teacher influence No Teacher influence Future influence Parents beliefs STEM stereotype Parent Boys encouraged Legacy Parent

Senna: If the girls want to (laugh) they can study. **Interviewer**: Okay, for example in the first year that IGCSE Computer Science was offered at this school only 5 girls compared to 34 boys chose to study Computer Science. Would you find that intimidating, choosing a subject with such few girls? **Senna**: No! **Interviewer**: No? **Senna**: No! **Interviewer**: Thank you	Free choice No intimidation

Linus and Eric (Male IGCSE Computer Science students)

Transcript	Theme
Interviewer: Thank you both for agreeing to answer a few questions. We'll start with Eric if that is okay with you both. So, what did you choose for IGCSE?	
Eric: For IGCSE I chose Geography, Computing Science, err, I can't remember, ah yes Drama.	Creative
Interviewer: Why did you choose those options?	
Eric: Because first of all I am not sure what I am going to do in future so I chose what I can do best and by doing the subjects I am good I plan to know what I am good at and choose my dream.	Future Ability
Interviewer: Okay, so do you enjoy drama?	

Eric: Yes	
Interviewer: and Computer Science?	
Eric: Yes	
Interviewer: What about Geography?	
Eric: It's quite hard but I think it helps my, it helps to develop my knowledge?	Difficulty
Interviewer: Why Geography instead of History? *(students are required to take one or the other)*	
Eric: Because in History I have to memorise important person's name and since there are too many people, important people I can't memorise everything but in Geography all I have to do is just, I understand what is going on.	Memory Difficulty
Interviewer: What about you Linus? When you took your option choices what did you choose?	
Linus: Drama, Computer Science and (turning to Eric) then what was my last one? History!	Creative
Interviewer: Why did you choose History and not Geography?	
Linus: Personally, I like analysing stuff rather than measuring things.	Analyse
Interviewer: Okay, and Drama, why did you choose that subject?	
Linus: Because I enjoy acting and then um stage fright was my only	Enjoyment

drawback but I enjoyed overcoming it so…	
Interviewer: And Computer Science?	
Linus: It's fun to program, like games so…	Enjoyment
Interviewer: What about, do you have an idea what you want to do at university?	
Eric: No, not yet!	Future
Interviewer: Did your choices, your options, have any bearing on future career pathways?	
Eric: Yes, I just choose what my mum told me to choose and what I am good at.	Parent Ability
Interviewer: So your parents wanted you to choose these subjects.	Parents
Eric: Yes	
Interviewer: And you Linus, did you discuss your option choices it with your parents?	Parents
Linus: Yes	
Interviewer: And what did they say?	
Linus: They said do what ever you want.	Parents
Interviewer: So did you choose subjects that you thought your parents would approve of?	

Linus: No	
Interviewer: Do you have an idea of what you want to study at University or what you want to do for a career?	
Linus: I wanted to um, learn about robotics and engineering but recently I am not sure whether I am going to drama or technology.	Future
Interviewer: Okay, do you think that it is common for parents to influence students in their option choices?	
Eric: I think for me, the parents gave me lots of choices so first they made me to choose what I want and if I can't choose between, for example, History and Geography they choose it for me and I just agree.	Parents
Interviewer: And what about other students, do you think parents influence their decisions?	
Eric: Yes, I think they are the same.	Parents
Interviewer: And you Linus, do you think that parents influence IGCSE choices?	
Linus: I'm not sure. Some might!	Parents
Interviewer: Eric, do you think teachers influence IGCSE choices?	
Eric: Yes, last year I studied Mandarin and the teacher's attitude was so bad that I didn't want to learn Mandarin and my parents agreed with that. But, I think if the teacher is kind then they study very hard and try to learn.	Teacher Kindness

Interviewer: What about you Linus? Did the teachers try and persuade you to choose their subjects?	
Linus: Not much. Well, one teacher said I should do dance but it was just a joke I think.	Teacher
Interviewer: So do you think that your IGCSE choices influence your IB options or not?	
Linus: Of course, yes!	Future
Interviewer: *(looks at Eric)*	
Eric: Yes. I wouldn't choose a subject *(at IB)* I hadn't already studied first.	Future
Interviewer: What about students in general? Do you think if a student didn't choose a subject at IGCSE they would be likely to go back to it ?*(looking at Linus)*	
Linus: Some might. I've seen students do that.	
Interviewer: and your opinion, Eric?	
Eric: I don't know.	
Interviewer: Right, there are 58 boys taking Computer Science compared to 20 girls this year. Why do you think more boys than girls take Computer Science?	
Eric: Umm, I am not sure but I think it is because they are generally very good at computing and boys I think boys, boys, spend more time on computers than girls because they play more games and they	Ability Gender difference

download games to play more games and I think, I think, that if they know more computers they can download games more safely and play. **Interviewer**: and Linus, your thoughts on why there are so many more boys than girls? **Linus**: I think its cause boys use more time on a computer, than girls, obviously because of gaming and (xshow? – *difficult to hear exactly what was said*) things. So they might be more used to computer stuff. **Interviewer**: That's it! Thank you both for your time. Your answers are extremely useful to this research and as I said at the beginning, your identity and responses will be kept confidential.	 Gaming

Appendix S: Emerging Themes

Name	Sources	Referen...
Ability	4	8
Confidence	2	2
Creativity	3	5
Diffiiculty	5	10
Dislike	3	4
Enjoyment	5	15
Future	5	13
Gender difference	5	12
Genetic	3	3
Interest	5	13
People	6	40
Friends	2	2
Parents	6	26
Teacher	4	12
Required	2	2

Interviewer: So when you took your option choices what did you choose?

Janice: Art, History and Mandarin

Interviewer: Ok, so why did you choose Art, History and Mandarin?

Janice: First of all, I chose History because I have to at least choose one of the Humanities subjects. So I prefer History than Geography and I thought History is very interesting subject so I chose History and for Mandarin, it is very important to speak another language instead of English and Korean, which is my mother tongue. And China is like increasingly getting more developed and so parents and I thought speaking Chinese will help my future career and help me to be more successful in the future so I chose Mandarin. And for Art, I had thought I had to balance my life and so if I only focus on academic stuff I won't be able to um manage a lot of stuffs and so my brain might just explode (small laugh). I like Art and I like Art History, it helps to calm down myself and so I chose Art.

Interviewer: You said that you think Mandarin will be useful career-wise in the future. What do you see yourself doing?

Janice: Cause um as you know China is the biggest population. That means that they the people the economy is very productive so their companies and their economics are getting bigger and they are becoming more modernising industrialised and you know Korea is near to China and so if I work in Korea and work global if I work in Korea there will be a lot of stuffs that I have to cooperate with Chinese companies and a lot of economies predict that China will be develop as much as US economy in the future. So, yeah! A lot as I said before, China is the biggest economy in the world and so a lot that means that means most of world population speaks China so if I speak Chinese I will be able to communicate with a lot of most people.

Appendix T – Word Tree for the word 'parent'