

# **SECONDARY SCHOOL GIRLS' EXPERIENCES OF PAIR-PROGRAMMING IN INFORMATION TECHNOLOGY**

**JANET ADRI LIEBENBERG**

**BSc, HED, Hons BEd.**

**10088873**

**Dissertation submitted in fulfilment of the requirements for the degree**

**MAGISTER EDUCATIONIS**

**at the**

**Potchefstroom Campus**

**of the**

**North-West University**

**Supervisor: Prof E Mentz**

**Co-supervisor: Mrs EA Breed**

**April 2010**

## ACKNOWLEDGEMENTS

A word of heartfelt thanks to the following persons, who, each in a particular capacity, contributed in making this study possible:

- Prof. Elsa Mentz for her enthusiasm, professional, though humane guidance and continued support.
- Betty Breed for her wisdom, support and contributing in her own special way.
- Various friends and family members for support and encouragement.
- My dad, mom and sisters for their support – be it financially, baby-sitting or just listening.
- The North-West Department of Education for permission to carry out the empirical study.
- The six girls whom this whole study is all about.
- Rudolf Homan for being open-minded and agreeing to implement pair programming in his class.
- Jackie Viljoen for language editing and proof reading.
- My husband Christiaan for his support and prayers whilst handling his own work pressures.
- My three small children for unconditional love, hugs, kisses and sitting on my lap while I'm working on the computer – it gave me strength to carry on.
- Emily Mphoseloa for keeping the household afloat and serving numerous mugs of coffee.
- Above all our Heavenly Father for blessing me with this opportunity to study a mere speckle in His wonderful creation.

This research is based on work financially supported by the National Research Foundation (NRF) in South Africa. Any opinion, findings and conclusions or recommendations expressed in this material are those of the authors and therefore the NRF does not accept any liability in regard thereto.

## SUMMARY

The low number of girls in the secondary school phase taking Information Technology as a subject, is a reason for great concern in this era of rapid technological development we are living in. The pipeline shrinkage problem (females not persisting in the computer field) is another concern and requires interventions that makes girls understand the importance of the subject IT, and makes the subject at school level more enjoyable for them.

Pair programming, whereby two programmers work at one computer on the same programming task, shows several promising properties for educational purposes. Furthermore, pair programming seems to have a positive effect in general on female computer science students at universities specifically in terms of enjoyment and on their view of the importance of the subject and a career in IT and resulting in an improvement in their enrollment and retention rate in IT.

This research was done to understand how pair programming shapes secondary school girls' experiences in the IT class. The literature study investigates girls' attitudes towards computers and the factors that have an effect on girls attitudes and involvement in the computer environment. Furthermore a review of pair programming was done with attention to the advantages of pair programming in general and specifically to girls. The empirical study was aimed at determining to what extent does pair programming change girls' enjoyment of programming and the subject IT, as well as the importance they attach to programming, the subject IT and to a career in IT. Analysis of the girls' responses before and after pair programming leads to the conclusion that pair programming increases girls enjoyment and view of importance of programming, the subject IT and a career in IT. This knowledge can convince IT teachers that pair programming is an untapped resource worth considering in any programming class, but especially to attract and retain more girls to IT.

**Keywords:** Girls, pair programming, enjoyment, importance, programming, the subject IT, and IT career.

## OPSOMMING

Die klein getal meisies wat Inligtingstechnologie in die sekondêre skoolfase as vak neem is 'n rede tot groot kommer in hierdie era van snelle tegnologiese ontwikkeling waarin ons leef. Die *pipeline shrinkage problem* (vrouens wat nie aanbly en volhard in die rekenaarveld) is 'n verdere bekommernis en vereis intervensies wat meisies die belangrikheid van die vak IT sal laat besef, en wat sal veroorsaak dat meisies die vak op skoolvlak meer sal geniet.

Paarprogrammering, waar twee programmeerders by een rekenaar aan dieselfde programmeringstaak werk, toon verskeie belowende eienskappe vir opvoedkundige doeleindes. Verder wil dit voorkom asof paarprogrammering in die algemeen 'n positiewe invloed het op vroulike rekenaarwetenskapstudente by universiteite maar ook spesifiek in terme van hul genot, en hul beskouing van die belangrikheid van die vak en 'n beroep in IT, wat weer 'n verbetering in die inskrywings en volharding in IT tot gevolg het.

Hierdie studie is gedoen om te verstaan hoe paarprogrammering sekondêre skoolmeisies se ervarings in die IT klas vorm. Die literatuurstudie ondersoek meisies se houding teenoor rekenaars en die faktore wat 'n rol speel in meisies se houdings en betrokkenheid in die rekenaaromgewing. Verder is 'n studie van paarprogrammering gedoen waarin aandag gegee is aan die voordele van paarprogrammering in die algemeen en ook spesifiek vir meisies. Die empiriese studie is daarop gemik om te bepaal tot watter mate verander paarprogrammering meisies se genot van programmering en die vak IT, asook die waarde wat hul heg aan programmering, die vak IT en 'n IT-beroep. Analisering van die meisies se antwoorde voor en na paarprogrammering lei tot die gevolgtrekking dat paarprogrammering meisies se genot en waardebeskouing van programmering, die vak IT en 'n IT-beroep vermeerder. Hierdie kennis kan IT-onderwysers oortuig dat paarprogrammering 'n onontginde hulpbron is wat oorweeg kan word in enige programmering klas, maar spesifiek ook om meer meisies te kan lok en behou vir IT.

Keywords: Meisies, paarprogrammering, geniet, belangrik, programmering, die vak IT, en IT-beroep.

## TABLE OF CONTENTS

SUMMARY.....	ii
OPSOMMING.....	iii
LIST OF TABLES.....	xi

### Chapter 1 Orientation

1.1 BACKGROUND AND PROBLEM STATEMENT.....	1
1.2 CLARIFICATION OF TERMINOLOGY.....	3
1.3 LITERATURE REVIEW.....	4
1.4 RESEARCH QUESTIONS.....	7
1.5 AIM OF THE RESEARCH.....	8
1.6 RESEARCH DESIGN AND METHODOLOGY.....	8
1.6.1 Literature study.....	8
1.6.2 Empirical investigation.....	8
1.6.2.1 Research design.....	8
1.6.2.2 Study participants.....	9
1.6.2.3 Data collection methods.....	9
1.6.2.4 Data collection.....	9
1.6.2.5 Data analysis.....	10
1.6.2.6 Trustworthiness.....	10
1.6.2.7 Ethical aspects of the research.....	10
1.7 STRUCTURE OF DISSERTATION.....	11
1.8 LIMITATIONS OF THE STUDY.....	11
1.9 CONTRIBUTION OF THE STUDY.....	11

### Chapter 2 Girls and the computer environment

2.1 INTRODUCTION.....	12
2.2 GIRLS' PREFERENCES REGARDING THE COMPUTER ENVIRONMENT.....	13
2.3 THE ATTITUDES OF GIRLS TOWARDS COMPUTERS.....	13
2.3.1 Enjoyment.....	14

2.3.2	Importance, interest and persistence.....	15
2.3.3	Anxiety and confidence.....	17
2.3.3.1	Anxiety.....	17
2.3.3.2	Confidence.....	18
2.4	GENERAL FACTORS THAT HAVE AN EFFECT ON GIRLS' ATTITUDES TOWARDS COMPUTERS AND INVOLVEMENT IN THE COMPUTER ENVIRONMENT.....	19
2.4.1	Societal influences.....	19
2.4.1.1	Parents.....	20
2.4.1.2	The media.....	20
2.4.1.3	The male culture of IT.....	20
2.4.1.4	Race and ethnicity.....	21
2.4.1.5	Socio-economic status.....	22
2.4.2	Age.....	22
2.4.3	Experience.....	23
2.4.4	Games and other software.....	23
2.5	FACTORS IN THE CLASSROOM THAT HAVE AN EFFECT ON GIRLS' ATTITUDES TOWARDS COMPUTERS AND INVOLVEMENT IN THE COMPUTER ENVIRONMENT.....	24
2.5.1	The influence of peers on girls.....	24
2.5.2	Teachers' effect on girls.....	25
2.5.3	The effect of teaching strategies on girls.....	26
2.5.3.1	Collaboration.....	26
2.5.3.2	Mentors and role models.....	27
2.5.4	The IT curriculum.....	28
2.5.5	The role of programming in IT.....	29
2.5.6	Academic achievement of girls in IT.....	30
2.6	INTERVENTIONS.....	31

## Chapter 3

### Pair programming in the IT class

3.1	INTRODUCTION.....	35
3.2	THE ORIGIN AND DEVELOPMENT OF PAIR PROGRAMMING.....	36
3.3	THE PROGRAMMERS IN A PAIR.....	37
3.3.1	Roles of programmers in a pair.....	37

3.3.2	The composition of programming pairs.....	38
3.3.3	Attributes of successful pairs.....	40
3.3.4	Principles for successful pair programming.....	41
3.4	THE FACILITATOR IN THE PAIR PROGRAMMING CLASS.....	42
3.4.1	Guidelines for facilitators of pair programming.....	42
3.4.2	Pair programming rules for secondary school learners.....	46
3.5	ADVANTAGES OF PAIR PROGRAMMING.....	47
3.5.1	A list of advantages of pair programming.....	47
3.5.1.1	Enjoyment.....	48
3.5.1.2	Attitudes.....	48
3.5.1.3	Confidence.....	48
3.5.1.4	Persistence.....	48
3.5.1.5	Program quality.....	49
3.5.1.6	Program completion and submission.....	49
3.5.1.7	Time taken.....	49
3.5.1.8	Bugs.....	50
3.5.1.9	Program design.....	50
3.5.1.10	Teacher workload.....	50
3.5.1.11	Learning and comprehension.....	50
3.5.1.12	Learner achievement.....	51
3.5.1.13	Learner morale.....	51
3.5.1.14	Motivation and focus.....	52
3.5.1.15	Getting stuck.....	52
3.5.1.16	Cheating.....	52
3.5.1.17	Communication.....	52
3.5.1.18	Problem solving.....	53
3.5.1.19	Code Reviews.....	53
3.5.1.20	Introduction to programming.....	53
3.5.1.21	Going solo.....	54
3.5.1.22	Limited resources.....	54
3.5.1.23	Social and career benefits.....	54
3.5.2	Behaviours behind the advantages of pair programming.....	54
3.6	DIFFICULTIES OF PAIR PROGRAMMING.....	55
3.6.1	Experience conflicts.....	55
3.6.2	Scheduling conflicts.....	56
3.6.3	Reliability conflicts.....	56
3.6.4	Conflict in effort.....	56

3.6.5	Personality clash.....	57
3.6.6	Disagreements.....	57
3.6.7	Programming style differences.....	57
3.6.8	Distractions.....	57
3.6.9	Bad communication.....	58
3.6.10	Hard to reward talent.....	58
3.6.11	Accountability and co-dependence.....	58
3.6.12	Comprehension.....	58
3.6.13	Implementation of pair programming.....	58
3.6.14	Noise.....	59
3.7	GIRLS AND PAIR PROGRAMMING.....	59
3.7.1	Advantages of pair programming to girls.....	59
3.7.1.1	Enjoyment.....	59
3.7.1.2	Confidence.....	60
3.7.1.3	Interest.....	60
3.7.1.4	Persistence.....	60
3.7.1.5	Social context.....	61
3.7.1.6	Efficiency.....	61
3.7.1.7	Program quality.....	61
3.7.1.8	Academic achievement.....	61
3.7.1.9	Time management.....	61
3.7.1.10	Safety.....	61
3.7.1.11	Stereotyping.....	62
3.7.2	Disadvantages for girls.....	62
3.8	SUMMARY.....	62

## Chapter 4

### Research methodology and data analysis

4.1	INTRODUCTION.....	63
4.2	RESEARCH QUESTIONS.....	63
4.3	AIM OF THE RESEARCH.....	63
4.4	RESEARCH DESIGN AND METHODOLOGY.....	64
4.4.1	Research design.....	64
4.4.2	Study participants.....	64
4.4.3	Data collection methods.....	65



4.4.4	Data collection.....	65
4.4.5	Data analysis.....	68
4.4.6	Trustworthiness.....	68
4.5	RESULTS.....	69
4.5.1	RESPONDENT 3.....	70
4.5.1.1	THE PERSONAL STORY.....	70
4.5.1.2	THE SUBJECT IT.....	70
4.5.1.2.1	Enjoyment of the subject IT.....	70
4.5.1.2.2	The importance of the subject IT.....	71
4.5.1.2.3	The attributes of the subject IT.....	71
4.5.1.2.4	Achievement in the subject IT.....	72
4.5.1.3	PROGRAMMING.....	72
4.5.1.3.1	Enjoyment of programming.....	72
4.5.1.3.2	The importance of programming.....	73
4.5.1.3.3	Skills required for programming.....	73
4.5.1.4	A CAREER IN IT.....	74
4.5.1.4.1	The importance of a career in IT.....	74
4.5.1.4.2	Future plans.....	75
4.5.1.5	PAIR PROGRAMMING.....	75
4.5.1.5.1	Pairing versus Solo.....	75
4.5.1.5.2	The enjoyment of pair programming.....	76
4.5.1.5.3	The advantages of pair programming.....	76
4.5.1.5.4	Difficulties of pair programming.....	80
4.5.1.5.5	Pair programming in the workplace.....	80
4.5.1.6	ATTRACTING GIRLS TO IT.....	80
4.5.1.6.1	Interests of girls.....	80
4.5.1.6.2	Reasons and solutions for the shortage of girls in IT.....	81
4.5.1.6.3	Boys in the equation.....	81
4.5.2	RESPONDENTS 1, 2 AND 4.....	82
4.5.2.1	THE PERSONAL STORIES.....	82
4.5.2.1.1	Respondent 1.....	82
4.5.2.1.2	Respondent 2.....	83
4.5.2.1.3	Respondent 4.....	83
4.5.2.2	THE SUBJECT IT.....	84
4.5.2.2.1	Enjoyment of the subject IT.....	84
4.5.2.2.2	The importance of the subject IT.....	85
4.5.2.2.3	The attributes of the subject IT.....	85

4.5.2.2.4	Achievement in the subject IT.....	86
4.5.2.3	<b>PROGRAMMING</b> .....	87
4.5.2.3.1	Enjoyment of programming.....	87
4.5.2.3.2	The importance of programming.....	88
4.5.2.3.3	Skills required for programming.....	89
4.5.2.4	<b>A CAREER IN IT</b> .....	90
4.5.2.4.1	The importance of a career in IT.....	90
4.5.2.4.2	Future plans.....	91
4.5.2.5	<b>PAIR PROGRAMMING</b> .....	92
4.5.2.5.1	Pairing versus Solo.....	92
4.5.2.5.2	The enjoyment of pair programming.....	93
4.5.2.5.3	The advantages of pair programming.....	94
4.5.2.5.4	Difficulties of pair programming.....	99
4.5.2.5.5	Pair programming in the workplace.....	99
4.5.2.6	<b>ATTRACTING GIRLS TO IT</b> .....	100
4.5.2.6.1	Interests of girls.....	100
4.5.2.6.2	Reasons and solutions for the shortage of girls in IT.....	101
4.5.2.6.3	Boys in the equation.....	102
4.5.3	<b>RESPONDENTS 5 AND 6</b> .....	103
4.5.3.1	<b>THE PERSONAL STORIES</b> .....	103
4.5.3.1.1	Respondent 5.....	103
4.5.3.1.2	Respondent 6.....	103
4.5.3.2	<b>THE SUBJECT IT</b> .....	104
4.5.3.2.1	Enjoyment of the subject IT.....	104
4.5.3.2.2	The importance of the subject IT.....	105
4.5.3.2.3	The attributes of the subject IT.....	105
4.5.3.2.4	Achievement in the subject IT.....	106
4.5.3.3	<b>PROGRAMMING</b> .....	107
4.5.3.3.1	Enjoyment of programming.....	107
4.5.3.3.2	The importance of programming.....	108
4.5.3.3.3	Skills required for programming.....	108
4.5.3.4	<b>A CAREER IN IT</b> .....	109
4.5.3.4.1	The importance of a career in IT.....	109
4.5.3.4.2	Future plans.....	110
4.5.3.5	<b>PAIR PROGRAMMING</b> .....	110
4.5.3.5.1	Pairing versus Solo.....	110
4.5.3.5.2	The enjoyment of pair programming.....	112

4.5.3.5.3	The advantages of pair programming.....	112
4.5.3.5.4	Difficulties of pair programming.....	115
4.5.3.5.5	Pair programming in the workplace.....	116
4.5.3.6	<b>ATTRACTING GIRLS TO IT.....</b>	<b>116</b>
4.5.3.6.1	Interests of girls.....	116
4.5.3.6.2	Reasons and solutions for the shortage of girls in IT.....	116
4.5.3.6.3	Boys in the equation.....	118
4.5.4	<b>MORE RESULTS.....</b>	<b>119</b>
4.5.4.1	<b>Exams and tests.....</b>	<b>119</b>
4.5.4.2	<b>Technological era.....</b>	<b>120</b>
4.5.4.3	<b>Different methods and solutions.....</b>	<b>120</b>
4.5.4.4	<b>The teacher.....</b>	<b>121</b>
4.5.4.5	<b>A computer scientist.....</b>	<b>121</b>
4.6	<b>SUMMARY.....</b>	<b>122</b>

## Chapter 5

### Discussion of findings, conclusions and recommendations

5.1	<b>SYNOPSIS OF STUDY.....</b>	<b>123</b>
5.2	<b>DISCUSSION OF THE FINDINGS OF THIS STUDY.....</b>	<b>123</b>
5.2.1	<b>SYNOPSIS OF THE LITERATURE STUDY.....</b>	<b>124</b>
5.2.1.1	Girls and the computer environment.....	124
5.2.1.2	Pair programming in the IT class.....	125
5.2.2	<b>DISCUSSION OF THE FINDINGS OF THE EMPIRICAL STUDY.....</b>	<b>127</b>
5.2.2.1	Response to Subquestion 1: .....	127
5.2.2.2	Response to Subquestion 2: .....	130
5.2.2.3	Response to Subquestion 3:.....	131
5.2.2.4	Response to Subquestion 4:.....	132
5.3	<b>CONCLUSIONS ARISING FROM THIS STUDY.....</b>	<b>133</b>
5.4	<b>RECOMMENDATIONS ARISING FROM THIS STUDY.....</b>	<b>134</b>
5.5	<b>SHORTCOMINGS IN THE EMPIRICAL STUDY.....</b>	<b>135</b>
5.6	<b>RECOMMENDATIONS FOR FURTHER RESEARCH.....</b>	<b>135</b>
5.7	<b>FINAL REMARKS.....</b>	<b>136</b>
	<b>BIBLIOGRAPHY.....</b>	<b>137</b>

<b>APPENDIX A:</b>	<b>Networks of code families</b>
<b>APPENDIX B:</b>	<b>Letter to Department of Education</b>
<b>APPENDIX C:</b>	<b>Letter from Department of Education</b>
<b>APPENDIX D:</b>	<b>Letter to school principal</b>
<b>APPENDIX E:</b>	<b>Letter from NWU Ethics Committee</b>
<b>APPENDIX F:</b>	<b>Letters from language editor</b>
<b>APPENDIX G:</b>	<b>Form for informed consent</b>

## List of tables

<b>Table 1.1:</b>	Number of learners taking Information Technology in the North-West Province.....	<b>1</b>
<b>Table 1.2:</b>	Research subquestions.....	<b>7</b>
<b>Table 4.1:</b>	Questions used as guidelines for the interviews.....	<b>66</b>

## Chapter 1

### Orientation

#### 1.1 BACKGROUND AND PROBLEM STATEMENT

In a time of great technological advancements where the computer plays an ever-increasing role and women increasingly take up positions in the labour market, the low number of girls in the secondary school phase taking Information Technology (IT) is conspicuous. This situation is not limited to South Africa. In the United States of America (USA), the American Association of University Women's (AAUW) commission of experts (academics, educators, businesswomen and journalists) published two reports that focused specifically on this problem (AAUW, 1998; AAUW, 2000). The Association for Computing Machinery (ACM) also has a committee on women in computing. Their so-called Ambassador Program consists of people in Germany, India, Australia, to name but a few, who report on the same phenomenon (ACM-W, 2006).

Table 1.1 below reflects statistics provided by the North-West Department of Education (South Africa) of the number of males and females taking IT as a subject at a number of schools in the province. In 2007, the Grade 12s were still on the old Computer Studies curriculum; some schools in the province had to be omitted because of erroneous or incomplete information. It is obvious from this table that the females are far fewer than the males. Another alarming phenomenon is the low retention rate of learners, especially the females, in their first year of taking the subject: in these schools, there were 44 Grade 11 girls in 2008 as opposed to the 80 who had started the subject in 2007.

**Table 1.1** Number of learners taking Information Technology in the North-West Province

School	2007				2008					
	Grade 10		Grade 11		Grade 10		Grade 11		Grade 12	
	Male	Female	Male	Female	Male	Female	Male	Female	Male	Female
Wolmaransstad	4	8	6	1	5	2	1	3	5	1
Brits	21	6	12	3	12	2	11	6	12	3
Rustenburg	17	12	16	2	11	4	4	4	9	0
New Vision	2	8					2	5		
Stella	3	7	2	2	3	2	3	4	2	2
Gimnasium	29	1	13	1	19	4	22	1	13	1
Fields College	12	3	6	3	6	9	14		6	4
Volkskool	10	1	9	2	16	1	5	1	8	0
Hartbeespoort	12	1	12	2	17	8	8	1	12	1
Zinniaville			7	6	5	12			6	6
Fochville	4	5	10	1			2	2	6	
Schoonspruit	16	13			10	1	12	8	9	5
Klerksdorp	31	13	22	7	20	10	25	9	18	6
Ventersdorp	6	2	3	3	1	1	3		3	3
<b>Total</b>	<b>167</b>	<b>80</b>	<b>118</b>	<b>33</b>	<b>125</b>	<b>56</b>	<b>112</b>	<b>44</b>	<b>109</b>	<b>32</b>

The *White Paper on Education and Training* in South Africa (Department of Education, 1995:50,89) states that girls and young women exhibit significantly narrower subject and career choices than boys. If girls have been systematically discouraged from selecting subject combinations that emphasise mathematics and science, then achieving equitable education requires that new ways be found to encourage more girls to select those subjects. The spread of women across different learning areas is uneven, with females clustered in the humanities and under-represented in science, engineering and technology (Department of Education, 2001:34). Institutional plans place far less emphasis on gender equity than on race equity. There are few, if any, strategies or interventions in place to address issues of gender equity (Department of Education, 2001:34).

The above-mentioned *White Paper* (Department of Education, 1995:29) mentions a shortage of students in particularly Science, Mathematics and Technology. It also states that special criteria will be needed to prepare students for those subjects in short supply. The National Plan for Higher Education in South Africa (Department of Education, 2001:15) also emphasises an endemic shortage of high-level professional and managerial skills in South Africa, especially in IT, engineering, technological and technical occupations. It also points out that these are the fields in which future demand is likely to be the greatest. The government is particularly keen to increase enrolments in the broad field of information and communications technology, which the Cabinet has identified as a key focus for skills development (Department of Education, 2001:26).

According to Minister Naledi Pandor (2006), Science, Mathematics and Information Communication Technologies play a vital role in social and economic development. Developing countries need to enhance their human and institutional capacity in Mathematics, Science and Technology if they are to succeed in their developmental goals. She further stated that South African learners' performance in the TIMMS (Trends in International Mathematics and Science Study) in 2003, 1999 and 1995 confirmed that South Africa's most important educational priority was to expand mathematical and scientific capacity in schools.

In South Africa, provision is made for learners to prepare for a career in IT by offering them the subject IT in the Further Education and Training band of the National Qualifications Framework. One of the four learning outcomes of IT focuses on the design and development of appropriate computer-based solutions to specific problems using programming (Department of Education, 2003:13). The weight allocated to this learning outcome is 60% (Department of Education, 2008b:7), which means that learners should be busy with programming for more than half of the time in the IT class.

From the above it is clear that an intervention is needed that will address the shortage of students in IT in general, but more specifically for girls. One possibility of attracting more girls to IT is to make them understand the importance of the subject and to make the subject more enjoyable for them at school level.

Williams and Upchurch (2001:327) suggest that pair programming, whereby two programmers work at one computer on the same programming task, shows several promising properties for educational purposes. Pair programming seems to have a positive effect in general on female computer science students at universities (Ho *et al.*, 2004; Werner *et al.*, 2004b:1; Werner *et al.*, 2005b), specifically in terms of enjoyment (McDowell *et al.*, 2003b:606; Werner *et al.*, 2004c; Ho *et al.*, 2004), and on their view of the importance or usefulness of the subject and a career in IT (McDowell *et al.*, 2003b:607; Werner *et al.*, 2004b:1; Werner *et al.*, 2004c).

Although pair programming seems advantageous to education in general and might lead to a retention in the number of female university students enrolled in IT courses, the question still remains whether pair programming could have an effect on girls' enjoyment and their view of the importance of programming, the subject IT and a career in IT, and whether it could possibly result in an improvement in their enrolment and the retention rate in IT at school level.

## 1.2 CLARIFICATION OF TERMINOLOGY

For the purpose of this study, it is necessary to explain and clarify certain key terms:

- According to the World Book Dictionary (WBD) (2003a:702), *enjoyment* refers to the pleasure that one gets from something; to have or use with joy; be happy with; take pleasure in. For the purpose of this study, *enjoyment* will refer to the pleasure or joy derived from programming and taking the subject IT.
- According to the WBD (2003b:1061), *importance* refers to the fact or quality of being important; meaning much; worth noticing or considering; having value or significance. For the purpose of this study, *importance* will refer to the perceived value or significance of programming, the subject IT and a career in IT.
- *Secondary school* refers to the training of Grade 8 to Grade 12 learners (13–17 years)
- *Female students* refers to women at a tertiary level, while *girls/learners* in this study refers to secondary school learners.
- *Information Technology* (IT) refers to one of the subjects that can be chosen from Grade 10 to Grade 12 in South African schools, focusing primarily on programming skills.
- *Pair programming* refers to two programmers working together, side-by-side on one programming task (This definition will be expanded in the literature review below).

### 1.3 LITERATURE REVIEW

There are many reasons for the shortage of girls in the IT class.

In an extensive study regarding attitude towards computers, Christensen *et al.* (2005:23) found that girls and boys in Grade 1 began with little or no difference in their attitude towards computers. The study further revealed that girls in Grades 4 and 5 were actually more positive than boys in their enjoyment of a computer. Starting at approximately Grade 6, girls' perception of computers became less positive than that of boys and by Grade 8 their perception of the subject was considerably less positive than that of the boys. However, there was a trend towards an increase in the girls' enjoyment of computers so that, by the end of high school, the same perception levels with girls and boys were revealed again (Christensen *et al.*, 2005:23). In South Africa, IT can be chosen as a subject at the end of Grade 9. If one took the above into account, it will have a significant impact on girls' choice of IT as a subject. In addition, the research of Seymour *et al.* (2005:103) in South Africa showed that learners did not know what the subject IT entails.

According to McKinney *et al.* (2008:84), the IT industry faces more of an *input problem* than a *throughput problem*. In other words, the under-representation problem is driven by fewer women entering IT and not by large numbers of women leaving, which suggests that greater attention should be paid to *supply-side* issues such as young girls identifying the importance of computer careers, early female identification with IT, and attention to learning styles in computer education.

Several solutions and strategies have been suggested to solve the problem of very few women in the computer science field. The AAUW (2000:10) recommends that the human, social and cultural dimensions and applications of computers (which women will enjoy), rather than the technical advances, the speed of the machines, or the entrepreneurial culture surrounding computers (which women do not enjoy) should be highlighted to reduce the continued under-representation of women in IT. Henson (2002) states that computing is perceived by females as non-social and that it is seen as a solitary occupation with little or no human interaction. Girls have reservations about the culture of computers – according to a report of the AAUW (2000:ix), the girls' concerns regarding the culture of computers must focus educators' attention on the changing of software, the way computer science is taught and the aims which they have with the use of computer technology. This could make IT more enjoyable for girls and enhance their view of the importance of the subject and a career in IT. Girls do not like the competitive and anti-social environment of computers, but rather like the collaboration, completion and non-



aggressive games associated with it (Chou & Tsai, 2007:812). Carlson (2006:235) concludes by saying: *Computer-science programs are stacked against women and the way they learn.*

There is a great need for strategies to foster females' enjoyment of IT and to enhance their interest. Pair programming appears to be one such approach (McDowell *et al.*, 2003b:607). *This approach touches on many issues that women face in the IT field, such as the issues of socialization, images, and confidence* (Balcita *et al.*, 2002:34).

An emerging software development methodology, Extreme Programming (XP) (Beck, 2000), has popularised pair programming. With this style of programming, *two* programmers work side-by-side at *one* computer, continuously collaborating on the same design, algorithm, code, or test. One of the pair, called the *driver*, types at the computer or writes down a design (Williams *et al.*, 2002:197). The other partner, called the *navigator*, continuously and actively examines the work of the driver – watching for errors, thinking of alternatives, looking up resources, and considering strategic implications of the work at hand. The navigator identifies tactical and strategic deficiencies in the work. The driver and navigator can brainstorm at any time. Periodically, it is also very important to switch roles between the driver and the navigator (Williams & Kessler, 2000a:111).

In industry, programmers collaborate for the greatest part of their workday. Traditional introductory programming courses generally require that learners work individually on their programming assignments. This is because working with another learner on a homework programming assignment is deemed to constitute cheating and is therefore not tolerated. According to Werner *et al.* (2005b), one of the reasons why female students do not enjoy programming is that this traditional pedagogical approach teaches learners that software development is an individual activity, potentially giving learners the mistaken impression that software engineering is an isolating and lonely career. Werner *et al.* (2005a) also state that many IT classes initially had female students who were interested in computer science, but became discouraged by its focus on individual, socially isolating work.

A number of research studies in tertiary education have revealed the enjoyment potential of pair programming:

- Students who program in pairs, enjoy programming more than solo programmers; they are happier and less frustrated (McDowell *et al.*, 2003b:607; Williams & Upchurch, 2001:327; Cockburn & Williams, 2000:2).

- Students who pair are more confident in their work (McDowell *et al.*, 2003b:607; Bishop-Clark *et al.*, 2006:213). Thomas *et al.* (2003:367) found that students with less self-confidence seem to enjoy pair programming most.
- It appears that because of pair programming, students that might otherwise have dropped the course, complete the course, and subsequently pass it. It also contributes to an enhanced view of the importance of the subject, and this leads to greater persistence in computer science related subjects and encourages students to pursue computer science careers (McDowell *et al.*, 2003b:607; McDowell *et al.*, 2002:41; Nagappan *et al.*, 2003).

Pair programming has enjoyment potential for students who take IT as a subject, but will it specifically address the unique perceptions and problems of girls in terms of their enjoyment and interest in programming, the subject IT and a career in IT? Werner *et al.* (2005b) paints the following picture: *A nerdy looking guy sits alone working at a computer late at night. Is this a portrait of your typical computer science student? Or instead, does your typical student look like one of a pair of students working together at one computer laughing, talking, pointing to the monitor, looking at each other, and having fun? This later picture is possible if you use pair programming.* The quote already points to the solution of a number of the typical problems encountered by girls in IT. The following advantages of pair programming for females were found by researchers in a tertiary education context:

- Female students working in pairs enjoy the programming process (Werner *et al.*, 2004c).
- Typically, female computer science students are less confident in their abilities than male students, even when their actual levels of competence are the same. This lack of confidence leads female computer science students to doubt their capabilities, question whether they belong, and frequently leads them to select other majors. The gender gap in confidence is significantly reduced when the students program in pairs (Margolis & Fisher, 2002:81; Werner *et al.*, 2004c).
- Female students also are less likely than males to persist in computing-related majors. Pair programming increases the retention rate in computing-related majors for all students; the gender gap in retention rates is reduced when students apply pair programming (McDowell *et al.*, 2003b:607; Werner *et al.*, 2005a).
- The collaborative nature of pair programming teaches female students that programming is not the competitive, socially isolating activity that they imagined (Werner *et al.*, 2004b:7).
- Ho *et al.* (2004) found that pair programming helps female students work more efficiently in programming tasks.

- Female students working in pairs achieve significantly higher grades than those working alone (Werner *et al.*, 2004c).
- Female students observed that they were more productive when working collaboratively, taking less time and producing a higher quality product. With more productivity, these women experienced more confidence and consequently more interest in IT careers (Berenson *et al.*, 2004:12).

From the above-mentioned advantages it is clear that pair programming can be beneficial to female students and female programmers because it might address factors that potentially limit their participation in the subject IT. No research on pair programming and secondary school girls' enjoyment of the subject IT, and of their view of the importance of programming, the subject IT and a career in IT could however be found. Research on secondary school girls' experiences could provide valuable information to attract more female students to IT on tertiary level.

#### 1.4 RESEARCH QUESTIONS

The research questions of this study are:

- How does pair programming shape secondary school girls' experience with regard to:
  - their enjoyment of programming and the subject IT
  - their view of the importance of programming, the subject IT and a career in IT?

**Table 1.2** Research subquestions

1. How does pair programming shape secondary school girls' experience with regard to their <b>enjoyment of programming</b> ?
2. How does pair programming shape secondary school girls' experience with regard to their <b>enjoyment of the subject IT</b> ?
3. How does pair programming shape secondary school girls' experience with regard to their view of <b>the importance of programming and the subject IT</b> ?
4. How does pair programming shape secondary school girls' experience with regard to their view of <b>the importance of a career in IT</b> ?

## **1.5 AIM OF THE RESEARCH**

The aim of the research is to understand how pair programming shapes secondary school girls' experiences. This aim is operationalised as follows (to form pertinent objectives that have to be reached):

- To understand how pair programming shapes secondary school girls' experiences with regard to:
  - their enjoyment of programming and the subject IT
  - their view of the importance of programming, the subject IT and a career in IT.

## **1.6 RESEARCH DESIGN AND METHODOLOGY**

### **1.6.1 Literature study**

A literature study of both primary and secondary sources was undertaken. Applicable books, magazine articles and documents were gathered and evaluated. An internet search was undertaken using the search engines Google and Google scholar. The Ebscohost database was used, including Eric, Academic Search Premier, Masterfile Premier, Computers and Applied Sciences Complete, and Teacher Centre, with the following descriptors: Pair programming, Gender, Female, Computer, Enjoyment, Importance, Persistence, Retention, Interest, Information technology, Computer studies, Computer Science Education, Collaborative learning, as well as a combination of these terms.

### **1.6.2 Empirical investigation**

#### **1.6.2.1 Research design**

When choosing a research design, a researcher must identify the theoretical framework that forms the scaffolding of the study (Merriam, 1998:2). A qualitative approach departing from an interpretivistic theoretical framework was used to conduct the research. With an interpretivistic theoretical framework, education is considered to be a process and school is a lived experience. Understanding the meaning of the process or experience constitutes the knowledge to be gained from an inductive mode of inquiry (Merriam, 1998:4).

A basic qualitative design was used (Merriam, 1998:11) aiming at assisting the researcher to discover and gain understanding regarding the lived experiences from the perspective of girls experiencing programming and pair programming. According to McMillan (2000:9,14), qualitative research stresses a phenomenological model in which multiple realities are rooted in the subjects' perceptions. The purpose of such research is to provide rich narrative descriptions of phenomena that enhance understanding. A focus on understanding and meaning is based on verbal narratives and observations rather than numbers.

Qualitative studies have limitations too. The lack of generalisability of results is a well-known drawback of all qualitative research, but the goal of the study was to understand the particular phenomenon in depth rather than to know what is generally true. Validity and reliability are also issues that needed to be addressed since the researcher is the primary instrument of data collection and analysis and the subjectivity of the researcher could lead to the problem of bias (Merriam, 1998:42; Mouton, 2001:150). Section 1.6.2.6 contains a discussion of the measures that were taken to minimise the effects of these problems.

#### **1.6.2.2 Study participants**

The six participants were all the Grade 11 girls in the IT class of a school in the North-West Province. The school is multi-cultural and has pupils from an average to high socio-economic background. These Grade 11 female learners of the school were used because they had gained experience in solo-programming in their Grade 10 year.

#### **1.6.2.3 Data collection methods**

Semi-structured interviews were conducted with the aid of interview schedules to determine the girls' perceptions of and attitudes towards programming, the importance they attach to the subject IT and to a career in IT. According to Merriam (1998:72), interviewing is necessary when we cannot observe behaviour, feelings, or how people interpret the world around them.

#### **1.6.2.4 Data collection**

At the first meeting, interviews were conducted to determine the girls' level of enjoyment of programming and the subject IT and their views on the importance of programming, of the subject IT and of a career in IT. After the first interview, all the Grade 11 IT learners were trained by their class teacher in the use of pair programming and they were informed that pair programming was to be used for all programming assignments during the term.

After 3 months, once the learners had completed several paired assignments, the girls were interviewed again by the researcher. The purpose of these interviews was to qualitatively determine if there was a change in each girl's enjoyment of and in their views regarding the importance of programming, of the subject IT and of a career in IT. The girls' experiences during pair programming were also determined. All interviews were recorded.

#### **1.6.2.5 Data analysis**

The analysis of the data of the interviews progressed as follows:

- Transcribing the interviews
- Using the computer program ATLAS.ti 5.2, each respondent's interviews were assigned to a single hermeneutic unit resulting in six hermeneutic units.

Each hermeneutic unit was then analysed as follows:

- The relevant information was broken into a number of segments, coding the themes as they emerged and linking the segments to the coded themes.
- Grouping the codes into code families and drawing networks that reflect the meaning of the experience (see Appendix A).
- Using the networks to develop an overall description of the phenomenon as girls typically experienced it (Leedy & Ormrod, 2005:140).

#### **1.6.2.6 Trustworthiness**

Although validity and reliability are issues that are hard to control and measure in qualitative research, the following was done to promote the trustworthiness of the study:

- Multiple data collection methods through a literature study and interviews with the participants were used.
- Peer review: Discussions with other researchers familiar with pair programming regarding the process of the study and tentative interpretations.
- Engagement: Spent adequate time collecting data.
- Rich, thick description: Provided enough description for readers to be able to determine the context of the study.
- Reflect: Critically self-reflected on the researcher's assumptions, experiences and biases.

#### **1.6.2.7 Ethical aspects of the research**

Permission to conduct the study in a secondary school in the North-West Province had been obtained from the superintendent-general of the North-West Department of Education (see Appendix B and Appendix C).

Permission from the principal of the target school in the North-West Province was obtained (see Appendix D). The teacher of the IT class in the relevant school was informed about the use of pair programming in education and his permission to participate was obtained.

The learners and their parents signed an informed consent form (see Appendix G) and they were informed that confidentiality and anonymity would be ensured. Participants were informed

that they could withdraw from the project without penalisation if they found any aspect of the process uncomfortable.

The study adhered to the ethical requirements of the North-West University (see Appendix E).

## **1.7 STRUCTURE OF DISSERTATION**

Chapter 2 entails a literature study of girls and the computer environment with attention to factors that have an effect on girls' attitudes towards computers and involvement in the computer environment.

Chapter 3 presents a literature study of pair programming in the IT class with attention to the possible advantages of pair programming.

The empirical study and its results are described in Chapter 4, together with a discussion and interpretation of these results.

In Chapter 5, a summary of the study is given, the findings and conclusions made are discussed and recommendations for further research are made.

## **1.8 LIMITATIONS OF THE STUDY**

The very reason for this study (the shortage of girls in IT) had limited the researcher in the sense that the number of girls taking IT as a subject in the North-West province is limited (often only one per class – see Table 1.1) and to convince the teacher to implement pair programming and then train the teacher in pair programming for the sake of only one or two participants was a major stumbling block.

## **1.9 CONTRIBUTION OF THE STUDY**

No research on pair programming and the way it shapes girls' experiences in secondary schools could be found. In addition, no research on the effect of pair programming on females in South Africa could be found. Furthermore, an in-depth qualitative investigation about this phenomenon has not been found. The study can contribute to a deeper understanding of girls' experiences with programming and the subject IT, which subsequently might lead to increased enrolment of females at tertiary level.

## Chapter 2

### Girls and the computer environment

#### 2.1 INTRODUCTION

Women's participation in university-level computing degrees is typically low worldwide, and as a result participation by women in computing professions is equally low (Galpin, 2002:99). Statistics for South Africa since 1992 show that the proportion of women graduating with Computer Science degrees were the highest (32.7%) in 2000 and the lowest (22.8%) in 1995 (Huhlwane, 2006). In the USA, who is at the forefront of IT, the picture is even bleaker with less than 30% of the total degrees in computer science awarded to women (Boyd, 2000:90).

On secondary level, the situation is not different. In the UK, the National GCE A-level results of June 2008 showed that of the 5068 learners that had written the Computing exam, only 480 (9.5%) were girls (Guardian, 2008). Table 1.1 shows the same phenomenon of girls being a minority in IT classes in the North West Province in South Africa.

Henson (2002), Frieze (2007:56) and Shashaani (1997:42) identified a number of reasons why girls and women are pushed away from computers:

- They do not find computer work enjoyable.
- They have less confidence in their own abilities and skills.
- They have less opportunities for friends and mentors in the computer environment.
- They are de-motivated from an early age.
- Computers are seen as anti-social.
- There is a shortage of female role models.
- Games and classes are aimed at boys and men.
- Advertisements and the media proclaim that computers are for men.
- The balance between one's social and work life is more important to girls.
- Computer classes do not address their interests of creativity, relevance and collaboration.

In view of the growing role of technology in the world at the beginning of the 21st century – in education, communications, occupations, and entertainment, and as a tool for solving the world's problems – women's low and decreasing representation in IT is a major concern. Women bring different ways of thinking to problem solving and teamwork and as a result



new areas can be discovered and thought about that might be missed by a pure male team (Mendels, 2000).

## **2.2 GIRLS' PREFERENCES REGARDING THE COMPUTER ENVIRONMENT**

Girls are more interested in the social and cultural applications of the computer, while boys are more attracted to the nuts and bolts of computers. Boys also get more hands-on experience of the computer than girls and girls have a smaller technical appreciation of computers (Carlson, 2006; Clarke & Finnie, 1998:26). Girls tend to see computers as equipment, where boys view it as a toy, and in the process, boys tend to substitute social skills with computer skills. Girls simply are not prepared to make that sacrifice: *We can, but I don't want to* (AAUW, 2000:7–9).

Girls do not like the competitive and anti-social environment of computers but rather like collaboration and relevance to the real world (Chou & Tsai, 2007:812; Frieze, 2007:56). Girls enjoy using existing systems rather than developing new ones and they are attracted when they recognise computing as a form of communication, a means of creative self-expression, or as a path to a helping occupation (Tillberg *et al.*, 2005:137; Shotick & Stephens, 2006:269).

## **2.3 THE ATTITUDES OF GIRLS TOWARDS COMPUTERS**

Woodrow (1991:165) describes three major factors or constructs of attitudes towards computers:

- enjoyment or liking of computers;
- computers' importance or perceived usefulness; and
- anxiety about computers, i.e. confidence in using computers.

In a study determining the reliability of 14 computer attitude instruments, the three above-mentioned constructs proved to be the constructs most often used. Some other constructs emerged to a lesser extent: efficacy, awareness, societal impact, attitude towards e-mail, and computer seclusion (Christensen & Knezek, 2000:328). Computer attitudes not only play an influential role in the extent to which learners accept the computer as a learning tool, but also influence future behaviour towards the computer, such as using it for further studies and vocational purposes (Teo, 2007:127).

In the next section, attention will be given to the three major constructs of girls' attitudes towards computers, i.e. enjoyment, importance and anxiety.

### 2.3.1 Enjoyment

Various studies in different countries found that girls have more negative feelings about the computer than boys – girls tend to be less fond of computers and do not enjoy working with computers as much as boys do (Sanders, 2005a:10; Comber *et al.*, 1997:130–132; Reinen & Plomp, 1997:67; Shashaani, 1997:42).

Colley and Comber (2003:163) compared findings of an early 1990's study with findings of a 2003 study to determine if the increased use of computers in the school curriculum had an effect on computer attitudes. They found that boys remained more self-confident and liked computers more and they still found less of a liking among girls, especially in older girls. The increased use of computers in the school curriculum therefore doesn't seem to positively shape girls' experience of computers. Bovée *et al.* (2007:1771) as well as Dryburgh (2000:197) found that girls in a single-sex school or grouping enjoyed computers more than girls from a co-educational school setting or mixed grouping. Girls seem to experience computers more positively when they are in the company of their female peers.

In an extensive study regarding enjoyment of computers, Christensen *et al.* (2005:23) found that girls and boys in Grade 1 begin with little or no difference in their attitude. The study further revealed that girls in Grade 4 and Grade 5 are actually more positive than boys in their enjoyment of a computer. Starting at approximately Grade 6, girls' perception of computers becomes less positive than that of boys and by Grade 8, their enjoyment is considerably less than of boys. However, there is a trend to then have an increase in the girls' enjoyment of computers so that, by the end of secondary school, girls and boys tend to have the same levels of enjoyment again. This result echoed the results of the study by Colley and Comber (2003:163) mentioned in the previous paragraph and since IT as a subject is chosen at the end of Grade 9 in South Africa, the above findings might explain to a large extent the lack of girls in the IT class.

Researchers attempt to come up with solutions for increasing girls' enjoyment of computers. Teo (2007:134) suggests that the challenge is for educators to transfer the enjoyment of using computers for personal use and entertainment to educational purposes. Knezek and Christensen (1996) found that junior secondary school learners receiving computer literacy training through thematic teacher teaming activities enjoyed computers more and they more often view computers as important compared to their counterparts in traditional computer literacy classes. If we bear the above results of Christensen *et al.* (2005:23) in mind, this not only increases their enjoyment, but it also speaks to the most vulnerable, namely the junior secondary school students.

### 2.3.2 Importance, interest and persistence

Like many things in life, girls' view of the importance or usefulness of computers, their interest in computers, the subject and a career in IT and their persistence in the subject are all interrelated and cannot be dealt with as single independent identities.

In terms of the importance of computers, Koohang (1989:214) found that male university students rated computers as more useful than did female students, but Reinen and Plomp (1997:67) found that of the 10 countries they studied, only in Austria, the Netherlands and Latvia, were male students more convinced of the relevance of learning to work with computers. Shashaani (1997:42) found no significant differences between men and women in respect of the perceived usefulness of computers. Both genders were aware that a knowledge of computers is important for obtaining a job, saves time and work, and is useful for data processing and problem solving. Bovée *et al.* (2007:1770) also found in their study of learners aged between 10 and 16 in South Africa that girls as well as boys regarded computers as important and useful for daily life. Most learners indicated that it is important to learn how to use a computer, that a computer is useful, and that a computer gives one the opportunity to learn new things. In addition, the learners were convinced that possessing computer skills contributes to getting a good job in the future.

Shashaani (1997:48) found that the female students' negative attitudes were unrelated to their performance and she speculates that females' perception of the usefulness and importance of computers in their education and career, and their strong belief that girls have as much ability as boys in learning to use computers, encourage them to put more effort into their studies and obtain better marks.

In terms of parents' influence on girls' perception of the importance of computers, Shashaani (1994:443) revealed that parents express the difficulties or drawbacks of studying a subject to their daughters but stress the importance of that same subject to their sons. Her study illustrated that only 22% of the parents of female students recommended a computer science course as an important subject for their daughters to take, whereas 67% of the parents perceived computer science as an important subject for their sons. Considering the significant effect of parental encouragement, one might expect girls to develop a conception that IT is not important for girls. Reinen and Plomp (1997:73), on the other hand, found that females agree more on the relevance of computer use when they use the computer at school. If one can assume that relevance of working with a computer is a prerequisite for optimal computer use, the conclusion is that the school environment and the possibilities of using computers in the school are especially important for female learners.

Although it seems from the above that girls do view computers as important and useful, they are not necessarily interested. Girls in South Africa showed very limited interest in technology-related careers (Seymour *et al.*, 2005:104; Bovée *et al.*, 2007:1770). Margolis *et al.* (1999:13) reveal that most boys describe an early and persistent magnetic attraction between themselves and computers, while girls much more frequently link their computer science interest to a larger societal framework like medicine or the arts. Margolis *et al.* (2000a:121) states: For girls,

*trying to find their place in a culture that challenges whether they are “really into it” and a curriculum that assumes their learning will occur in the same sequence and timing as their male peers, too many conclude that they “just aren’t interested”.*

The AAUW (2000:7) confirms the notion with what they call the “*we can, but I don’t want to*” syndrome.

Not only are girls not interested in computers, but those who might have started with an interest, get lost along the way. The pipeline shrinkage problem for women in computer science is a well-known and documented phenomenon where the ratio of women to men involved in computing shrinks dramatically from taking computer courses in secondary school on through university and into IT careers (Gürer & Camp, 2002:121). Vegso (2005) found that the proportion of women in the U.S. considering an IT major *has fallen to levels unseen since the early 1970s*. In South Africa, a report published by the Department of Science and Technology (2004:28) shows that there is a drop from the proportion of undergraduates who are female to the proportion of professors who are female in the IT field. Even from Table 1.1 it is clear that the number of girls in these schools who had started the subject IT in 2007, were 80 and that number dropped to 44 girls in 2008.

Researchers suggest many causes for the *leaky pipeline* (Galpin & Sanders, 1993; Camp, 1997; Gürer & Camp, 1998; Dryburgh, 2000; Gürer & Camp, 2002):

- the prevalence in many students’ minds of negative stereotypes about computer workers (Durndell *et al.*, 1995:226; Herbert, 2000; Kekelis *et al.*, 2005a:100)
- the lack of accurate information about IT careers (Chan *et al.*, 2000; Goode *et al.*, 2006:112; Kekelis *et al.*, 2005a:101). The research of Seymour *et al.* (2005:103) in South Africa showed that secondary school learners do not know what a degree in IT entails, but they are convinced that remuneration and occupational benefits are positive factors in the pursuing of this career direction. According to Herbert (2000),

perceptions exist that computer science is a male domain and that limited career possibilities exist for women in IT.

- lack of career guidance and support from family (Kekelis *et al.*, 2005a:102)
- the perception in student' minds of IT being non-social and solitary (Henson, 2002). Many secondary school girls believe that computer careers involve programming all day, every day (Dryburgh, 2000:191). Comparing perspectives of women working in computing with those of students, Teague and Clarke (1991:377) found that the reality of computer work is that it is much more people-oriented, diverse, and interesting than was perceived by students.

Goode *et al.* (2006:112) found that the scientific heart of computer science is *lost in translation* at the secondary school level, and as a result the field continues to lose the participation and interest of a broad layer of students, especially females.

### **2.3.3 Anxiety and confidence**

Anxiety and confidence are two interdependent constructs – if you are anxious about the use of something like a computer, you definitely won't be confident in using it and vice versa.

#### **2.3.3.1 Anxiety**

Computer anxiety is an affective component and can be described as the fear of computers or the tendency of a person to be uneasy, apprehensive and phobic towards the use of computers (Bovée *et al.*, 2007:1767).

Most studies have found computer anxiety higher in females than in males (Baloğlu & Çevik, 2008:2646). King *et al.* (2002:80) revealed in their study that when comparing gender and grade simultaneously, the males were the higher anxiety group in Grade 11 and the females were the higher anxiety group in Grade 7. These results again indicate that when learners have to make subject choices in Grade 9 in South Africa, girls are very vulnerable, also in terms of anxiety.

Whitley (1996:275) determined that prior experience did not mediate gender differences in anxiety, but that anxiety mediated gender differences in computer behaviour. Computer anxiety levels differ according to ownership and frequency of use. Nelson *et al.* (1991:185) found that females who dropped out of computer courses had higher computer anxiety than those who persisted, but that males who dropped out had lower anxiety than those who stayed. Anxiety therefore has an influence on girls' persistence in the subject IT.

Beckers *et al.* (2007:2859) suggested two mutually exclusive conceptions of the nature of computer anxiety: computer anxiety predominantly as a variable mood state and computer anxiety seen as a stable trait. They found in their study that computer anxiety is more strongly related to trait anxiety than to state anxiety. It is further concluded that since computer anxiety has a base in trait anxiety, it will negatively influence the success of treatments that are solely focused on teaching computer users the intricacies of various applications.

#### **2.3.3.2 Confidence**

Females' confidence levels in their ability to learn about and use computers are significantly lower than that of males even when females are more successful than the males in the class (Gürer & Camp, 1998; Shashaani, 1994:440). In a three-year study by the university of Hong Kong where first year students had to evaluate themselves at the beginning of the year on a 5-point scale which varies from expert to beginner, 9.2% of the male students had described themselves as experts in comparison with only 0.7% of the female students. Only 3.8% of the male students described themselves as beginners versus 5.7% of the females (Lee, 2003:492). Gürer and Camp (1998) however, found that in general, women students have confidence in their ability to master the projects presented in class but they do not feel confident whether or not they will be able to successfully apply their knowledge to a real-world problem. Since we saw in 2.2 that solving real-world problems is important to girls, and since Powell (2008) found that girls with lower confidence are likelier to drop out of computer programs, teachers will have to work on improving girls' confidence in applying the knowledge and skills they gained in class.

Parents' beliefs that the computer is more appropriate for males than for females inversely affect their daughters' interest in computers and reduce their daughters' confidence in working with computers (Shashaani, 1994:440). Parental encouragement correlates with confidence for both girls and boys, but boys receive more of it (Shashaani, 1994:438; Shashaani & Khalili, 2001:372).

Computer use has a positive effect on perceived computer self-confidence as well as on computer-related attitudes (Levine & Donitsa-Schmidt, 1998:139). The more time spent on a computer becoming familiar with the program or games, the more confident the users become with using the technology (Klawe, 2002:16).

Margolis and her colleagues explored computer interest in several studies, finally concluding that in the *nexus of confidence and interest* a female's loss of confidence in her computer

abilities precedes a drop in her interest in computers (Margolis *et al.*, 2000b; Margolis & Fisher, 2000).

The section is concluded by a quote from Margolis *et al.* (2000a:117) which beautifully wraps up the whole issue of girls' confidence:

*The computer science culture assumes that men will succeed. Success is linked to a stereotype based on a common male pattern of desires, interests and attachments to computing. Hence, it bolsters men's confidence and sense of belonging. This same culture does not assume (often accurately) that women conform; hence they enjoy no default expectation of success, and their interests and attachments to computing may be regarded as deviant from the norm, and less serious than those of the male students. This, combined with a vast array of gender socialization factors, chips away at women's sense of confidence and belonging in the field.*

## **2.4 GENERAL FACTORS THAT HAVE AN EFFECT ON GIRLS' ATTITUDES TOWARDS COMPUTERS AND THEIR INVOLVEMENT IN THE COMPUTER ENVIRONMENT**

In the next section, attention will be given to societal influences, age, experience and software as factors that have an effect on girls' attitudes towards computers and their involvement in the computer environment.

### **2.4.1 Societal influences**

The persistent gender gap in computer participation is produced by the influence of the social environment within which boys and girls are socialised. Most researchers involved in this area agree that the basis of gender differences in computer attitudes is essentially social and cultural and not related to inner ability (Shashaani, 1994:441). Galpin and Sanders (1993:4) found that the persistence rates for undergraduates and the pass rates for men and women were similar, which led them to conclude that the gender imbalances which are observed in IT classes are determined by factors outside of the course and subject and these factors are largely societal and environmental.

Societal influences recognised by researchers are parents, the media, the male culture of IT, race and ethnicity and socio-economic status. These five influences will be discussed in the following section.

#### **2.4.1.1 Parents**

Parents are one source of gender stereotypes with respect to computing. Shashaani (1994:442) found that parents' computer stereotypes in favour of males encouraged their sons' computer involvement and discouraged their daughters'. Chen (1986:265) found that more parents buy a computer with their son in mind, rather than their daughter and Henson (2002) also confirms the point with the fact that in America the family computer is rather kept in the son's room. Martineau (1998:13) claims that expectations and opinions of parents and society cause girls to be driven away from computer classes.

Parents are learners' number one source for choice of a career; girls tend to follow in their mothers' footsteps and usually it is not the mother involved in technology, but it is rather the father who is busy with the computer, or busy installing the sound system, or he even follows a career involving technology (Carlson, 2006).

#### **2.4.1.2 The media**

The media often proclaims that computers are a male domain. Knupfer examined computer advertisements, the Internet, television, magazines and movies and found rampant gender stereotypes about people in technical roles (Knupfer, 1998:55; Knupfer, 1997:36). In advertisements and movies, it is usually a man behind the computer and if a women is ever depicted as a programmer, much more time is spent on her curves than on a demonstration of her competency as programmer. Angelina Jolie in the movie "Hackers" is a case in point (Henson:2002).

According to Carlson (2006), the image of a brilliant but socially inept mumbler who could use a few tips on hairstyles and clothes, as so often portrayed by the media, is not good to improve girls' confidence in taking the subject. He continues by saying that a teenage girl who claims she wants to become a computer scientist, is often treated the same as a boy who says he wants to be a nurse. The stereotypical images of male and female roles related to technology are often perpetuated subconsciously in classroom practice and therefore teachers should be aware of their own discriminatory actions (Knupfer, 1997:31).

#### **2.4.1.3 The male culture of IT**

The culture of computing is male-dominated and in the literature, gender stereotyping is seen as an essential problem with negative effects on women (AAUW, 1998:10; Shotick & Stephens, 2006:270). There is a tendency in technology classes to direct girls away to training for clerical work, data-input (the post-1990's version of typing) and other occupations traditionally seen as women's work (AAUW, 1998:10). Society created an association



between computers and masculinity and Elkjaer (1992:25) points out that masculinity, not femininity, is the problem when boys retreat into the computer to avoid human interactions and when they consider themselves the hosts in the computer environment, with girls as guests (Shotick & Stephens, 2006:269).

Girls have negative notions of the computer culture and perceive computer enthusiasts as male, nerdy, very smart, social isolates who are competitive, exclusively focused on programming and content to sit in front of a computer for long hours (Klawe, 2001:67; Chou & Tsai, 2007:812; Dryburgh, 2000:192; Canada & Brusca, 1991:47, Kekelis *et al.*, 2005a:99). The typical stereotype of a computer scientist is therefore not an attractive image for women (Dryburgh, 2000:198).

The violent language of technology may be invisible to males, but can be a problem for females. Consider hard disc, hard drive, reboot, cold boot, hits, permanent fatal error, and so forth (Sanders, 2005a). Recreational or even educational software for children often include title words such as "attack" or "war". A shocking example is a system, created by a group of men, named F.A.L.L.U.S., which is actually advertised as *a firm favorite, an uprising, pulsating tower of strength that shows breath-taking expandability and has a tempting range of speeds that will more than satisfy your needs* (Gürer & Camp, 1998).

The difference in sex-stereotypical attitudes to computers becomes greater with more computer experience, which suggests that by the secondary stage, girls not only have to deal with being a minority in computing classes, but they also face a majority of learners who believe men do computing better than women (Dryburgh, 2000:198). Canada and Brusca (1991:47) agree that it is not necessarily computers and technology *per se* that females avoid, but rather the competitive, male environment that surrounds the field.

#### **2.4.1.4 Race and ethnicity**

There is evidence of different cultural influences on women in the IT field that are related to differences in nationality and ethnicity (Trauth *et al.*, 2008:16). Students of colour are afforded fewer computer opportunities than white students and females of colour are therefore subject to the double discriminatory burden of femininity and minority status with respect to computing (Goode *et al.*, 2006:112; Garcia & Giles, 2000).

In a study in South Africa, Seymour *et al.* (2005:104) found that 23% of the study sample of 497 Grade 12 learners chose IT as a possible study choice. A total of 43% of those interested in IT were females, but it transpired that there were no white girls in the sample whom had expressed an interest in IT. The only inclination for females to study IT therefore

came from black or coloured scholars. Seymour *et al.* (2005:104) speculate that this factor might be attributed to the favourable job perceptions of previously-disadvantaged scholars and the fact that they probably do not know that they might not meet the necessary entrance requirements.

#### **2.4.1.5 Socio-economic status**

Shashaani (1994:433) found in her study that SES (socio-economic status), including the parents' occupations and education, exerted a significant effect on students' attitudes toward computers. In general, gender-differential attitudes were more pronounced in the lower socio-economic group and SES had a stronger effect on girls than boys. The effect of parental education, particularly the mother's, on students' attitudes toward computers was higher than that of parental occupation (Shashaani & Khalili, 2001:372). Lower socio-economic class learners have less access to computer technology, come from secondary schools with limited resources and ill-prepared teachers, and often belong to an under-represented minority (Varma, 2002).

Bovée *et al.* (2007:1772) reveal in their study in South Africa that although no significant gender differences were found in learners' computer attitudes, those from the upper and middle SES schools had a more positive computer attitude and less stereotypical views about women in computing than the learners from the lower SES schools. It can therefore be derived that the daughter of a lower SES mother will most probably have a negative attitude towards computers.

The societal influences discussed are now succeeded by having a look at age and experience in computer use.

#### **2.4.2 Age**

Research studies have found that gender differences in attitudes and behaviour are relatively small at younger ages but increase as students become older (Christensen *et al.*, 2005:23; Collis & Williams, 1987:26; Durndell *et al.*, 1995:219). Computer enjoyment and interest decrease with age for both girls and boys, but more strongly for girls (Christensen *et al.*, 2005:23; Collis & Williams, 1987:17; Gürer & Camp, 2002:121). Whitley (1997:13), in a review of 82 studies, concluded that gender differences in attitudes toward computers result from socialisation processes: the longer children are in school, the greater the gender difference becomes. He noted, however, that such differences were smaller for university-level students and speculated that perhaps young women with more positive computer attitudes were more likely to go to university.

Subject choices are normally made in the mid-teens and as Durndell *et al.* (1995:226) point out, it is these choices that have the greatest impact on career choice. In South Africa, IT as a subject is chosen at the end of Grade 9 and if one takes the above into account, it will have a significant impact on the choice of IT as a subject by girls and consequently on a career in IT.

#### **2.4.3 Experience**

Boys have an edge in home computer use, school computer use, computer course-taking, games, and in free-time exploratory use. Of these, games and free-time exploratory use are most frequently cited as the primary causes of boys' greater computer experience (Fisher *et al.*, 1997:107; Goode *et al.*, 2006:93; AAUW, 1998:4; AAUW, 2000:52). At home, girls' access is problematic due to competition with male family members (Gunn, 2003:22). According to Statistics South Africa, boys between the ages of 15 and 19 spend twice as much time using computers than girls (Chobokoane & Budlender, 2002).

Once the population of women in the USA who used the Internet exceeded the number of men who used the Internet, the perception existed that the gender digital gap had disappeared and equal experience had been reached, but this is not true because e-mailing and Web-surfing do not imply that the computer is used pro-actively (Gorski, 2002:23; Fox, 2000:10).

According to Tocci and Engelhard (1991:280), a person's attitude towards an object like a computer is influenced by direct experience with the object and through interaction with other similar objects. The initial encounter with the computer may be crucial to the formulation of computing attitudes, especially anxiety attitudes (McIlroy *et al.*, 2001:31). Galpin *et al.* (2003:18) found that learners with high usage of the computer are more likely to decide to take IT as a subject. Seymour *et al.* (2005:101) found in a study in South Africa that boys had significantly more access to computers and the Internet in the home environment than female learners and lack of computer access had a negative influence on the choice of IT as a study field, especially for the females. Governments, schools, teachers, and parents should all attempt to provide more access to computers which will result in increased exposure and experience, especially for girls.

#### **2.4.4 Games and other software**

Games and computer software in general are designed for men, by men (Goode *et al.*, 2006:93). The very existence of software "for girls" confirms that software is indeed for boys (Sanders, 2005a). Software developed for girls is even called "pink software" and is based

on common gender stereotypes: shopping, makeup, fashion and dating (Rubin *et al.*, 1997). Sexism in software occurs in characters, content, reward systems, and structure (Bhargava *et al.*, 2002:205). According to Chou and Tsai (2007:812), boys spend more time playing games and enjoy it more than girls and Kaiser (2000:3) believes that violent computer games cause girls to reject computers.

When teachers were asked to design software for girls, boys or students, they designed tool software for girls and game software featuring violence and competitiveness for boys as well as students – in other words when the gender was not mentioned, the designed software had the same features as those for the boys (Huff, 2002:112). Huff (2002:113) concludes that gender stereotypes can make their way into the design of software, and software that is based on those stereotypes can differentially affect boys' and girls' experience of interaction with the software. Gorriz and Medina (2000:48) state that for most girls their first introduction to the computer is by playing software games and through the creation of software games which will interest the girls, potentially more girls at a young age will be exposed to the computer world, which will increase the numbers right through the pipeline.

## **2.5 FACTORS IN THE CLASSROOM THAT HAVE AN EFFECT ON GIRLS' ATTITUDES TOWARDS COMPUTERS AND THEIR INVOLVEMENT IN THE COMPUTER ENVIRONMENT**

In the next section, attention will be given to peers, teachers, teaching strategies, the IT curriculum, programming, and academic achievement as factors that have an effect on girls' attitudes towards computers and their involvement in the computer environment.

### **2.5.1 The influence of peers on girls**

Gender stereotyping among learners exists, although they might not admit or realise it, which in turn influences their peers. The notions that males excel in mathematics, science, and technology, and that females excel in the arts, are two of many beliefs and cultural influences that are passed down through generations and children reflect and reinforce these attitudes through their peer interactions (Sanders, 1997). In studies by Durndell *et al.* (1995:226) and Shashaani (1997:42), girls as well as boys, believed males to be better at computing than females; just as consistently, boys were more likely to hold stronger stereotypes in this regard than girls.

Nass *et al.* (1997:874) did an interesting experiment with university students. When computers "spoke" about male- or female-stereotyping topics in low- (male) or high-pitched (female) voices, students of both sexes rated the "female" computer more knowledgeable

about feminine topics and the “male” computer more knowledgeable about male topics. Students of both sexes found evaluation from the “male” computer more credible. Nevertheless, students denied harbouring stereotypes about their peers or being influenced by the gender of the computer voices.

A closely related issue is critical mass. It is not the presence of boys at the computers that discourages girls’ participation, but rather the absence of the girls’ girlfriends (Sanders, 2005a). Cohoon (2001:112) found that a critical mass of other women correlated more strongly than any other factor with women’s retention in computer science majors in USA universities. One of the factors credited for raising the female presence in Carnegie Mellon University’s School of Computer Science, was the critical mass provided by increasing numbers of women (Blum, 2001:6). Blum and Frieze (2005) claim that when computer enrolment becomes more equalised by sex, the culture changes in ways that are positive for both boys and girls.

### **2.5.2 Teachers’ effect on girls**

Although unintentional, gender stereotyping exists among teachers. In Canada, teachers explained gender differences in computing with stereotypes, but denied that gender was a consideration in their explanations (Bryson & de Castell, 1998:560). However, in the survey of Wasburn and Miller (2005:76), female IT students reported their observation that professors did not treat male and female students equally. Female students’ retention in IT is positively related to their professors’ positive attitudes toward women students and negatively related to their professors’ belief that female students are not well-suited to take IT as a subject (Cohoon, 2001:112). A large sample of American secondary school learners of both sexes agreed that teachers, counsellors and parents all believed that computers were more appropriate for boys than girls (Shashaani, 1993:169). The preceding findings indicate that teachers and university lecturers alike should be mindful about their own attitude towards girls in the IT class because they might, through their own beliefs and actions, push girls away from the very subject which is near and dear to them.

Computer teachers, much like other teachers and managers, are not aware of data or broader gender issues that could illuminate the computer gender gap for them because when they took their education courses at university, gender was in all likelihood barely mentioned or not mentioned at all – education texts rarely mention it (Sanders, 2005b). Teachers are almost always unaware of the biased behaviours they exhibit through verbal interactions, eye contact, and body language, which means they cannot correct themselves (Sanders, 1997). It is the subtle, often unintentional, and individually trivial incidents of

gender bias that are cumulatively powerful and have the effect of discouraging female participation in technology (Sanders, 2005b).

Sanders (2005b) claims that teachers believe that because they are not intentionally discriminatory, the gender gap they see – if and when they see it – cannot be helped. Teachers wish it were otherwise and they often try to encourage girls to continue in computing. When their efforts do not bear much fruit, they tend to be resigned to the status quo.

### **2.5.3 The effect of teaching strategies on girls**

A considerable amount of research investigates the relationships between teaching strategies, gender and technology. Collaboration, and mentors and role-models are two important issues that will be addressed in the following section.

#### **2.5.3.1 Collaboration**

The second critical outcome of the National Curriculum Statement (Department of Education, 2003:2) requires that learners work effectively with others as members of a team, group, organisation and community. The Learning Programme Guidelines for IT (Department of Education, 2008a:7) also states that the purpose of IT is to develop multifaceted and multileveled systems through collaborative teamwork.

The approach to work and learn collaboratively seems to be especially acceptable to girls. Miller *et al.* (1996:32) found in their study that sixth- to twelfth-grade girls preferred software that required them to collaborate rather than compete. Girls more often want to work as equal members of a team of people, whereas boys are more inclined to elect to be the sole leader of a team (Bennett *et al.*, 2006:66). In a project which had sought to retain first-year female undergraduates by giving them access to collaborative laboratory work, it was found that the retention rate was nearly double that of a control group of students (AAUW:2004:10).

Beyer (2006:12) suggests that the negative stereotype of computer scientists as asocial loners working with machines needs to change, because she claims that IT classes that emphasise competition and discourage cooperation and teamwork, send the message that IT is a cut-throat profession where collaboration is not the norm. She continues by emphasising that all students in IT need to be educated about the realities of work in the field, that the typical workplace involves teamwork and interactions with colleagues and/or customers. Classes emphasising student collaboration need to play a prominent role early

in the IT curriculum and throughout it. Warschauer (2007:149) suggests that developing technology-based instruction in schools that is more collaborative and more closely tied to academic content, may be a successful strategy for increasing girls' comfort level with pursuing computer-based careers.

Williams *et al.* (2006a:182) state that all the agile methodologies, for example pair programming, are highly collaborative and social in nature and they believe the collaboration and the social component inherent in these methodologies are appealing to people whose learning and training models are socially oriented. The transition from, for example, programming alone to the more collaborative IT styles, may make a career in IT more attractive to people who have concerns regarding the lack of social interaction in many IT positions.

Collaboration through pair programming has a positive effect on female computer science students at universities in terms of retention, confidence, enjoyment and program quality (McDowell *et al.*, 2006:90; Werner *et al.*, 2005a). Chapter 3 will expand on this topic.

#### **2.5.3.2 Mentors and role-models**

Researchers and practitioners have long realised the importance of individual mentoring relationships between teachers and learners. Girls know about being doctors, teachers, pilots, astronauts, lawyers, and nurses from seeing these professional women on television programs and from family and friends who are currently in these professions. They serve as role-models and mentors to girls and help to attract them into those fields. In IT, there are far fewer women visible in the media or at home, among family or friends, to serve as role-models and mentors in attracting girls into the IT field (AAUW, 2004:16). Image is important to adolescent girls and role-models can help dispel the *mad scientist* and *geek* stereotypes many students associate with science and technology by highlighting their social pursuits (Kekelis *et al.*, 2005b:19).

Reinen and Plomp (1997:65) found the number of female role-models to be one of the main causes for the lack in gender equity, but their findings also suggest that the type of role-model female teachers provide, is qualitatively different from the one male teachers provide, because male teachers had significantly greater self-confidence regarding computers. In a survey of teachers in 21 countries, Reinen and Plomp (1993:353) found that most computer teachers were male and that most female computer teachers had less confidence in their own skills and knowledge.

Girls are encouraged to enter computing by a positive secondary school experience in computing, by having had a teacher who mentored them, but schools lack female role-models in computing (Dryburgh, 2000:197,198). Cohoon (2001:113) determined that the time that computer science teachers of either sex spent mentoring female students correlated with the students' retention in IT. Secondary school IT teachers can therefore make a significant contribution to girls' retention in IT by merely spending time with them and providing them with positive experiences through mentoring and by being a positive role-model.

#### **2.5.4 The IT curriculum**

Most countries do not provide a diverse, interesting curriculum in secondary schools and the standard computer curriculum focuses exclusively on programming or it emphasises basic skills as opposed to problem solving (Reinen & Plomp, 1993:362; Goode *et al.*, 2006:92). Goode *et al.* (2006:100) assert that the image of the solitary, geeky, overworked male computer scientist persists because this image of computer science work is often affirmed through secondary school computer science curriculums.

It is conspicuous that in an advertisement of the University of Johannesburg (2008:12) the following is said (freely translated): *Remember that if you take IT at your school, it often doesn't reflect the enjoyable exciting aspects of the subject – so don't be put off in case IT is not what you expected – at tertiary level it will be much more interesting.* One can thus derive from the quote that in South Africa, the secondary school IT curriculum is seen as uninteresting and unexciting and learners are *put off* from persisting with the subject.

Goode *et al.* (2006:104) describe the following problematic situation in the U.S. In today's increasingly competitive market for university acceptance, learners do not choose their secondary school subjects primarily because of interest in or enjoyment of a particular subject of study. Learners who are planning to apply at elite universities are faced with intense pressure to take courses that will push their results higher in order to make them more competitive in their university applications. The high-achieving, university-bound girls focus on high final results and being accepted into a good university. These learners take classes that counsellors tell them will look favourably on the university application and for which they expect to get good results. Goode *et al.* (2006:105) found that the Programming curriculum is not marketed as one of those classes. Programming at some schools satisfies the same requirement as completing a course like Floristry. At some schools, Programming is on par with Data Processing in terms of fulfilling their secondary school graduation requirement. Thus, girls who are wary of risking their final results, often take the easier of



the courses. The IT curriculum in South Africa echoes the problem in the U.S. because IT is definitely seen as one of the more difficult subjects to achieve good results in, and girls do not want bad symbols on their matric certificate as it will also negatively influence their bursary applications.

Measures for curriculum improvement for girls in technology can be put in place by addressing the interests of women, namely *creativity*, *relevance*, and *collaboration* (Frieze, 2007:56). The curriculum must be relevant to real-world concerns and not abstract and devoid of social relevance (Chou & Tsai, 2007:812). Girls prefer a contextualised curriculum in which computing and technology in general are seen as tools for solving humanity's problems and enriching humanity's experiences (Tillberg & Cohoon, 2005:135). In order to make the curriculum relevant, technology must be infused across the curriculum and especially below university level (Starr, 2000; Gürer & Camp, 1998; Shashaani, 1997:48).

Another measure for curriculum improvement is to use different curricular approaches and teaching methods to appeal to diverse learning styles (Warschauer, 2007:150). Starr (2000) emphasises that it is important to have a flexible curriculum to accommodate people's diverse paths to technology and Powell (2008) asks for different entry points into the curriculum.

One must be mindful of what Sanders (1997) states, that curriculum materials have a subtle but powerful cumulative impact on girls' and boys' understanding of the world and their places in it. Government, education departments and teachers have a critical role to fulfil when setting up the curriculum.

#### **2.5.5 The role of programming in IT**

The very first programmers were six women called "computers" who calculated ballistics trajectories on the ENIAC computer during World War II (WITI, 1997). The ENIAC was the first all-electronic digital computer, a machine of approximately 18 000 vacuum tubes and forty black 8-foot panels. The women had to physically program the ballistics program by using the 3000 switches and dozens of cables and digit trays to physically route the data and program pulses through the machine. Since then, however, programming has become a male-dominated field with secondary school and university programming enrolments primarily male (Goode *et al.*, 2006:104).

Programming is a major source of the gender gap in computing (Sanders, 2002:4). Many females erroneously believe that computer science is nothing but programming, and these

beliefs form the basis for a general aversion to the field of IT (Fisher *et al.*, 1997:109; Dryburgh, 2000:193). Nelson *et al.* (1991:185) found that there is a correlation between taking programming in secondary school and persistence in IT at university level. Dryburgh (2000:193) found that programming experience is predictive of IT success at university, especially for women. Interestingly enough, Goode *et al.* (2006:102) found in their study that all the girls taking the secondary school programming course, had male relatives in technical jobs, which again points to experience and role-models as positive factors to girls' interest in the subject.

Waker (2008), who is manager of the SA Computer Olympiad, is concerned that in South Africa fewer and fewer schools are offering computer programming courses. Many schools are switching to a new subject Computer Application Technology (CAT) which teaches the use of common applications such as word processors, spreadsheets, etc. He feels that the eventual result of this will be that learners will enter or avoid a tertiary programming course without having any idea what programming is, whether they like it or are suited to it. He claims that some learners who should have pursued a career in ICT will not do so; some who should have avoided ICT will waste a year on a course which does not interest them. He recommends that somehow an introduction to programming should be included in another subject or at least in the new subject CAT. Although Waker (2008) does not express his concern over the lack of girls in programming, it goes without saying that in the same website where Waker's press release appears, it transpires that the SA Computer Olympiad for secondary school learners has since 1984 produced only three girls from the total of 125 medal winners!

#### **2.5.6 Academic achievement of girls in IT**

Shotick and Stephens (2006:269) found that there is no difference between the sexes when basic computer skills are tested, but if more advanced user skills in different applications are tested, there is a significant difference between the sexes, with boys achieving higher than the girls.

In South Africa, female university IT students predicted they would receive lower grades for the course than males; in reality they received quite similar grades (Galpin *et al.*, 2003). Shashaani (1997:48) found that women's negative attitudes were not related to their performance in the course and their final grades were actually much higher than those of the men. The overall conclusion from research is that females consistently under-estimate their technology skills and academic abilities regardless of what their skills really are (Warschauer, 2007:149; Shashaani, 1993:169).

Girls' tendency to deprecate their own skills, but to assert confidence in females' skills in general, is referred to as the *We can, but I can't* paradox (Collis, 1985:207; Makrakis, 1993:191). The AAUW (2000:7) modified the so-called paradox or syndrome to: *We can, but I don't want to* attitude of girls toward computer technology: girls insist on their abilities and skills in this area even as they vividly describe their disenchantment with the field, its careers, and social contexts. The *battle of the sexes*, also in terms of intellectual superiority, will always continue in the classrooms of schools, and from the above it is obvious that girls do not view their sex as inferior. The only problem is that from the statistics provided on the number of girls in IT classrooms, one cannot help but think that a battle usually is lost when you do not have the numbers.

## 2.6 INTERVENTIONS

Several interventions and strategies are recommended to eradicate the gender gap in computing. Sanders (2005b:4) recommends that to achieve gender equity in IT, one must know about the problem, have the resources to address it, and be determined to do so.

Littleton and Hoyles (2002:3) and James *et al.* (2006) recommend the following interventions to attract women to ICTs:

- Noticing the gender imbalance at home, in school, and in attitudes
- Changing female participation in ICT activities through role-models and collaborative groupings
- Challenging the dominant paradigm of ICT as culturally and historically male
- Establishing a resource centre for women in ICTs
- Strengthening the research capacity of women and ICTs
- Developing a workable and integrated system for measuring ICT in working life and the education system
- Training programmes for schoolteachers
- Training programmes for girls and young women
- Dissemination and awareness-raising activities

It must be proven to girls that women can have senior positions in IT (Boyd, 2000:90). Success with computers is often perceived as being in front of the computer every hour of every day. Girls however, prefer to have a more balanced life and to maintain a balance between their social life and their work (Henson, 2002). Carlson (2006) claims that we must celebrate successes in the likes of the female programmer, Grace Murray Hopper, who

found the first computer bug and Ada Byron Lovelace, the woman who wrote the first computer program.

A list of do's and don'ts that will help with the retention and increase of girls in the IT industry was drawn up by Val Henson (2002), a female Linux kernel developer, and other women working in the IT industry:

- Don't tell sexist jokes and do protest if they are told.
- Don't use gender-specific abusive names, but show respect to girls in IT.
- Don't take the keyboard away when explaining something, but rather give directions and explain clearly.
- Don't make romantic advances towards the handful of girls in IT classes, but do act friendly.
- Don't complain about the lack of girls in computing, but rather encourage girls in computing.
- Don't treat girls stereotypically by thinking for instance that they are only interested in fashion.
- Don't criticise too much, but rather compliment.
- Don't invite only male speakers, but do ask women to speak as well.
- Don't micro-specialise, do discuss broader topics as well.
- Don't underestimate girls and treat them as independent persons.

Girls agree more on the importance of computers when they use the computer at school (Reinen & Plomp, 1997:73). If one can assume that relevance of working with the computer is a prerequisite for optimal computer use, then the conclusion is that the schools and departments must provide access to computers in the school environment. Attempts to narrow the gender gap in computing should concentrate most particularly on interesting and attracting girls in the earlier years of secondary school (Durndell *et al.*, 1995:226). Galpin (1992:15) supports the idea of a foundation course to enable female students that have not had access to computers, and therefore lack experience, to gain positive experiences. She feels that a foundation course could also be used to counter negative experiences and attitudes amongst female students. Powell (2008:5) suggests that creating more entry points into the field will allow learners to begin with others who are at the same level as them. Reinen and Plomp (1997:77) point out the importance of establishing gender equity policy at the primary and secondary schools and to direct it to parents as well.

Goode *et al.* (2006:112) advocate the need for teachers to be properly prepared and proactive in recruiting females and underrepresented students into their classes and to alter their pedagogy to engage what has become the non-traditional student. They continue by saying that the scientific heart of computer science is *lost in translation* at the secondary school level and as a result the field continues to lose the participation and interest of a broad layer of students, especially females. What seems to work to improve teachers' gender-related behaviour, although not with all teachers, is staff development that emphasises no personal blame for universally-learned gender stereotypes, attention to the WIIFM Rule (What's In It For Me?), praise for progress whenever possible, and the need for teachers to be explicit with students about gender bias, because merely modelling exemplary behaviour is often not sufficient to counteract the students' sexist notions (Sanders, 2005b:20). Sanders (1997) also emphasises the need for training of pre-service teachers on the matter.

There is a need for learning material tailored to local conditions (Galpin, 2004:13). Galpin (2004:13) believes that in South Africa it is important to emphasise what an IT career does not entail in order to avoid confusion with clerical employment and data typing. She also states that it is important to emphasise the status of an IT job, since anecdotal evidence suggests that students aspire to a limited range of careers that are viewed as having status: teaching and nursing for women, and law and medicine for men.

The chapter is concluded with two sections quoted from Goode *et al.* (2006):

*And again, all assignments in the programming course are individually based, with no group work or whole-class discussions. Students without knowledge of how computer science is used in the real world are therefore left with the impression that computer science is solitary in nature. This proved to be a turn-off for the females in our study who conceptualized computer scientists as "anti-social" and "isolated" (Goode et al., 2006:109).*

Goode *et al.* (2006:110) continue on the next page with a section entitled: *Needed: CS teachers to re-vision an alternative classroom culture and pedagogy*

*In our research we have encountered countless well-intentioned educators who do not have access to the knowledge and resources required to present a more accurate and relevant computer science curriculum to students. We place our observations within the context of the immense challenges presented to computer science teachers: teacher education programs do not offer methods classes for computer science teachers, creating no clear pathway for becoming a computer science teacher. Also, unlike other teachers, computer science*

*teachers rarely have a home department, resulting in limited (if at all) opportunities to collaborate with colleagues to develop curriculum and support their teaching endeavours. Additionally, these teachers have technical requirements they must work around, taking on an additional role as troubleshooter for the computers in the classroom. The constantly changing field of computer science also presents barriers to teachers who strive to keep up with the field. The programming language for APCS, for example, has changed from Pascal to C++ to JAVA, all within the last six years. Keeping up to date with these changes, without any professional development support, seems to be an insurmountable challenge. With such bare essential needs not addressed, it is not surprising that few teachers have time or will to think about alternative pedagogy and curriculum that can be meaningful for females and a more diverse pool of male students. And, yet, teachers are one of the critical gatekeepers assuring the existence and success of a diverse classroom, and this must be their challenge.*

From the above quotes it is obvious that an intervention is needed, which can serve as pedagogical tool for IT teachers, not only to address the gender gap in computing, but to benefit all learners in their class. Pair programming is recommended to be one such intervention, because it touches on many issues that women face in the IT field, such as the issues of socialisation, images, and confidence (McDowell *et al.*, 2003b:607; Balcita *et al.*, 2002:34).

Chapter 3 will shed more light on this untapped resource of pair programming.

## Chapter 3

### Pair programming in the IT class

#### 3.1 INTRODUCTION

An introductory IT course has many objectives. One of the objectives of such a course is to excite learners about computing and programming. A second objective is that students better understand basic programming principles. For some learners, the course may be the only formal exposure they have to IT. For others, the introductory IT course creates their first impression of programming and is a deciding factor on whether they will continue in computing (Bishop-Clark *et al.*, 2006:213). Traditionally, IT courses started with programming, but it took learners several weeks before they could create even a simple program. Learners would work individually and become quickly frustrated with the syntax requirements and multiple iterations of work required before any results could be observed (VanDeGrift, 2004:2; Bishop-Clark *et al.*, 2006:214). Not only is programming complex and difficult to learn, there are also cultural and social influences on learners in introductory IT courses (Shashaani, 1994:441).

Traditional introductory programming courses generally require that learners work individually on their programming assignments and it originates from an educational system of individual evaluation. Working with another learner on a homework programming assignment constitutes cheating and is not tolerated (Williams & Kessler, 2000b:65). The only resources available to help learners to overcome possible problems are the teacher, the textbook, and possibly a teaching assistant. They are not allowed to work with their peers, who are struggling with the same material (Werner *et al.*, 2005b). This pedagogical approach teaches learners that software development is an individual activity, potentially giving learners the mistaken impression that programming is an isolating and lonely career (Williams *et al.*, 2002:197).

IT educators must create a collaborative, socially-engaging environment, with clearly defined boundaries, that appeals to the current generation of students and that paints a more realistic picture of the collaborative nature of professional IT careers (Cliburn, 2003:21; Williams *et al.*, 2007). Furthermore, it is clear from Chapter 2 that the continued underrepresentation of girls in IT underscores the need for strategies that foster girls' interest and make the IT class a more enjoyable environment. Pair programming seems to be an approach that will address several significant factors that limit girls' participation in IT

(Werner *et al.*, 2005a), although the benefits associated with pair programming extend to both boys and girls (McDowell *et al.*, 2006:95).

### 3.2 THE ORIGIN AND DEVELOPMENT OF PAIR PROGRAMMING

In pair programming, two programmers develop software side-by-side at one computer. The pair works collaboratively at one computer on the same design, algorithm, or code to create a single solution (Williams & Upchurch, 2001:327). The partners each fulfil a specific role and purpose. One person is the *driver* and has control of the pencil/mouse/keyboard and develops the design or code. The other person, the *navigator*, continuously and actively examines the work of the driver (Beck, 2000:50–51; Williams & Kessler, 2000a:111). Pair programming is very different from a two-person team project where the task is divided in half and each programmer does one half. With pair programming, all code is developed at a single workstation with both partners working together (McDowell *et al.*, 2003a:60).

Pair programming has its origin in the IT industry. Pair programming is a major practice of extreme programming (XP) which is one member covered by the umbrella of agile methods (Jensen, 2005:22) and is proved to be popular in the software industry. Agile developers place more value on individuals and interactions over processes and tools. XP is currently probably the most popular agile software development methodology. XP has five values which aim to promote communication, simplicity, feedback, courage, and respect (Atli, 2006:2).

In industry, programmers collaborate for the majority of their day. Perhaps the largest and best-known example of successful pair programming in the software industry is the Chrysler Comprehensive Compensation system launched in May 1997. Plagued by significant development problems, the project was restarted using XP programming principles, including the exclusive use of pair programming. The payroll system is still operational today and pays approximately 10,000 employees and has 2,000 classes and 30,000 methods. The system's success is largely credited to the reduction in defects and improved functionality brought about by pair programming (Anderson *et al.*, 1998:24).

The software industry has practiced pair programming with great success for years (Williams *et al.*, 2000:19), which sparked the interest of researchers to experiment with industry's pair programming model in the Computer Science classroom (Williams & Kessler, 2001:7). In an educational setting, the way the efficacy of pair programming is evaluated shifts somewhat from the industrial setting. For example, while industry is very concerned with issues of *efficiency* (how many worker hours are needed to complete a block of code), educators are



more concerned with learning outcomes, attitudes of the students and facilitators, and classroom management issues (Williams *et al.*, 2002:198).

A fact often neglected by researchers is that pair programming is a form of collaborative learning which originates from groupwork. Groupwork in educational settings has been researched and recommended for decades and the body of evidence should not be forgotten in the research of pair programming (Mentz *et al.*, 2008:249). Mentz and Goosen (2007:341) found in their study that teachers were uninformed, and they did not seem to appreciate the dynamics of group work and the contribution that group work could make to effective learning and teaching in the IT class. Although the majority of the teachers in their study indicated that they saw themselves as capable of implementing group work in their classes, most of the teachers were unaware of the fact that they should allocate specific roles to each member of the group and that the principles of cooperative learning are important in the success of group work (Mentz & Goosen, 2007:339). While an assessment of individual competency is necessary for professional or academic certification, many teachers view collaboration as an unnecessary grading problem, and prefer to avoid it entirely (Bevan *et al.*, 2002:101).

In the rest of the chapter, the focus will be on pair programming in the IT class. Most of the research on pair programming in IT classes was done at university level, and will be reported as such.

### **3.3 THE PROGRAMMERS IN A PAIR**

The programmers in a pair are in a very unique relationship. In the next section attention will be given to the different roles of the partners in a pair, the composition of programming pairs, attributes of successful pairs and principles the pair should apply for successful pair programming.

#### **3.3.1 Roles of programmers in a pair**

An effective pair-programming relationship is very active (Williams *et al.*, 2002:197). A good pair should be communicating continually during programming and can brainstorm on demand at any time (Williams & Kessler, 2000a:111). The partners work together on the same task in much the same way as an actor and a director of a movie production. An actor delivers the dialogue while the director provides feedback based on a broader view of the entire production. Likewise, the pair-programming *driver* creates the program under the direction of the *navigator* (Howard, 2006:90).

The *driver* is responsible for entering software design, source code, and test cases and focuses on the actual coding (Hanks *et al.*, 2004:176). The driver types at the computer or writes down a design and directly implements the solution by working at the tactical level and explaining the implementation to his or her partner (Williams *et al.*, 2002:197; Davies, 2006:5).

The *navigator* continuously and actively reviews, in real-time, the work of the driver – watching for syntax/logical/design errors, asking questions, raising objections, suggesting alternatives, looking up resources, and considering strategic implications of the work at hand. The managing of programming language reference materials is the task of the navigator (Werner *et al.*, 2004a:165). He/she identifies tactical and strategic deficiencies in the work and thinks at a strategic level (Williams & Kessler, 2000a:111; Davies, 2006:5), i.e., how the current code will fit the existing code, how the new lines of code will be tested, what should be done after this part of code has been written, and will the existing code need to be changed to adapt the new code (Atli, 2006:1). The navigator has a much more objective point of view and is the strategic, long-range thinker (Williams *et al.*, 2002:197).

Pairs should alternate between driving and navigating (Bevan *et al.*, 2002:102), but researchers are not unanimous in their recommendations of how often roles should be switched. It is suggested that pairs switch at least once per hour but sometimes passing the keyboard over every few minutes is necessary (Davies, 2006:4; Williams & Kessler, 2000a:111). According to Hanks (2005), a typical interval is 20 minutes and Atli (2006:1) recommends that if the navigator sees that the driver gets stuck, then this might be an indication that it is time to change the roles. In the end, one program is produced that is the sum of the efforts of both partners.

### **3.3.2 The composition of programming pairs**

Pair forming can occur in different ways (Cliburn, 2003:22; McDowell *et al.*, 2002:38; Nagappan *et al.*, 2003; Thomas *et al.*, 2003:363; Williams *et al.*, 2002:197):

- Learners can be allowed to choose their own partner.
- A software program can be used to make random partner assignments. Katira *et al.* (2004:7) and Williams *et al.* (2006b:411) found that students are compatible if being paired randomly.
- The facilitator can have learners rate themselves on a scale of 1 to 9 in terms of their programming skill and then make partner assignment based on these ratings:

- learners who rated themselves high (7–9) can be paired with learners who rated themselves low (1–3).
- learners can be paired with those who rated themselves similarly.
- Pairs can be formed based on personalities.
- Pairs can be formed based on gender.
- Programmers can regularly be assigned new partners. The switching of partners can be beneficial because learners get the opportunity to get to work with more of their peers and will be less likely to be intolerant of an incompatible partner if they know they will get a new partner in a week or two (Williams *et al.*, 2008:450; Cliburn, 2003:29), but staying with the same partner has its own advantage because of the time it takes to adjust (Cliburn, 2003:28; Vanhanen & Lassenius, 2007:211).

Researchers have investigated several compatibility factors that might affect the effectiveness of a pair-programming team. These factors are: personality type, skill level (actual and perceived), self-esteem, gender, ethnicity, learning style, work ethic, confidence level and time management skills (Salleh, 2008:155). In the next paragraphs, attention will be given to general research regarding pair forming and pair compatibility, but in 3.4.1 a conclusion in terms of specifically the secondary school IT class will be made.

Researchers have found that pair compatibility and successful pairing is significantly influenced by the perceived skill and actual skill of the partners. Students work well with partners of similar actual skill level. Students prefer to pair with someone they perceive to be of similar technical competence, however, facilitators cannot predict this perception nor can pairs be formed based on this fact (Katira *et al.*, 2004:7,11; Katira *et al.*, 2005; Williams *et al.*, 2006b:411; Cliburn, 2003:29).

Students seem to work better with partners with different Myers Briggs personality type (Katira *et al.*, 2004:7). Choi *et al.* (2008:1114) found in their study with university students that partners who were diverse (partially alike and partially opposite) in Myers Briggs personality type, exhibited higher productivity than both alike and opposite pairs. In a comparison between alike and opposite groups, the productivity of the opposite group was greater than that of the alike group. In the secondary school IT classroom with its packed curriculum, very few teachers will have the time to do personality testing, but teachers can take note and at least try to avoid putting two learners with obvious similar personalities together.

Paired learners of similar confidence level can cause greater performance (Thomas *et al.*, 2003:363) and learners are compatible with partners with similar work ethic and different learning styles (Williams *et al.*, 2006b:411).

Although self-esteem and time management skills are identified as compatibility factors, Katira *et al.* (2004:7) and Williams *et al.* (2006b:411) indicate that these factors do not appear to be major contributors to pair compatibility.

Learners will benefit more from same-gender pairs than they do from mixed-gender pairs and more collaboration is found within all-girl groups than in mixed-gender or all boy groups at the secondary school level. Several problems can arise in mixed-gender pairings at school level because cross-gender antagonism is often found and boys dominate the computer (Werner *et al.*, 2004a:162; Katira *et al.*, 2005). These findings are particularly important for this study because it means that teachers should rather pair girls with girls in order to get the full benefit of pair programming and at the same time foster a positive attitude with girls in the IT class.

The study of Katira *et al.* (2005) is the only one that investigated ethnicity by classifying students as either belonging to majority or minority group. Their results showed that minority students were in favour of pairing. However, they did not investigate the effects on pairing compatibility when students within the same ethnicity were paired. Denner *et al.* (2005:94) found in their study on secondary school girls that successful pairings depend mostly on age and ability, rather than on race or ethnicity. These findings are significant for the South African context because of the diverse composition of some classes in terms of race. It seems that teachers can in the first place pay attention to skill levels and not be too concerned about race.

### **3.3.3 Attributes of successful pairs**

The following are important attributes to have in a good pair-programming team and in a partner (Begel & Nagappan, 2008:125; Klawe, 2001:67; Atli, 2006:11):

- Complementary skills: A good pair-programming team have skills complementary to each other's technical and design skills. They work off each other's strengths and weaknesses.
- Good communication skills: Pair programming is a communication-intensive process and partners should be verbal and good listeners. The pair must have compatible communication styles and they must be able to communicate effectively about what they do to others.

- Flexibility: A good programming partner is open-minded and not stubbornly attached to his/her own ideas.
- Compatible personality types: With cooperative personalities, the partners work well together, rather than trying to compete with one another. The partners are tolerant and a mutual trust exists in the team.
- Effective team: The partners in the team work well together by delivering quality code on time. Both partners should be described as team players.
- Ego-less: The team should leave their egos at home. Partners are not overly critical, are permissive to mistake and they disagree respectfully with each other. Partners do not take criticism on their code as a personal attack against them and they exhibit a willingness to co-excel.
- Personality: The most important personality traits of partners in a successful pair-programming team require that partners are open-minded, creative, attentive, logical, flexible and responsible.

#### **3.3.4 Principles for successful pair programming**

Williams and Kessler (2000a:111–114) and Werner *et al.* (2004a:163) identified the following elementary principles for the members of a pair to follow to make pair programming more successful:

- The members of the pair are equal participants in the process and the partners must realise that both own and share everything.
- It is important to take turns driving, so that the navigator does not become disjointed or feel out of the loop. The person not driving should not be a passive observer, but instead should always be actively guiding the driver.
- Make sure the partner stays focused and on-task and follows the prescribed development practices.
- Negative thoughts should be rejected because nobody, no matter how skilled, is infallible and above the input of another.
- Use the “over-the-shoulder” technique of pair programming for defect prevention and defect removal.
- Ego-less programming is essential for effective pair programming in order not to damage the collaborative relationship. Use the words “we” and “us” to talk about the decisions and work.
- Effective communication, both within a collaborative pair and with other collaborative pairs, is paramount. Ask for clarification and confirm understanding.

- Each partner must refrain themselves from any scepticism and develop an expectation of success because pair programming is an unprecedented opportunity for the two to excel as one.
- Taking a break periodically is important for maintaining the stamina for another round of productive pair programming. During the break, it is best to disconnect from the task at hand and approach it refreshed when restarting. However, most secondary school periods are short enough and a break is therefore automatically given.
- With pair programming, the two programmers become one and there should be no competition between the two. Blame for problems or defects should never be placed on either partner. The pair needs to trust each other's judgment and each other's loyalty to the team.
- The partners should not see each other as someone who forces them to compromise, but as colleagues who can help them amplify their talents and skills.

### **3.4 THE FACILITATOR IN THE PAIR-PROGRAMMING CLASS**

There are several benefits for the educator who incorporates pair programming into their classroom. The number of cheating cases facilitators need to deal with is reduced because pair-pressure causes the students to start working on projects earlier and to budget their time more wisely (Williams & Upchurch, 2001:330). Additionally, the students have a peer to turn to for help, and therefore the facilitator's workload is reduced (Williams *et al.*, 2002:209). For the facilitator, observing the collaborative process that revolves around critical thinking is a wonderfully rewarding experience (Howard, 2006:101).

The role of the facilitator in the laboratory is crucial to the success of pair programming. When facilitators explain and reinforce the pair-programming protocol on a regular basis, students are more apt to assume appropriate roles as well as reverse roles when necessary. Facilitators that enforce the pair-programming protocol are more likely to get students involved in team learning. Without facilitator reinforcement, students very easily revert to the individual work with which they are so accustomed (Williams *et al.*, 2002:208).

#### **3.4.1 Guidelines for facilitators of pair programming**

In the next section, implementation and practise-specific guidelines are given to facilitators in order to make the most of pair programming in their classroom. Firstly, attention will be given to the principles of cooperative learning and then some more guidelines will follow.

Johnson and Johnson (2009:106–113) advise facilitators to take cognisance of the five principles of cooperative learning; to ensure that all the members of a group achieve the desired outcomes. Although the groups in pair programming consist of two members only, the facilitator will find that incorporating these five principles of cooperative learning into pair programming, leads to more effective achievement of outcomes (Mentz *et al.*, 2008:259).

The five principles of cooperative learning are described and facilitator guidelines are supplied for the application of each principle:

*Positive interdependence.* Both members of a pair, driver and navigator, should understand that one of them cannot succeed unless they both do. The success of one depends on the success of the other (Johnson & Johnson, 2009:107). One of the objections often heard against pair programming is that some students undeservedly receive credit for group assignments (McDowell *et al.*, 2003a:62). This objection originates from the fact that it is difficult to assess the contribution of a certain member of a pair in achieving the outcomes within the pair. The same concern had been raised in cooperative learning environments. Johnson and Johnson (2009:106) indicated that successful cooperative learning can only take place if certain guidelines are strictly followed by the facilitator. According to these researchers, cooperative learning will not be successful if the responsibility of each member is not assessed together with the promotion of positive interdependence within the group. The facilitator should ensure that the pair takes turns at being driver and navigator (Williams *et al.*, 2008:448). Both partners should understand that their work will be assessed periodically, and they should be informed of the way assessment will be executed (Mentz *et al.*, 2008:249).

*Individual accountability.* Measures should be in place to ensure that both partners contribute towards achieving the expected outcomes. The facilitator must assess the performance of each individual member as well as the overall performance of the pair (Johnson & Johnson, 2009:110). The facilitator should give the results to the individuals and the pair to reflect upon (Johnson & Johnson, 2009:110). By doing this, the facilitator ensures that the partners keep each other accountable. Accountability can be reinforced by the facilitator requesting any partner to demonstrate and explain their program to the rest of the class, and this can be followed by peer assessment. For individual assessment, each member of a pair can be asked to write a program similar to that written in the pair (Mentz *et al.*, 2008:249).

*Face-to-face interaction.* The partners encourage and facilitate each other's efforts to successfully achieve the desired outcomes (Johnson & Johnson, 2009:110). The members provide help and assistance to their partner, exchange resources, challenge one another's conclusions and reasoning, and act in trusting and trustworthy ways (Johnson & Johnson, 2009:111). The facilitator must guide the pair to work toward a common goal (Williams *et al.*, 2008:451) and train them in pair programming in a supervised setting in order for them to experience the mechanics of successful pairing (Williams *et al.*, 2008:447).

*Social skills.* The partners must get to know and trust each other, they should communicate clearly and regularly with each other, they should develop interpersonal skills, and resolve conflict constructively (Johnson & Johnson, 2009:111). As their social skills improve, so the enjoyment factor increases, and the partners become more motivated for the task. The facilitator can prepare the members of a pair by offering activities to help them develop a language to communicate more effectively within their partnership as they program (Denner *et al.*, 2005:94). The facilitator must ensure that the partners reverse the roles of driver and navigator (Williams *et al.*, 2002:208).

*Group processing.* The partners should periodically reflect on how well they are functioning and they must plan to improve their achievements (Johnson & Johnson, 2009:112). The facilitator should give the pair time for reflection on how well they are functioning and for planning on improvements. During pair-programming activities, the facilitator should be available for consultation, guidance and assessment (Mentz *et al.*, 2008:250). The facilitator should provide a systematic mechanism for obtaining students' feedback about their partners and must act upon the feedback when indications are a student is not being an equal participant. Learners must understand that problems with their partner must be surfaced immediately to give the facilitator a chance to correct the situation (Williams *et al.*, 2008:448,450).

Now that attention was given to the principles of cooperative learning, some more implementation and practise-specific guidelines are given to facilitators in order to make the most of pair programming in their classroom.

*Continuous assessment* should take place. Assessment of pair programming implies that the process as well as the outcomes is assessed. Assessment is done by the facilitator, the two members of the pair, each individual learner, as well as the other learners in the class. Self-assessment, peer assessment, teacher assessment, pair assessment and individual assessment all form part of the assessment process in pair programming (Hahn, 2007:54–



76). Assessment should be balanced between individual and collaborative work to ensure that individual learners are learning the course material, and that some learners are not relying solely on their partners (Williams *et al.*, 2008:449; Jacobson & Schaefer, 2008:94).

*Pairing.* Facilitators should attempt to maximise the probability that learners will work well together when they assign pairs (Williams *et al.*, 2008:445). Research has shown that the teacher of a secondary school IT class should pair by skill and confidence level to achieve the most effective and compatible pairs (Bevan *et al.*, 2002:104; Thomas *et al.*, 2003:367).

*Schedule lab time* and make working in pairs in the practical lab mandatory because partner reliability can be a problem if pairs have to do assignments in their own time. In addition, the facilitator must set the expected length of the assignments such that a reasonable percentage of the class can finish within the class time (Melnik & Maurer, 2003; Bevan *et al.*, 2002:105). Although the South African IT curriculum is packed, the learners should be able to write all their programs in class time. In the past, when learners were doing pair programming, many teachers found that giving practical assignments to do at home, proved to be a problem. Some learners did not have a computer at home, some did not have the required software at home, and the storage used for the homework was left, lost or damaged!

*Arrange the furniture.* Make sure that the pair can work comfortably at a workstation and can change roles with ease by having easy access to the mouse and keyboard. The desk must accommodate two persons and the screen must be big enough so that both can read the code without crowding each other (Davies, 2006:4; Williams *et al.*, 2008:450; Williams & Kessler, 2000a:112).

*Manage and control.* The facilitator must be actively engaged in the management of pair interactions like reminding pairs to switch roles. The facilitator must be polite but adamant that switching does take place (Williams *et al.*, 2008:447; Jacobson & Schaefer, 2008:94). The facilitator must also ensure that the roles of driver and navigator are properly understood and executed (Howard, 2006:99). Strict policies for attendance and tardiness should be put into place to protect learners from a non-participatory partner (Williams *et al.*, 2008:448). Ground rules should be established by the facilitator in collaboration with the learners. The ground rules might include accepting responsibility, collective code ownership, incremental change, simplicity, enthusiasm and good manners, and a zero-tolerance of laziness (Melnik & Maurer, 2003).

*Standards and testing.* Less-experienced programmers tend to view their personal coding style as “right”, and anything different as “wrong”. A coding standard should be instituted by the facilitator in order to smooth out differences between partners, because neither individual can then dominate the coding style. The resultant decrease in friction increases the effectiveness of the pair (Bevan. *et al.*, 2002:105). The facilitator should teach software testing techniques early on in the program so that students realise the importance of testing and see the benefits of finding bugs at an early stage of programming (Melnik & Maurer, 2003).

*Different partners.* Learners should have different partners throughout the year. It is beneficial because they have the opportunity to work with more of their peers. In addition, they will be less likely to be intolerant of their partner if they know their relationship only lasts a week or two. It is also beneficial for the facilitator because obtaining multiple forms of peer evaluation on each learner provides a more accurate picture of the contributions of the learner (Williams *et al.*, 2008:449).

*Independence.* The facilitator should encourage pairs to find answers on their own rather than providing them with answers. In doing so, the learners gain confidence in their ability to work independently and to learn about searching for and finding answers (Williams *et al.*, 2008:451).

By creating a climate where collaboration is not only required but also rewarded, learners have the opportunity to view collaboration positively and are able to transfer what they learned from their collaborative experience to future experiences (Nagappan *et al.*, 2003).

### **3.4.2 Pair-programming rules for secondary school learners**

Along with Werner *et al.* (2004a:165) and all of the afore-mentioned researchers, the principles for successful pair programming can be translated into a set of rules specifically aimed at secondary school learners:

1. The driver operates the keyboard and mouse.
2. The navigator follows what the driver is executing on screen and looks out for errors and proposes corrections. The navigator is in charge of programming language reference materials.
3. Positions: The driver is in front of the keyboard with the mouse in hand. The navigator sits on the opposite side of the mouse. The screen is angled so the navigator can clearly view it. The navigator positions his/her chair so that he/she can point at the screen. The

navigator has reference materials organised on the available table space to ensure easy access.

4. The navigator ensures that he/she clearly understands the aims of the program.
5. Partners physically get up and move positions when switching roles.
6. Partners pay close attention to each other when pair programming: they look at each other when talking and listen carefully, they ask for clarification, and confirm understanding.
7. Partners work hard to ensure each person understands what is being created. The driver must check for agreement on operations before execution. The partners point at the screen to support clear communication and the driver, while he/she is working, describes what he/she is doing. The navigator shows notes recorded in the design notebook to the driver to check for agreement.
8. Partners are respectful of each other: navigators do not handle the mouse or keyboard; drivers do not grab for reference materials; disagreement is natural and should be resolved respectfully.
9. Partners share ownership of the project.
10. Partners help each other, create opportunities for each other to learn, promote trading off of pair-programming roles, and share the creation of their project.
11. Partners must assess each other and must be honest in their evaluation.
12. Partners take time to reflect on how well they are functioning and they must plan to improve.
13. Partners use the words “we” and “us” to talk about their decisions and work.
14. Partners must trust each other's judgment and each other's loyalty to the team.
15. Problems with a partner must be surfaced immediately to give the facilitator a chance to correct the situation.

### **3.5 ADVANTAGES OF PAIR PROGRAMMING**

Researchers have identified several advantages of pair programming, as well as behaviours behind these advantages. Although the research done was mostly on students at tertiary level, some of these advantages would probably apply to school learners as well.

#### **3.5.1 A list of advantages of pair programming**

Several factors of pair programming are interrelated. Williams *et al.* (2007) developed the Social Interaction Model of Pair Programming (SIMPP) with five interrelated components, which shows that a change in one factor (for better or for worse) will perpetuate a change in the others. Williams *et al.* (2007) report for example that by producing a higher quality

product in less time, a student's confidence should increase, which in turn, should foster or retain students' interest in IT careers.

Although several factors of pair programming are interrelated and some advantages of pair programming have already been mentioned in Chapter 1, a list of the different advantages of pair programming will be discussed separately and in more detail in the next section.

#### **3.5.1.1 Enjoyment**

Students who pair, enjoy programming more than solo programmers and they are happier and less frustrated (McDowell *et al.*, 2003b:607; McDowell *et al.*, 2006:95; Williams & Upchurch, 2001:327; Cockburn & Williams, 2000:2; Bishop-Clark, 2006:213; Cliburn, 2003:28). Although these studies were done with university students of both sexes, it looks promising for this study where the enjoyment of programming of secondary school girls will come into the spotlight.

#### **3.5.1.2 Attitudes**

Students in paired labs have a positive attitude towards collaborative programming settings and they show positive reactions to working with a partner using the pair-programming paradigm (Nagappan, 2003; Howard, 2006:89). Howard (2006:97–98) found that as the programs became more complex throughout the semester, students' appreciation of pair programming increased and the students' attitudes about pair programming were positive even with the scheduling challenges that they faced.

#### **3.5.1.3 Confidence**

Students who pair, are more confident in their programming solutions and are more satisfied with the programming process than students who work alone (McDowell *et al.*, 2003b:607; McDowell *et al.*, 2006:95; Bishop-Clark *et al.*, 2006:213; Hanks *et al.*, 2004:180). Thomas *et al.* (2003:367) found that students with less self-confidence seem to enjoy pair programming the most. Pair programming therefore produces more proficient, confident programmers.

#### **3.5.1.4 Persistence**

It appears that as a result of pair programming, students that might otherwise have dropped the course, complete the course, and consequently pass it. It also contributes to greater persistence in computer science related subjects and encourages students to pursue computer science careers (McDowell *et al.*, 2002:41; McDowell *et al.*, 2003b:607; McDowell *et al.*, 2006:95; Nagappan *et al.*, 2003, Braught *et al.*, 2008:204). These findings are

significant for this study because it shows that pair programming can most likely already have an effect on the persistence of girls in the subject IT in secondary schools.

#### **3.5.1.5 Program quality**

Programs produced by students working in pairs are significantly better than the programs produced by individuals for the same or comparable assignments (McDowell *et al.*, 2003a:64). Pairing students are able to produce programming assignments of a higher quality; which are less complex and, better to read (Bipp *et al.*, 2008:239), shorter and easier to understand and extend (Williams & Upchurch, 2001:330; Thomas *et al.*, 2003:367; Jensen, 2005:23; McDowell *et al.*, 2006:95; McDowell *et al.*, 2002:41). Hanks *et al.* (2004:176) found that paired students were more likely to turn in working programs, and these programs correctly implemented more required features. This implies that students who pair, are more likely to turn in programs that compile, which shows that pairing students are more successful at overcoming the hurdles that frustrate solo students.

Cockburn and Williams (2000:2) found that students in their pair-programming study had in exchange for a 15% increase in development time, improved design quality and an average reduced defect prevalence of 15%. Pair programmers were also found to generate more concise output, implementing the same functionality in fewer lines of code.

#### **3.5.1.6 Program completion and submission**

Pairing students are more likely to submit solutions to their programming assignments (Hanks *et al.*, 2004:176; Williams *et al.*, 2003:143). Since pair programming encourages learners to work on their programming assignments for submission and because they are actually attempting the assignments, it seems likely that these learners will be learning more.

#### **3.5.1.7 Time taken**

With pair programming, assignments take less time to complete and it has even been reported that pairs can finish programming assignments in half the time it would take an individual (Williams & Upchurch, 2001:330; Cockburn & Williams, 2000:2; Williams *et al.*, 2000:23). On the other hand, Cockburn and Williams (2000:2) found that pairs took 15% longer than individual programmers, but generated 15% fewer defects, but because fixing defects takes longer than initial programming, programming in pairs still took less time than programming alone. Shore (2006) claims that two people working together are less likely to be interrupted, and when an interruption does occur, one person can deal with it while the other keeps the flow going and less time is lost. Sanders (2001) reported on students'

perception that pair programming leads to improved time management. IT is often seen as a time consuming subject and all learners welcome strategies to save time.

#### **3.5.1.8 Bugs**

There is a great reduction in defect count because pair-programming's shoulder-to-shoulder technique serves as a continual design-and-code-review, leading to most efficient defect removal rates (Williams & Upchurch, 2001:327; Cockburn & Williams, 2000:2; Jensen, 2005:23). The longer bugs live in the code, the more difficult they are to fix, but by using pair programming, bugs are spotted earlier in the development process, and it may prevent bugs from getting deeply embedded (Begel & Nagappan, 2008:124). Müller (2007:1460) found that programmer pairs make as many algorithmic mistakes, but fewer expression mistakes than solo programmers and Begel and Nagappan (2008:124) also reported on fewer *petty bugs*.

#### **3.5.1.9 Program design**

Lui *et al.* (2008:197) explored the efficacy of pairs versus individuals in program design-related tasks separately from coding. In both experiments, pairs significantly outperformed individuals, providing evidence of the value of pairs in program design-related tasks.

#### **3.5.1.10 Teacher workload**

Student participation in pair programming causes improved self-reliance, which will lead to a reduced workload for the educator in terms of the number of assignments to grade, questions answered and teaching effort when compared with the teaching of students who work solo (Williams & Upchurch, 2001:327; Nagappan *et al.*, 2003; Sanders, 2001; Cliburn, 2003:20).

Williams *et al.* (2002:197) found that when pairing students could not solve problems between them, they would ask the instructor, but the interaction with the instructor was usually very brief and frequently pairs resolved their own problems without the instructor's help. Overall, instructors spent less time answering questions on syntax errors. Often, solo students who are stuck and need to ask questions, get frustrated because they sit and wait for long periods of time before they can get help. Pairs are more self-sufficient and that in turn reduces their reliance on the teaching staff.

#### **3.5.1.11 Learning and comprehension**

Pair-programmers learn a lot from each other (Williams & Upchurch, 2001:330; Cockburn & Williams, 2000:2) and their partner serves as a learning resource (Simon & Hanks, 2007:76).

Knowledge is constantly being passed between partners, from tool usage tips (even the mouse), to programming language rules, design and programming idioms, and overall design skill (Cockburn & Williams, 2000:2). Pair programming has reciprocal benefits for both partners, because partners constantly learn from each other, which results in each member of the pair being a stronger programmer (Begel & Nagappan, 2008:124). Pair programming increases learning (Sanders, 2001; Hulkko & Abrahamsson, 2005:503) and the comprehension of programming concepts (McDowell *et al.*, 2002:38; Howard, 2006:93). Students appear to learn faster and they learn important collaboration skills they will need in industry (Williams & Upchurch, 2001:327; Cliburn, 2003:20).

Williams *et al.* (2002:208) found that in the pair-programming lab most teacher-student interactions seemed to take the form of extended discussions. Students would want to know how to apply what they were doing to another scenario and these hypothetical discussions of applications showed evidence of higher-level thinking processes that went beyond the scope of the programming assignment. Programming is therefore not learned through mere repetition of concepts.

#### **3.5.1.12 Learner achievement**

Pair programming improves learner achievement. It is consistent with collaborative learning research, which shows that academic achievement is enhanced when an individual learns information with others (Bevan *et al.*, 2002:106; McDowell *et al.*, 2002:41). Nagappan *et al.* (2003) found in their study that on average a higher percentage of pair-programming students completed the CS1 class with a C grade or better when compared with solo programmers. Braught *et al.* (2008:204) found that pair programming seems to *level the playing field* for acquiring individual programming skills since lower achieving students are able to achieve higher lab practical scores when using pair programming.

#### **3.5.1.13 Learner morale**

In spite of all the good intentions and diligent work of computer science educators, students often find introductory computer science courses very frustrating – so frustrating that typically one quarter of the students drop out of the classes. The use of pair programming in the class increases the satisfaction of students and they are happier and less frustrated (Howard, 2006:98; Williams & Upchurch, 2001; VanDeGrift, 2004:2).

Cockburn and Williams (2000:4) reported on an incident in a study where individuals were given one program per cycle to complete while the pairs were given two programs per cycle to complete. After several programming cycles, one pair complained that this arrangement

was unfair because they felt they had to work harder than the individuals during each cycle. The facilitator suggested that the students split up and work as solo programmers as part of the individual group so they would no longer feel they were being unjustly overworked. Both students rejected this offer almost instantaneously and did not complain about the additional workload again. This is a strong indicator of the satisfaction of pair programming.

#### **3.5.1.14 Motivation and focus**

Jensen (2005:24) found that programming pairs are much more motivated and focused than their single counterparts. First-year students in the study of Simon and Hanks (2007:78) reported on the motivational factor of having a partner in times of frustration.

#### **3.5.1.15 Getting stuck**

Novice students frequently get stuck while working on their programming assignments, but students who pair get stuck less often than students who work alone, because the partner might have a solution to the problem (Simon & Hanks, 2007:75; Hanks, 2008:8; Davies, 2006:5).

#### **3.5.1.16 Cheating**

Cliburn (2003:21) asserts that what he had always despised was that some students blatantly copied assignments from others, requiring him to give them failing grades in the course –students did not seem to understand the difference between collaboration and cheating. With pair programming, the number of cheating cases teachers need to deal with, is reduced. Williams and Upchurch (2001:330) believe that pair programming cuts down on cheating because pair-pressure causes the students to start working on projects earlier and to budget their time more wisely. Additionally, the students have a peer to turn to for help, and therefore, do not feel as helpless and as a result the temptation to cheat is greatly reduced. Many IT teachers can vouch for the time spent on dealing with learners that had cheated. Even the time spent proving that learners had copied can lead to a lot of frustration because learners often insert a few spaces and change a few variable names and then expect the teacher to believe it is their own work. It seems that pair programming can reduce this frustration for teachers.

#### **3.5.1.17 Communication**

Students in a pair-programming lab show a high level of interaction with each other. Students discuss issues related to the programming assignment constantly and question, direct, and guide each other throughout the lab session (Williams *et al.*, 2002:208). The partners learn to discuss and bounce ideas off each other and work together, which



improves communication, teamwork and effectiveness (Williams & Upchurch, 2001:330; Cockburn & Williams, 2000:2; Sanders, 2001; Flor, 2006:58; Simon & Hanks, 2007:76).

#### **3.5.1.18 Problem solving**

Govender (2006) found that many students perceive programming as problem solving, and yet solving problems pose the greatest challenge and difficulty in learning to program. Pair programming leads to improved problem solving, strategic thinking and more opportunities for considering alternative solutions (Sanders, 2001; Hulkko & Abrahamsson, 2005:502). Paired students demonstrate more higher-order thinking skills than students who work alone and pair programmers claim that the team has the ability to solve *impossible* problems faster (Williams & Upchurch, 2001:327; Williams *et al.*, 2002:208).

Using the pair programming approach leads to a noticeable improvement in several aspects of introductory programming classes. It fosters interaction between students, allows students to see different approaches to solving problems, and develops students' skills to reflect on what they are learning, and helps them put the newly-gained knowledge in context with the rest of their educational experience (Howard, 2006:98).

#### **3.5.1.19 Code reviews**

Pairing reinforces good programming habits like continuous reviews and testing (Shore, 2006) which in turn leads to more efficient defect removal rates (Williams & Upchurch, 2001:327).

#### **3.5.1.20 Introduction to programming**

Pair programming is an effective pedagogical tool for teaching introductory programming (McDowell *et al.*, 2002:38; McDowell *et al.*, 2003b:602; Williams *et al.*, 2002:197). Students who pair in the introductory programming course are more likely to attempt the subsequent programming class, and more likely to pass it, than those who initially learn to program independently (McDowell *et al.*, 2006:90).

Williams and Upchurch (2001:328) found that students find introductory computer science courses so frustrating that typically one quarter of the students drop out of the classes and many others perform poorly. Incorporating pair programming in the introductory programming class, may ameliorate these problems. Sanders (2001) found in his study that the majority of students were opposed to using XP in the software engineering class, but favoured the use of pair programming in an introductory programming course. Since the

subject IT is an introduction to programming, it seems that the use of pair programming in this subject should be non-negotiable.

#### **3.5.1.21 Going solo**

Students' pair programming in an introductory computer science course does not hamper students' performance in future solo programming courses and students in paired classes continue to be successful in subsequent programming classes that require solo programming (Nagappan *et al.*, 2003; Jacobson & Schaefer, 2008:95). This is a strong indicator that pairing does not result in a significant number of students passing the course without learning how to program, due to a free ride from their partner (McDowell *et al.*, 2006:95).

#### **3.5.1.22 Limited resources**

Pair programming can help teachers with limited computers to overcome this major barrier to integrating technology into classrooms (Werner *et al.*, 2004a:162). Although the South African government is keen to increase enrolments in the broad field of information and communications technology (Department of Education, 2001:26), the matter of funding is a barrier and pair programming can allow for teachers to manage comfortably with half the number of computers required in the traditional IT classes.

#### **3.5.1.23 Social and career benefits**

There is a variety of social benefits, including career benefits to the pair-programming experience. Socially, pair programming makes programming more of a group effort than a solitary piece and it is useful in terms of learning how to work with other people, to communicate with another person effectively and how to coexist with other people (Simon & Hanks, 2007:76). In the study of Simon and Hanks (2007:81) students also reported that they valued the fact that through pair programming they meet more people, they can build relationships and they have someone who *suffers* with them.

From a career preparation perspective, pair programming is a team-building exercise and something students might encounter when they enter the workforce. Students value the practical applications of social interaction (Simon & Hanks, 2007:76; Williams *et al.*, 2007).

### **3.5.2 Behaviours behind the advantages of pair programming**

Williams and Kessler (2002:21–31) point out seven behaviours that support the benefits of pair programming and happen naturally when programming in pairs. These seven behaviours are:

1. *Pair pressure*: People work harder in order not to disappoint their partner and in order to finish the task within the limited time allocated to the pair-programming session.
2. *Pair negotiation*: The two programmers work together to solve a problem. They have different prior experiences but a common goal. The two have to negotiate to share a common approach. By solving a problem with more than one person, it leads to better solutions and allows solving harder problems
3. *Pair courage*: The programmers give each other the courage to do something they might not do if working alone.
4. *Pair learning*: The programmers can learn from their partners' continual critique and review. Additionally, because the programmers work closely together, their knowledge, including programming tips, design skills, tool usage, is transferred between them constantly. When partners are changed the knowledge spreads efficiently among the whole team.
5. *Pair trust*: Pair programmers work in a collaborative fashion. They learn to know and trust their partners to get the work done.
6. *Pair review*: Reviews are an efficient way to find defects, but are often neglected in practice, because developers do not like them. When working in a pair, both programmers review their joint product continuously and immediately. This review technique has been shown effective and enjoyable.
7. *Pair debugging*: Debugging is a tedious and laborious task. However, if we can discuss the problem with someone, we might find new ideas and solutions. When doing pair programming, there is always someone who can comment on the explanation.

### **3.6 DIFFICULTIES OF PAIR PROGRAMMING**

Some limitations and disadvantages of pair programming have also been found by researchers. In the next section the difficulties with pair programming found in the review of the literature will be discussed, but it is important to point out that many of the problems encountered are the result of incorrect implementation of pair programming by the facilitator. If the facilitator had applied the five principles of cooperative learning and followed the guidelines in 3.4.1, a number of the problems could have been avoided (Mentz *et al.*, 2008:259).

#### **3.6.1 Experience conflicts**

A source of intra-pair stress is a significant disparity between the experience levels or skills of the students (Bevan *et al.*, 2002:100,102). Often the faster/smarter/better programmers worry that they will be paired with a partner who is not as smart or skilled as they are. Some say their code is "personal" or that another person would only slow them down (Cockburn &

Williams, 2000:1). The more experienced students are often unwilling to explain the relevant concepts to the other, or to wait for the other to understand the material, and would simply write the entire program alone and submit it as a combined effort. The consequences for the partnership could be detrimental, because it could be that one partner becomes redundant in the process (Begel & Nagappan, 2008:124; Bevan *et al.*, 2002:103; Simon & Hanks, 2007:78). Pair programming in education is particularly important for the stronger programmers, because they are the ones who are most likely to follow a career in IT, yet they are typically the ones who least desire to work with other programmers (Cliburn, 2003:28). One can conclude that in educational settings there should not be a significant disparity between the experience levels or skills of the members of the pair and attention should be given in pairing the experienced programmers.

### **3.6.2 Scheduling conflicts**

Another difficulty of pair programming often mentioned in university settings is that the members of the pair cannot find a time that will suit both of them to complete assignments (Howard, 2006:89; Bevan *et al.*, 2002:100; Ho *et al.*, 2004; Cliburn, 2003:28; VanDeGrift, 2004:5). The two partners require equivalent schedules and suffer twice the scheduling complications and pairing therefore reduces the freedom of work hours of individual contributors (Begel & Nagappan, 2008:124). As mentioned in 3.4.1, secondary school learners should not be expected to complete assignments outside of class, which will eliminate scheduling problems completely.

### **3.6.3 Reliability conflicts**

Bevan *et al.* (2002:103) state that one of the more surprising discoveries in their study was the willingness of students to submit an assignment with both partners' names attached, even if one partner had not contributed at all. An investigation revealed that the students in question believed that it was more important that they appear to be following the pairing requirements than it was to be honest about the division of labour. If the facilitator applied the principles of individual accountability and positive interdependence in cooperative learning, then reliability would have been addressed.

### **3.6.4 Conflict in effort**

Another potential problem for students is the possibility of being paired with a parasite where one member has to do all the work (Thomas *et al.*, 2003:367; Cliburn, 2003:28). Again, if the facilitator applies the principles of individual accountability and positive interdependence in cooperative learning, conflict in effort should not be a problem.

### **3.6.5 Personality clash**

Finding pair programmers that have compatible personalities, value systems and lifestyles is a difficult process. Personality differences are disruptive to productivity and result in potential bad quality programs due to infighting, egos, and one person trying to be the superstar (Begel & Nagappan, 2008:124; VanDeGrift, 2004:5; Simon & Hanks, 2007:76). The facilitator must guide the pair to work toward a common goal (Williams *et al.*, 2008:451) and they must develop their social skills by communicating clearly and regularly with each other, and by resolving conflict constructively (Johnson & Johnson, 2009:111).

### **3.6.6 Disagreements**

Pairs sometimes find it hard to get consensus on ideas and time is wasted on resolving disagreements. To many learners, especially boys, *discussion* is synonymous with *argument* (Begel & Nagappan, 2008:124). However, the effects of disagreements can be minimised by three features of the facilitator's methods: first, by promoting positive interdependence by setting goals and creating incentives; second, by promoting individual accountability by assessing the performance of each individual member as well as the overall performance of the pair; third, by developing social skills to resolve conflict constructively (Johnson & Johnson, 2009:110–111).

### **3.6.7 Programming style differences**

When partners have different approaches to solving the problem in question, and because working with another would force them to change their style, students feel their programming style suits them and they do not want to participate in a paired process (Begel & Nagappan, 2008:124; Bevan *et al.*, 2002:100; Simon & Hanks, 2007:78). As mentioned in 3.4.1, the facilitator should institute a coding standard because neither individual should then dominate the coding style (Bevan *et al.*, 2002:105).

### **3.6.8 Distractions**

In schools and universities learners work on different projects in different subjects and they get distracted and interrupted by deadlines, lecturers, peers, timetables and social engagements (Melnik & Maurer, 2003; Begel & Nagappan, 2008:124). If face-to-face interaction (3.4.1) takes place in the lab where the partners encourage and facilitate each other's efforts to successfully complete the assignment (Johnson & Johnson, 2009:110), and the facilitator sets assignments that can be completed in a lab session, the above-mentioned distractions will be eliminated.

### **3.6.9 Bad communication**

Some people find it difficult to communicate with others and since communication is such an important aspect of pair programming, the pair-programming process will suffer unless learners are directed to improve their communication skills (Begel & Nagappan, 2008:124; Bishop-Clark *et al.*, 2006:217; Melnik & Maurer, 2003).

### **3.6.10 Hard to reward talent**

Students get a feeling of pride from completing programs by themselves and therefore have a weaker sense of accomplishment with pair programming (Simon & Hanks, 2007:85). Students feel that it is difficult to properly attribute rewards to each member of a pair for the work that they do (Begel & Nagappan, 2008:124). If the facilitator implements individual assessment, peer assessment and group assessment correctly, learners should feel rewarded for their efforts.

### **3.6.11 Accountability and co-dependence**

Simon and Hanks (2007:79) found in their study that peer pressure and co-dependence caused problems, because there was no one person accountable to get the task completed or to report back when they got stuck, and at times neither member wanted to admit his or her lack of technical expertise. It is clear that pairs lacked positive interdependence and the facilitators should have given more attention to that.

### **3.6.12 Comprehension**

A negative aspect of knowledge sharing is that pairing may result in programmers with a broader, but more superficial understanding of the system (Ally *et al.*, 2005:5). There is also the possibility that partners can learn mistakes from each other, but by giving feedback on errors to the whole class, this situation can be avoided (Arslan, 2003). If the facilitator applies the principle of group processing in cooperative learning, where partners periodically reflect on how well they are functioning, comprehension should not be a problem (Johnson & Johnson, 2009:112).

### **3.6.13 Implementation of pair programming**

The implementation of pair programming can often be a big challenge to teachers. Programming has traditionally been taught and practiced as a solitary activity and convention speaks against having two people work together to develop code (Cockburn & Williams, 2000:1). Many programmers resist (at least initially) pair programming. There are many facets to this issue, including: a reluctance to share ideas, ego problems where some people think they are always right, lack of trust where comments may be taken as personal

criticism, and *old-school* programmers who find it difficult to change habits (Ally *et al.*, 2005:4). A teacher wanting to implement pair programming in his or her classroom must expect resistance and prepare properly to overcome the initial problems.

#### **3.6.14 Noise**

With pair programming, the class becomes noisy at times and for some facilitators it will seem that the class is a bit chaotic. As long as other classes are not interrupted, the *productive noise* produced by pair programming should be tolerated (Arslan, 2003).

Now that the advantages and difficulties of pair programming for all learners have been discussed, the next section will concentrate on research of girls and pair programming in particular.

### **3.7 GIRLS AND PAIR PROGRAMMING**

Chapter 2 extensively explained why many girls are not taking the subject IT and why they do not want to pursue a career in IT. We saw that girls generally do not enjoy working with computers, they are not interested in computers and they lack confidence in their capabilities. Girls are less likely to pursue and persist in an IT career, because they tend to believe it involves solitary work, entails competition rather than collaboration, and has little social value. The male-dominated culture of computers causes them to question whether they belong. Girls are more likely to pursue and persist in computer science when they have the confidence to problem-solve and explore without fear of failure, see social aspects of computing, and see a value to computing consistent with their self identity (Werner *et al.*, 2004a:161).

#### **3.7.1 Advantages of pair programming to girls**

Although pairing helps all students, it is particularly beneficial to women, because it addresses several of the factors that limit women's participation in computer science (Werner *et al.*, 2005a). The following advantages of pair programming for female students were found by researchers, but the benefits can probably be applied to secondary school girls as well.

##### **3.7.1.1 Enjoyment**

Women working in pairs enjoy the programming process (Werner *et al.*, 2004c) and the enjoyment comes from the usefulness of the program and teamwork (Ho *et al.*, 2004). These findings are significant for this study, since the aim of this study is to determine if pair programming has an effect on girls' enjoyment of programming and the subject IT.

Secondary school girls want to be entertained and they want to enjoy life, so pair programming answers to the need for enjoyment, while they learn at the same time.

#### **3.7.1.2 Confidence**

Women who program in pairs report greater confidence in their solutions and the difference between the confidence levels of men and women is significantly reduced (Werner *et al.*, 2004c; McDowell *et al.*, 2003b:605; Berenson *et al.*, 2004:17). Girls are usually less confident in their abilities than boys, even when their actual levels of competence are the same. This lack of confidence leads girls to doubt their capabilities, question whether they belong, and frequently leads them to select other subjects (Margolis & Fisher, 2002:81). Girls' confidence in their abilities to succeed in computer science declines as they feel they are spending long hours on assignments that their peers appear to finish in comparatively little time. Pair programming help boost girls' confidence in two ways: Firstly, when pairing, girls get to know how much time their peers actually spend on completing assignments and how much they actually know (or do not know); and secondly by working together, the pair tends to more easily figure out an assignment and to finish it faster (Williams *et al.*, 2007; Margolis & Fisher, 2002:82).

#### **3.7.1.3 Interest**

Berenson *et al.* (2004:23) stated that women who are confident in their activities will retain interest in those activities and found in their study that after working collaboratively through pair programming, the women showed interest in IT careers.

#### **3.7.1.4 Persistence**

Women are also less likely than men to persist in IT. Women who program in pairs, have higher retention rates than women who program independently (McDowell *et al.*, 2003b:607; Werner *et al.*, 2005b). The gender gap in retention rates is reduced when students pair, in other words, the difference between women and men in retention is smaller among paired programmers than among solo programmers (Werner *et al.*, 2004c).

The collaborative nature of pair programming teaches girls that programming is not the competitive, socially isolating activity that they imagined and encourages them to pursue computer science as a subject and as a potential career (Werner *et al.*, 2005a). These findings are significant for this study since the second aim of this study is to determine if pair programming has an effect on girls' view of the importance of programming, the subject IT and a career in IT.



#### **3.7.1.5 Social context**

Girls' belief about the solitary nature of IT is confirmed when they enrol in an introductory programming course that requires programming assignments to be done individually (Werner *et al.*, 2005a). Pair programming shows girls that programming is not as solitary as they imagined it to be (Werner *et al.*, 2004b:7).

#### **3.7.1.6 Efficiency**

Ho *et al.* (2004) found that pair programming helps female students work more efficiently in programming tasks because it can reduce debugging time and help them with exploration and problem solving (Werner *et al.*, 2004a:165). Women are more productive when working collaboratively, taking less time and producing a higher quality product. With higher productivity, women will experience more confidence and consequently more interest in IT careers (Berenson *et al.*, 2004:17).

#### **3.7.1.7 Program quality**

Women who program in pairs, produce better programs in terms of functionality and readability (Werner *et al.*, 2004c). The programs of pairers are of a higher quality and take less time to produce than those of solo programmers (Berenson *et al.*, 2004:17).

#### **3.7.1.8 Academic achievement**

Women working in pairs achieve significantly higher grades than those working alone (Werner *et al.*, 2004c; Werner *et al.*, 2005a). Although high grades should not be the ultimate goal in a subject, it is clear from 2.5.4 that girls are not interested in taking IT if they are not convinced that they can achieve high grades.

#### **3.7.1.9 Time management**

Pair programming enhances women's time management skills, because they feel responsible for their partners (Ho *et al.*, 2004; Berenson *et al.*, 2004:17) and would therefore plan ahead to avoid letting their partner down.

#### **3.7.1.10 Safety**

Women fear for their safety when having to work alone in computer laboratories on weekends and late at night. Pair programming addresses this problem efficiently (Werner *et al.*, 2005b). Although more people have a computer at home, it was indicated in Chapter 2 that girls are less likely to own a computer than boys, and in South Africa many people still do not own a computer. Girls in South Africa will therefore still need to do computer assignments away from home, but if they have a programming partner, they do not have to

fear for their safety. However, in 3.4.1 facilitators are advised to make provision for learners to be able to finish assignments in class, which will solve the problem of safety for girls.

### 3.7.1.11 Stereotyping

Pairing is a context where boys and girls can transcend gender stereotypes. A system in which cooperation is given its full value is liberating for both girls and boys (Pryor, 1995:286). Since the male culture of IT is a dominant problem for girls, it seems that pairing can break the stereotyping barrier.

### 3.7.2 Disadvantages for girls

Very few researchers have considered the disadvantages of pair programming for girls in particular, but Ho *et al.* (2004) found the following:

- Once girls are used to pair programming, it can be difficult for them to get back to solo programming. The negative effect comes from the sense of dependency when they are accustomed to pair programming, although not all programs should be done in pairs and individual assessment for grading purposes is a well-known fact of pair programming.
- Like for learners in general, girls also find schedule mismatch and bad pairing experiences as the enemy of effective pair programming.

## 3.8 SUMMARY

There are several reasons to believe that pair programming has the potential to increase girls' computer interest and consequently increase their participation and persistence in IT. Pair programming allows for interactions with a peer, increasing enjoyment and challenging the belief that computing is a solitary activity. It demonstrates the social value of IT by emphasising the importance of collaboration in working with computers, an important skill often not identified by girls considering a career in IT. When implemented correctly, pair programming promotes communication about the learning process, which in turn increases understanding. Pair programming allows participants to learn other learners' ideas for programming and problem solving, and to negotiate strategies with a partner. With pair programming, learners learn to share tasks with their peers on the computer, a skill that is highly valued in the technology industry. Pair programming is an untapped resource worth considering in any programming class, but especially to attract and retain more girls to IT.

*Two are better off than one, because together they can work more effectively. If one of them falls down, the other can help him up. But if someone is alone and falls, it's just too bad, because there is no one to help him (Ecc 4:9–10).*

## Chapter 4

### Research methodology and data analysis

#### 4.1 INTRODUCTION

In Chapter 2, a study was done of girls and the computer environment with specific reference to the attitudes of girls towards computers and the factors that have an effect on girls' attitudes towards computers.

Chapter 3 contained a study of pair programming as a programming technique with specific reference to the advantages and difficulties of pair programming for all learners, but in particular for girls.

In this chapter, the method of research is discussed and the results are given.

#### 4.2 RESEARCH QUESTIONS

The research questions in the empirical study were:

- How does pair programming shape secondary school girls' experience with regard to:
  - their enjoyment of programming and the subject IT
  - their view of the importance of programming, the subject IT and a career in IT?

#### 4.3 AIM OF THE RESEARCH

The aim of the research is to understand how pair programming shapes secondary school girls' experiences. This aim is operationalised as follows (to form pertinent objectives that have to be reached):

- To understand how pair programming shapes secondary school girls' experiences with regard to:
  - i. their enjoyment of programming and the subject IT
  - ii. their view of the importance of programming, the subject IT and a career in IT.

## **4.4 RESEARCH DESIGN AND METHODOLOGY**

### **4.4.1 Research design**

A qualitative approach departing from an interpretivistic theoretical framework was used to conduct the research (see 1.6.2.1). Qualitative researchers are interested in understanding the meaning people have constructed, that is, how they make sense of their world and experiences they have in the world (Merriam, 1998:6). According to McMillan (2000:9,14) the purpose of such research is to provide rich narrative descriptions of phenomena that enhance understanding and it is based on verbal narratives and observations rather than numbers.

A basic qualitative design was used (Merriam, 1998:11) aiming at assisting the researcher to discover and gain understanding regarding the lived experiences from the perspective of girls experiencing programming and pair programming.

Although the majority of previous research linked to the research problem was done quantitatively, the very reason for this research (the shortage of girls in IT) led to a qualitative investigation because the researcher wished to describe the meaning of lived experiences from the perspective of girls experiencing programming and pair programming.

Qualitative studies have limitations as well. The researcher is the primary instrument of data collection and analysis and the subjectivity of the researcher can lead to the problem of bias (Merriam, 1998:42; Mouton, 2001:150). Section 4.4.6 contains a discussion of the measures that had been taken to address the issues of validity and reliability.

The lack of generalisability of results is a primary drawback of all qualitative research, but the goal of the study is to understand the particular phenomenon in depth, rather than to know what is generally true of many.

### **4.4.2 Study participants**

The participants (n=6) were selected from a school in the North-West Province offering IT as a subject and they had the second highest number of girls in their class in the province. The school with the most girls in the Grade 11 IT class in North-West was firstly asked to participate, but they declined due to personal circumstances of the teacher at that time. The Grade 11 IT class had a total of 12 learners, 6 female and 6 male. They had a male teacher at the time of the study and the language of instruction was English.

The Grade 11 female IT learners of the school were selected because they had gained experience of solo-programming in their Grade 10 year.

#### **4.4.3 Data collection methods**

Semi-structured interviews were conducted with the aid of interview schedules to determine the girls' perceptions and attitudes towards programming, the importance they attach to the subject IT and to a career in IT. According to Merriam (1998:72), interviewing is necessary when we cannot observe behaviour, feelings, or how people interpret the world around them.

#### **4.4.4 Data collection**

At the first meeting, interviews of about 20 minutes each were conducted to determine the Grade 11 girls' level of enjoyment of programming and the subject IT and their views on the importance of programming, of the subject IT and of a career in IT.

The IT teacher was willing to implement pair programming in his Grade 11 class and he was trained in a 2-hour session on the implementation of pair programming and the principles of cooperative learning (see 3.4.1). The teacher was supplied with a manual on the implementation of pair programming and a poster on the pair-programming rules for secondary learners was given to him (see 3.4.2).

Shortly after the first interview, the whole Grade 11 class was trained by their teacher in the use of pair programming. The class was informed that pair programming was to be used for all subsequent programming assignments.

After the Grade 11 learners had worked in pairs for 3 months, completing several paired assignments, the girls were interviewed again for approximately half an hour each. The purpose of these interviews was to qualitatively determine if there was a change in each girl's enjoyment of and in their views regarding the importance of programming, of the subject IT and of a career in IT. The girls' experiences during pair programming were also determined. All interviews were recorded.

Table 4.1 indicates the questions used as guidelines for either the first interview or the second interview, or for both.

**Table 4.1** Questions used as guidelines for the interviews

Nr.	Question	Interview#	
		1	2
1.	Did you grow up with a computer in your home? Who in your house uses your computer the most?	•	
2.	Do you have your own computer? When did you get it and what do you use it for?	•	
3.	When and how did you get interested in computers?	•	
4.	Why have you decided to take IT as a subject? Have you ever regretted your decision?	•	
5.	Who has influenced you the most to take IT?	•	
6.	Do you enjoy the subject IT? Why?	•	•
7.	Do you like programming? Why?	•	•
8.	Tell me what in programming you don't like?	•	•
9.	In your view, what skills do you need to be a good programmer? Do you have those skills?	•	•
10.	Describe a computer scientist (programmer) in terms of what the person looks like and what the person does.	•	•
11.	Is IT an important subject to take?	•	•
12.	Are your marks a reflection of your abilities? Are you satisfied with your marks?	•	•
13.	Is programming an important skill to have? Why?	•	•
14.	Is an IT career valuable? Why?	•	•
15.	Are you going to further your studies in IT after school and follow a career in IT?	•	•
16.	What can be done to attract more girls to IT?	•	•
17.	Describe pair programming to a Gr 11 who has programmed before, but does not know what pair programming is. What happens when you get stuck? Do you consult your teacher more or less?		•
18.	Did you enjoy the pair-programming experience more than working alone?		•
19.	Do you think you did a better job with problems because you solved them in a pair?		•

Nr.	Question	Interview#	
		1	2
20.	Was there anything about the experience you particularly liked?		•
21.	Did you experience any particular frustrations with pair programming?		•
22.	Were you more confident in your assignments because you pair programmed?		•
23.	What do you think the others in your class prefer, pair or solo? What about you? What approach do you prefer?		•
24.	Do you think pair programming will work in today's IT workplace?		•
25.	Do you think the pair programming experience defeats the goal because you have to write the practical exam on your own?		•

The rationale behind the questions asked was:

- Questions 1 to 5 in the first interview served two purposes:
  - According to Merriam (1998:82) it is good practice to ask for relatively neutral, descriptive information at the beginning of an interview to put the respondent at ease and to lay the foundation for questions that access the respondent's perceptions.
  - It is important to remember that these Grade 11 girls had already come a long way with IT. Not only had they chosen the subject, but they had also persevered for more than a year. It is therefore essential that we understand their backgrounds in terms of IT.
- Questions 6 ,7 and 8 addressed research aim (i) directly.
- Questions 11, 13, 14 and 15 addressed research aim (ii) directly.
- Questions 9 and 10 assisted in gaining more information on these girls' views on programming and programmers and supported research aims (i) and (ii).
- Since enjoyment is often linked to achievement, question 12 served to gain information on their achievement in the subject and supported research aim (i).
- The reason for this study caused question 16 to be asked in order to gather these girls' views on ways to attract girls to IT and whether pair programming might be one solution.
- The first part of the research aim: *To understand how **pair programming** shapes secondary school girls' experiences*, was addressed through the stating of Questions 17 to 25.

#### **4.4.5 Data analysis**

In order to meet the requirements of a qualitative study, the following was done to analyse the data of the interviews:

- Transcribing the interviews
- ATLAS.ti is powerful software that supports the researcher in handling various types and large amounts of data during the process of qualitative analysis (Muhr, 2004:2, 5). Using the computer program ATLAS.ti 5.2, the data analysis continued:
- The general practice among researchers when using ATLAS.ti is to treat all the data as one hermeneutic unit, but since the study wants to understand how pair programming shapes secondary school girls' experiences with regard to:
  - their enjoyment of programming and the subject IT
  - their view of importance of programming, the subject IT and a career in IT,it would not have made sense to analyse all the interviews as one hermeneutic unit – the progression or regression would not have been evident. Each respondent's interviews were assigned to a single hermeneutic unit, resulting in six hermeneutic units.

Tools in Atlas.ti such as the auto coding tool, object manager and network editor help the researcher to navigate through the data structures and concepts (Muhr, 2004:35). By utilising these tools each hermeneutic unit was then analysed as follows:

- The relevant information was separated from the irrelevant information in the interviews by breaking the relevant information into a number of text segments, coding the themes as they emerged and linking the text segments to the coded themes
- Grouping the codes into code families and drawing networks that reflect the meaning of the experience (see Appendix A)
- Using the networks to develop an overall description of the phenomenon as girls typically experienced it

#### **4.4.6 Trustworthiness**

Although validity and reliability are issues that are hard to control and measure in qualitative research, the following measures were taken to promote the trustworthiness of the study:

- Multiple data collection methods (literature study and interviews with participants) were used.
- Peer review: Discussions with other researchers familiar with pair programming regarding the process of the study and tentative interpretations.
- Engagement: Spent adequate time collecting data.



- Rich, thick description: Provide enough description for readers to be able to determine the context of the study.
- Reflect: Critically self-reflected on the researcher's assumptions, experiences and biases.

#### 4.5 RESULTS

Once the 6 hermeneutic units were analysed, a decision had to be made about the way to group the respondents for reporting the results. It could have been decided to divide them into groups according to their enjoyment of the subject, programming or pair programming, to name but a few, but when the researcher conducted the interviews, it was already evident that the girls did not differ a lot in terms of enjoyment or even importance matters. It was then decided to divide the respondents into 3 groups based purely on their direct answer to question 23 – whether they prefer pair programming or solo programming.

- Respondent 3 stood out, because not only did she prefer pair programming to solo programming, but in her replies to questions 6, 7, and 12 at the second interview (before pair programming was even brought up) she attributed her enjoyment and achievements to the introduction of pair programming.
- Respondents 1, 2 and 4 preferred pair programming to solo programming.
- Since respondents 5 and 6 both replied 'solo programming' to the direct question on their preference, they were grouped together, but it became evident from their responses in the interviews after pair programming that they definitely were not negative towards pair programming. Respondent 6 even contradicted her initial response that she preferred solo programming with statements she made later on in the interview (see 4.5.3.5.1).

The results of the interviews will therefore be reported in three sections based on the abovementioned groups and the quotes of the girls' responses will be given in *italics*.

The results of each group will be discussed as follows:

1. The personal story of each respondent.
2. The themes that relate to the subject IT.
3. The themes that relate to programming.
4. The themes that relate to a career in IT.
5. The themes that relate to pair programming.
6. The themes that relate to attracting girls to IT.

### 4.5.1 RESPONDENT 3

The interviews with respondent 3 yielded the following:

#### 4.5.1.1 THE PERSONAL STORY

The family has had a computer since she was a little girl and her dad and she uses the computer the most. She has been using Microsoft Word, Powerpoint and Excel, which she basically learned through self-study, from a young age. In Grade 8, when her parents allowed her access to the Internet, her interest grew. She initially considered taking CAT, but a male family friend influenced her to take the subject IT– he has been supporting and helping her with the subject ever since. She answered the following when asked why she took the subject IT:

- The world around us is evolving and it's evolving in such a way that it involves a lot of technology, developments, advancements etc. and IT is becoming a huge part of that as well. So, I think that studying IT as a subject in a way will help me benefit in the future.*

Respondent 3 replied to question 4 that she regretted her decision to take IT once in Gr 10 when she didn't achieve her usual 80%, but she commented:

- once you get to fix the areas that you find difficult to deal with, it doesn't really become a problem. So, I haven't regretted it since then because I've actually picked up in my IT and doing quite well.*

#### 4.5.1.2 THE SUBJECT IT

##### 4.5.1.2.1 Enjoyment of the subject IT

Respondent 3 initially struggled to adjust to the subject, which had a negative impact on her enjoyment of the subject. She eventually adjusted and started enjoying the subject, but it is obvious that once pair programming was introduced, the subject became more enjoyable and she linked it directly to better understanding brought about by pair programming.

Before pair programming	After pair programming
<ul style="list-style-type: none"><li><i>Well, yes, now I do. I really enjoy doing it as a subject... last year, when we were just introduced to the subject I found it a bit difficult to adjust to it and understand it</i></li></ul>	<ul style="list-style-type: none"><li><i>Yes, I do. Well, compared to the first term and the second term now I think due to the different methods...We actually have bettered our understanding when it comes to the programming we've learned. Because we tend to work in pairs now. So we tend to help each other and pick up where our mistakes are. So, I think ... our learning is a bit higher and</i></li></ul>

	<i>better... So, I think we're enjoying it a bit more because we understand it better now.</i>
--	--

#### 4.5.1.2.2 The importance of the subject IT

The fact that technology is rapidly developing, makes respondent 3 acutely aware of the importance of the subject IT, although she also feels that it is not for everyone:

Before pair programming	After pair programming
<ul style="list-style-type: none"> <li><i>Well, it depends, I think, on your personality and what you enjoy doing and what you are passionate about ... I think it would be good for someone who knows they can deal with the subject.</i></li> <li><i>The world around us is evolving and it's evolving in such a way that it involves a lot of technology, developments, advancements etc. and IT is becoming a huge part of that as well. So, I think that studying IT as a subject in a way will help me benefit in the future.</i></li> </ul>	<ul style="list-style-type: none"> <li><i>At the moment I do think so because the world is developing and it's revolving around technology. And technology keeps advancing all the time. So, I think if you have IT as a subject, that helps you to know what's going on in your world of technology and how the world is advancing at the same time.</i></li> </ul>

#### 4.5.1.2.3 The attributes of the subject IT

Although not asked directly, respondent 3 made a number of comments regarding the subject IT and her views on the subject did not change from before pair programming to after pair programming. She saw the subject as a means to an end in that it would help her develop to end up with a variety of career opportunities. She described that the initial difficult adjustment to the subject, but mentioned that people who perform well in Mathematics and Science find the curriculum interesting and a unique challenge.

- doing IT as a subject would help me develop a bit in those areas of the programs that I enjoy*
- studying IT as a subject in a way will help me benefit in the future.*
- if you're doing this well in your Science and Maths subjects etc. then it shouldn't be a problem for you to do IT.*
- when we were just introduced to the subject I found it a bit difficult to adjust to it and understand*
- it becomes quite interesting*
- It's really challenging.*
- it requires a lot of research*
- it involves a lot of exploring*
- I do see it as a rare career as well, so there are a lot of career opportunities*
- it's something different compared to what people would normally do*

#### 4.5.1.2.4 Achievement in the subject IT

Respondent 3 was not satisfied with her marks when she started the subject IT, but her marks improved and once pair programming was introduced, she was even more satisfied with her achievements.

Before pair programming	After pair programming
<ul style="list-style-type: none"><li>• after writing my June exam. I wasn't too happy with my results. This was Gr 10. Because I'm normally used to achieving 80 or above in my subjects, so it was like in the 70's, my mark, so I wasn't too happy with it</li><li>• I try to keep in my 80's which I'm currently sticking to with my projects etc. So, I do think it reflects my skills and abilities.</li><li>• IT isn't my best subject. Out of my 4 subjects, IT is probably my lowest but my other subjects I do like quite well, especially Maths, Physics and Bio, I do extremely well</li></ul>	<ul style="list-style-type: none"><li>• I'm quite satisfied now, yes. I do feel that now that it's term 2, I have improved more than in term 1 because I honestly do understand my work better than before when I was working on my own.</li></ul>

#### 4.5.1.3 PROGRAMMING

##### 4.5.1.3.1 Enjoyment of programming

Respondent 3 had always enjoyed the programming part of the subject IT because of the challenge it provided, but because she had not always understood the programming and struggled with mistakes in her programming, she sometimes preferred the theory section of the subject IT. Once pair programming had been introduced, she responded as follows:

Before pair programming	After pair programming
<ul style="list-style-type: none"><li>• Well, I'm a person who enjoys doing challenges and things like that. So, when you get programming to do it's a bit of a challenge..... it involves a lot of exploring so I enjoy that, so I do actually enjoy Delphi a bit more than the theory after a while.</li></ul>	<ul style="list-style-type: none"><li>• Before I didn't, but now that I understand it and know exactly how to fix my mistakes I do enjoy it, much more than the theory. Where as before I enjoyed the theory more than the Delphi.</li></ul>

The detection of small errors continued to be an irritation after the implementation of pair programming.

Before pair programming	After pair programming
<ul style="list-style-type: none"> <li>• There are times where you have long sets of coding to do and for example you are suppose to put maybe inverted commas and not leave a space, but you leave a space and then your program, well that set of coding doesn't work and you can't figure out where the problem is and you just realise it's because you left a little space. So, that tends to get irritating after a while.</li> </ul>	<ul style="list-style-type: none"> <li>• Yes, it's still the errors that you have to fix sometimes. You make little mistakes in your coding, the smallest mistakes and it doesn't work. But once you fix the little error you realize it was something really small that you just had to fix. So, that tends to get a bit irritating at times.</li> </ul>

#### 4.5.1.3.2 The importance of programming

Respondent 3 feels that programming is important in preparation for careers in this technological world we live in.

Before pair programming	After pair programming
<ul style="list-style-type: none"> <li>• if you do have coding skills from an early age it won't be that difficult to grasp onto once you're studying for the career you want to put yourself into. So, it depends on your career, that's what I think.</li> </ul>	<ul style="list-style-type: none"> <li>• the world is revolving around technology and we can not work with those kind of technology, machinery, equipment etc. if we can't program a program that is going to help us to work with this machine. So, programming is really important.</li> </ul>

#### 4.5.1.3.3 Skills required for programming

This Grade 11 girl lists a number of skills required to be a good programmer, but it is conspicuous that before pair programming was introduced, none of the skills mentioned involved other people, whereas once pair programming had been used, the ability to brainstorm was added to the list.

Before pair programming	After pair programming
<ul style="list-style-type: none"> <li>• logical thinking</li> <li>• thinking out of the box</li> <li>• you have to be quite good at Maths</li> <li>• know your basics</li> <li>• being creative</li> <li>• always exploring</li> </ul>	<ul style="list-style-type: none"> <li>• think out of the box</li> <li>• in a way(think) logically</li> <li>• Be creative</li> <li>• Think of various methods in solving a problem, not just stick to one solution</li> <li>• You have to be able to brainstorm a lot of different ideas, different aspects to finding out a way of solving the problem</li> </ul>

She expresses her opinion about people who do not have the required programming skills:

- *But if you are a person that sticks to one routine, doesn't really like getting out of the box, just love like staying where you are and doing what you are good at, at the current moment, then I don't think it would really be good.*
- *if you don't know your basics, you could know how to program but you can't do your program if you don't know how to do your calculations.*

She is quite confident that she possesses the skills required to be a good programmer, but admits that no one is perfect:

Before pair programming	After pair programming
<ul style="list-style-type: none"> <li>• <i>I found it so easy just to start doing programs on my own, thinking of stuff out of the box and trying to do something different every time we learn something in class</i></li> <li>• <i>Yes, I definitely think I do have those skills. I do quite well academically and I do think out of the box. I'm a creative person so I do think I have got those skills.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>Yes, I do. I believe if I didn't, I don't think I would have been able to do my programming as well as I am now.</i></li> <li>• <i>when we do programming we're not perfect at it, we don't always remember everything, we do tend to forget some details at sometimes</i></li> </ul>

#### 4.5.1.4 A CAREER IN IT

##### 4.5.1.4.1 The importance of a career in IT

Respondent 3 felt that it is important to have a career in IT in order to make a contribution to this technological world we live in, but also to contributing to the conservation of our world.

Before pair programming	After pair programming
<ul style="list-style-type: none"> <li>• <i>Yes I think it is very valuable because like I said before: the way technology is developing it involves computers and computers is what's doing everything in our world today.</i></li> <li>• <i>develop things that could make the world a better place. Because we do suffer as well from a lot of global warming etc. due to computers and technology in a way. So, trying to research and do advancements that can help us benefit from computers but also help the environment etc. at the same time.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>I think an IT career is really valuable right now because like I did mention the world is revolving and if we don't have people in those categories or willing to take a chance and do something in IT there's no way we can develop further and we'll basically just deteriorate in stead of progressing.</i></li> <li>• <i>I do see it as a rare career as well, so there are a lot of career opportunities. And it's something different compared to what people would normally do.</i></li> </ul>

#### 4.5.1.4.2 Future plans

Respondent 3 had a definite inclination towards a career in IT, but she was convinced that she did not want to end up stuck in an office behind a computer.

Before pair programming	After pair programming
<ul style="list-style-type: none"><li>• <i>At this stage I'm not that sure about it. But perhaps yes because I have been encouraged to do something in it because people have told me: your IT isn't like as bad as what you think it is, you know. You do stand a chance of doing something in IT and succeeding well in it. So, most probably, maybe, yes.</i></li><li>• <i>I would prefer to like work with different parts of the computer etc. instead of sitting in an office, sitting at the computer and working on the computer. I would rather do the practical part of it. Assembling or breaking apart, taking pieces together or finding out from people what do they think.</i></li></ul>	<ul style="list-style-type: none"><li>• <i>Well, I do plan on doing engineering. I haven't had a fixed idea on exactly what type of engineering I want to do, so IT could be a possibility.</i></li></ul>

#### 4.5.1.5 PAIR PROGRAMMING

##### 4.5.1.5.1 Pairing versus Solo

Respondent 3 preferred pair programming to solo programming and she believed the rest of the class shared her sentiment.

- *I do prefer pair programming*
- *Everybody prefers the pair programming*

She made a number of comparisons between pair programming and solo programming, which made it clear why she preferred pair programming. When comparing it with the previous year when they programmed solo, she described it as much better, because they struggled quite a lot with solo programming. The following were typical comments:

- *when I worked on my own before it tends to get a bit frustrating when you're given a program and you had no idea what to do, you didn't have any guidance.*
- *We got to communicate more, instead of sitting in front of the computer with your own ideas and own concepts.*
- *when you're working on your own and you couldn't really pick up those mistakes.*
- *instead of sitting quietly in front of the computer we get to interact and socialize at the same time.*
- *It's much better than sitting on your own.*
- *last year we had to work on our own the whole year and we did struggle quite a lot.*

#### 4.5.1.5.2 The enjoyment of pair programming

Respondent 3 answered the question *Did you enjoy the pair programming experience more than working alone?* as follows:

- *I did because when I worked on my own before, it tends to get a bit frustrating when you're given a program and you had no idea what to do, you didn't have any guidance. But working in pairs, if you don't understand something you can also ask your friend what's going on. It just helps you to understand things better and it's less frustrating and stressing to like both do it.*

She commented further on pair programming being fun:

- *So it tends to be actually get fun. Once you know that I can do this, you tend to want to go on with that thing.*
- *Everybody prefers the pair programming. It's much easier for everybody and it's fun*

She particularly liked the communication and interaction provided by the pair programming experience:

- *We got to communicate more. Instead of sitting in front of the computer with your own ideas and own concepts. We got to communicate a lot with different people, get to know others better as well at the same time. We got to brainstorm a lot of ideas that sometimes we wouldn't think of on our own to reach certain solutions.*

#### 4.5.1.5.3 The advantages of pair programming

Respondent 3 also pointed out a number of the behaviours that Williams and Kessler (2002:21–31)(see 3.5.2) mentioned that support the advantages of pair programming. These behaviours will be discussed below, as well as more advantages of pair programming:

##### a) Pair learning and comprehension

It is notable that respondent 3, in the first interview, basically expressed a need for pair programming, although she had not known about it. Once pair programming had been introduced, that need was fulfilled and she felt the programmers improved their learning and comprehension. Because they worked together closely, their knowledge and ideas were constantly transferred to each other.

Before pair programming	After pair programming
• <i>Of course you can sit on your own and do it but you wouldn't have really a broad idea on the topic or subject you're working on. But the</i>	• <i>We actually have bettered our understanding when it comes to the programming we've learned because we tend to work in pairs now.</i>



<p><i>more you talk to people and get ideas from different people, I think the better it is. Because a person could give you an idea or something you would never think of but could play a vital role in the project you're working on or the topic you're researching</i></p>	<ul style="list-style-type: none"> <li>• <i>I think we're learning a bit, our learning is a bit higher and better</i></li> <li>• <i>I honestly do understand my work better than before when I was working on my own</i></li> <li>• <i>It's much easier to work and understand how to do things</i></li> <li>• <i>...during pair programming I managed to learn very quickly</i></li> <li>• <i>It just helps you to understand things better</i></li> <li>• <i>sometimes there are ideas we don't think of but the ideas that our friends think of are brilliant and really much better than what you thought of.</i></li> <li>• <i>most of the time what happens is if one person doesn't know what to do usually the other person has a general idea of what's going to happen or what should happen.</i></li> <li>✓ <i>different ideas and different approaches help us to develop much better in stead of just developing in what you know</i></li> <li>• <i>with pair programming it just seem to become easier by the day because you learned faster. It will allow for you to develop much faster, learn faster and understand the work better.</i></li> </ul>
---	---

## **b) Pair trust**

Respondent 3 described how they got to know and trust their partners to solve the problems.

- *you know your friend very well so you wouldn't hesitate to ask*
- *get to know others better as well at the same time.*
- *the ideas that our friends think of are brilliant and really much better than what you thought of.*
- *if one person doesn't know what to do usually the other person has a general idea of what's going to happen or what should happen.*

## **c) Help and negotiation**

Respondent 3 described how they helped each other to solve the problems. They had different prior experiences, but a common goal. They negotiated to come to a shared solution.

- *so we tend to help each other*

- *We seem to add ideas together and try to come to a conclusion, a final conclusion and set up a piece of coding.*
- *We got to brainstorm a lot of ideas that sometimes we wouldn't think of on our own to reach certain solutions.*
- *as you're typing you can notice: you know, I know that this isn't going to work, maybe we can do this, this way*

#### **d) Review and debugging**

Respondent 3 expressed her feelings that prior to pair programming being introduced, the detection of mistakes was an irritation, but she described that mistakes and 'getting stuck' were not such major issues any more since they were working in pairs.

Before pair programming	After pair programming
<ul style="list-style-type: none"> <li>• <i>you can't be perfect in everything you do. But once you notice your mistake and you learn how to rectify it, it doesn't actually become a problem any more.</i></li> <li>• <i>There are times where you have long sets of coding to do and for example you are suppose to put maybe inverted commas and not leave a space, but you leave a space and then your program, well that set of coding doesn't work and you can't figure out where the problem is and you just realise it's because you left a little space.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>pick up where our mistakes are</i></li> <li>• <i>Now that I understand it and know exactly how to fix my mistakes, I do enjoy it.</i></li> <li>• <i>At the same time we're learning how to correct our mistakes. It's easier to remember once you've been corrected.</i></li> <li>• <i>When we get stuck we do try to refer to the textbook together and say: Okay well, I think if we use this piece of coding, this might work or if we adjust this, we can do this.</i></li> <li>• <i>It's easy to remember where you made your mistakes, so when you work individually you realize that: Oh, this is what happened the last time, so I should do this.</i></li> </ul>

#### **e) Pair pressure**

They worked harder and took up their responsibility in order to finish the programs within the limited time allocated.

- *What happens is once we work in pairs, we work a bit faster and get our programs done.*
- *Sharing responsibility...*

#### **f) Pair courage and confidence**

She claimed that she was more confident and she felt more comfortable in her friend's presence, which gave her the courage to do something she might not have done if she had worked alone.

- *It's a bit much more comfortable whereas if you didn't understand something you'd sometimes hesitate to maybe ask your teacher but you know your friend very well so you wouldn't hesitate to ask*
- *I definitely was (more confident) because during pair programming I managed to learn very quickly.*

#### **g) Learner morale**

Respondent 3 mentioned that pair programming was good for their morale.

- *it's less frustrating and stressing to like both do it.*

#### **h) Communication and interaction**

In the first interview, before pair programming was introduced, respondent 3 shared the following sentiment:

- *the more you talk to people and get ideas from different people, I think, the better it is.*

In the second interview, after pair programming was introduced, when asked what she particularly liked about pair programming, she responded as follows:

- *We got to communicate more. In stead of sitting in front of the computer with your own ideas and own concepts. We got to communicate a lot with different people, get to know others better as well at the same time. We got to brainstorm a lot of ideas that sometimes we wouldn't think of on our own to reach certain solutions.*

It was obvious that for this girl the communication and interaction brought about by pair programming was a major advantage. She had more comments on the communication and interaction brought about by pair programming:

- *you get to interact, communicating, socialise with each other and at the same time work.*
- *I do prefer pair programming. Because it's just much better to interact with someone and find out what they think as well.*
- *Sharing responsibility, learning to communicate. It's much better than sitting on your own.*
- *You have to be able to brainstorm a lot of different ideas, different aspects to finding out a way of solving the problem.*

#### **i) Teacher consultation**

Respondent 3 was aware of the benefit of not having to consult their teacher every time a problem arose.

- *We do approach our teachers less because most of the time what happens is if one person doesn't know what to do usually the other person has a general idea of what's going to happen or what should happen.*
- *...if you didn't understand something you'd sometimes hesitate to maybe ask your teacher but you know your friend very well so you wouldn't hesitate to ask.*
- *Even the guys ... they didn't need help and they managed to finish their own projects without any assistance.*

#### **4.5.1.5.4 Difficulties of pair programming**

Respondent 3 did not experience major problems with pair programming, but mentioned that different personalities sometimes complicated the situation.

- *Well, not really. It's just there are some individuals that are sometimes a bit tough to work with so there will be a bit of ups and downs but most of the time we worked well together.*

#### **4.5.1.5.5 Pair programming in the workplace**

She is positive that the use of pair programming in the IT workplace will be successful.

- *I do think so because I think more minds are better than one. So, different ideas and different approaches help us to develop much better instead of just developing in what you know. You get to learn as well at the same time. So, I think it's much better.*

#### **4.5.1.6 ATTRACTING GIRLS TO IT**

##### **4.5.1.6.1 Interests of girls**

In this section, attention will be paid to the interests of this respondent, which might be an indication of the interests of other girls.

Respondent 3 liked the Internet, practical work, talking to people, doing research and making a contribution to society. She did not want to be stuck in an office behind a computer.

- *Once I had access to the Internet ... I started exploring that and I thought it was quite interesting.*
- *Well, I would prefer to like work with different parts of the computer etc. instead of sitting in an office, sitting at the computer and working on the computer. I would rather do the practical part of it. Assembling or breaking apart, taking pieces together or finding out from people what do they think. Do they think developing something like this in IT would be necessary? Do you think it would help? What do you think the side-effects are? etc. That would help, I think, my research and maybe help me develop things that could make the*

*world a better place. Because we do suffer as well from a lot of global warming etc. due to computers and technology in a way. So, trying to research and do advancements that can help us benefit from computers but also help the environment etc. at the same time.*

#### **4.5.1.6.2 Reasons and solutions for the shortage of girls in IT**

This Grade 11 girl felt that most computer games are aimed at boys and the perception exists among girls that IT is about the programming of games, and they are therefore not interested.

- *what girls think about IT is... guys do it because...if you look at the Play station and PC game market, there are games mostly for guys... Most of the guys didn't really have the passion for IT but because of the games they've played and enjoyed, they thought okay: programming of games – that's why they did IT. So, I think most girls think: "Well we are not really into that so why should we take it as a subject?"*

She felt that girls who do take IT should make an effort to challenge, motivate and encourage the girls who still need to make subject choices. She made it clear that the subject was not only for boys.

- *I think girls who are doing IT..., go in and talk to the Gr. 9's (because that's when you make your subject choices) and say: "It's not just a guy's subject. You know, we're doing it, we're fine. We're doing better than the guys actually. So, why can't you do it as well? If it's something you enjoy like exploring, researching and you're a creative person, it's a subject that you will enjoy doing, there's no...you can't fight that fact".*

In the second interview, she saw pair programming as a solution to a number of problems in IT and she insisted that pair programming already be used with the introduction of programming.

- *I think about from the beginning of Grade 10 that pair programming should be implied. Because what happened was, last year we had to work on our own the whole year and we did struggle quite a lot. But, this term when we started with pair programming it just seemed to become easier by the day because you learned faster. So, I think it would be much better if it was implied from the beginning of your IT schooling years. It will allow for you to develop much faster, learn faster and understand the work better.*

#### **4.5.1.6.3 Boys in the equation**

Respondent 3 was very proud of the achievements of the girls compared to the boys in their IT class and she felt that the boys lacked passion for IT.

- *Most of the guys didn't really have the passion for IT but because of the games they've played and enjoyed. They thought okay, programming of games, that's why they did IT.*

- *We're doing better than the guys actually.*
- *We have noticed like from our class we're 12, 6 girls and 6 boys and when it comes to looking at the marks at the end. It's all the girls first and then we get the guys afterwards. So, it's not a guy's subject. I mean us girls can do it as well. None of the guys have actually beaten us.*

She commented that even the boys benefitted once pair programming was introduced:

- *Even the guys we've noticed that their marks, like when we had to do our Delphi projects, they didn't need help and they managed to finish their own projects without any assistance.*

#### **4.5.2 RESPONDENTS 1, 2 AND 4**

The interviews with the above respondents yielded the following:

##### **4.5.2.1 THE PERSONAL STORIES**

###### **4.5.2.1.1 Respondent 1**

This Grade 11 girl gave an impression of being quite an individualist and even a bit of a feminist. She described herself as a bit of a tomboy when she was younger and she played computer games from a young age. She had collected a lot of games through the years, but she also likes to write stories using MS Word.

Many people (including herself) had influenced her to take IT as a subject:

- *Actually I'm a model to myself. That I can show other girls that girls are able to do it.*
- *It's also my father because he started with the computer and I really looked up to him because he brought me into the game.*
- *And I think it's also my mother, she's also part of it because she looks at me to become something more.*
- *...all my cousins... It's funny because they're all male.*
- *But, what's really inspired me is: males.*
- *I think it's the people out there that I really want to show.*
- *So it's the whole family that brought me into it.*

She felt that the reasons for her taking IT as a subject are the career opportunities provided, and also her enjoyment of the subject.

- *I was thinking to go more into graphic designing and game designing so that all followed by my interests and it's just opening up my careers in the computer industry because it is so wide. And I feel because I enjoy it so much I want to have a career one day that I will be able to enjoy as well.*

Respondent 1 had regretted her decision to take IT once during a practical exam, but she commented:

- *the practical exam that I went to, I just said: 'Ag, why did I have to take the subject!' But then afterward it's just that you're growing from it, and just going home and being able to do that program afterwards. 'I am able to do this!' So, actually give me some pride afterwards, you know. So, I have to build myself up again, once I've been put down from the exam.*

#### **4.5.2.1.2 Respondent 2**

This quiet Grade 11 girl worked on and got interested in computers from a young age.

- *Since we grew up with computers I have been like interested to know how a computer works and how to do like games and like programs on computers.*

In their household her brother mostly uses computers for music, while she and her father use computers for work purposes. She claimed that nobody had really influenced her to take IT, but she felt she needed it for the future, because she wants to become a medical doctor or a surgeon.

She had the following to say when asked if she had ever regretted her decision to take IT as a subject:

- *Sometimes I would when it comes to tests because sometimes they ask the questions a little bit confusing but otherwise no.*

#### **4.5.2.1.3 Respondent 4**

This talkative girl's father works in IT; computers thus play a prominent role in their household.

- *there's always computers around and that's how I got to love computers.*

Her dad had obviously woken her interest in computers and influenced her to take the subject, mainly because of the financial benefits of an IT career.

- *My dad. Well, because of the money. Apparently there is a lot of money in IT. That part.*
- *I plan on actually following my father's footsteps. Cause he is like my idol. He actually put me into this.*
- *Before I decided to take it, because my dad has these magazines, IT magazines and all. He used to like push it on us that we should actually read this.*
- *Well, I blame my dad for that part (getting interested) ... he would force us onto the computer cause he wanted us to know the whole... so I was actually forced to actually learn it. But it was 'lekker'(enjoyable).*

She explains her reasons for taking the subject as follows:

- *I wanted to take IT because I thought it would like open a lot of doors for me. And the fact that, my uncle, he wanted to do something that had to do with computer studies and he didn't have any qualifications for it so I thought maybe if I go into IT, it will open a lot more doors for me*

Exams caused her to regret her decision to take IT.

- *When it comes to exams. In Grade 10. At first we started off easily, like with easy work and I passed that in timely, like great. But then when it actually came to the hard work then I was like: 'Is this actually the one for me?'*

Since the respondents were grouped together for reporting purposes, for the rest of this chapter, the respective respondents will be indicated by their respondent numbers below the heading (R).

#### 4.5.2.2 THE SUBJECT IT

##### 4.5.2.2.1 Enjoyment of the subject IT

The three respondents enjoy the subject IT. With all three of them, a slightly more positive attitude is noticeable after pair programming.

R	Before pair programming	After pair programming
1	<ul style="list-style-type: none"> <li>• <i>I do enjoy it and then again I don't enjoy it. You get your on and off days with the IT.</i></li> <li>• <i>But I do enjoy IT over all. It's a very enjoyable subject.</i></li> <li>• <i>I think that's what makes me enjoy IT the most, because it's something that not many kids do.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>Yes I do. Yes I do. Ja, it's very, it's welcome me to like open my mind about computers, and it's really encouraged me to go in that direction.</i></li> <li>• <i>So, ja I've always had a passion for computers and I really enjoy the subject.</i></li> </ul>
2	<ul style="list-style-type: none"> <li>• <i>Yes, I do, a lot. Because I understand it and I know how to work with it so it becomes more enjoyable and our class is also very helpful with each other. We work together and also the teacher, he helps us a lot.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>Yes, I do. Because IT is a subject that I need and if I enjoy it then I will be able to be good at it. So, I enjoy IT because I understand it and it's getting easier to work with.</i></li> </ul>
4	<ul style="list-style-type: none"> <li>• <i>Ja. Even though it sometimes gives me headaches ... But it's actually nice. Starting a program from scratch, doing it on your own and actually developing it, putting in new things, taking out. I love it.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>Ja, I actually do. I'm learning a lot of new things these days. Things I didn't really think I'd be learning</i></li> </ul>



#### 4.5.2.2.2 The importance of the subject IT

Respondents 1, 2 and 4 agreed that IT is an important subject, mainly because of the fact that the world is increasingly relying on technology and the subject IT provides a stepping stone for further studies and career opportunities in IT.

R	Before pair programming	After pair programming
1	<ul style="list-style-type: none"> <li>• I believe it is because apparently CAT isn't like a well known subject that universities recognize. That's why I believe IT can actually open that.</li> <li>• It's also based on your practical and your theory side so if you want to go look for a job, they also check how you are practical, and how you are theory. [giggle]</li> </ul>	<ul style="list-style-type: none"> <li>• I think it's very important because it opens up a lot of careers</li> <li>• Also the practical side... no matter what kind of business you want to go into, it's going to be involving administration or anything like that with computers.</li> <li>• I think it's a really important subject to have.</li> </ul>
2	<ul style="list-style-type: none"> <li>• It is, because technology is developing in our world so it might become compulsory for everyone soon because everyone will be working with computers and jobs would involve more computers.</li> </ul>	<ul style="list-style-type: none"> <li>• I think yes because of more jobs being created using computers and technology, so yes.</li> </ul>
4	<ul style="list-style-type: none"> <li>• Ja, technology wise it could teach you a lot of things. Especially now that we are in a technological phase.</li> <li>• You can gain a lot from the subject. Money-wise and it will definitely benefit them in the long run. ... when they get there they are not going to know what to do, but if they have IT background</li> </ul>	<ul style="list-style-type: none"> <li>• I think it's actually important to take because now we're moving into the technological phase where everything is going to be technological. I don't know how these other ones are going to cope because we're learning a lot of stuff there and since we're moving into a new era and a new phase, we really need this background, it's really going to help us in the future.</li> </ul>

#### 4.5.2.2.3 The attributes of the subject IT

Although not specifically asked, the subject was described as unique and a gateway to further studies, a career and financial gains. The programming part of IT is important and requires practice and effort, whereas the theory part requires studying. The following were typical comments regarding the subject IT:

R	Before pair programming	After pair programming
1	<ul style="list-style-type: none"> <li>• But the practical there's much more depth into it and more practicing and much more</li> </ul>	<ul style="list-style-type: none"> <li>• makes me think harder, it makes me think more creatively, and more openly, and open-</li> </ul>

	<p><i>effort that you have to put in you know. Not just throwing on notes</i></p> <ul style="list-style-type: none"> <li><i>• theory ... it's study work</i></li> <li><i>• It's a very enjoyable subject</i></li> <li><i>• It's something ..., that's different than all the other students are doing.</i></li> <li><i>• it's all about coding, that's what separates CAT from IT, it's the coding</i></li> </ul>	<p><i>minded</i></p> <ul style="list-style-type: none"> <li><i>• it opens up a lot of careers</i></li> </ul>
2	<ul style="list-style-type: none"> <li><i>• for some jobs you need IT because they work with computers like in surgeries</i></li> </ul>	
4	<ul style="list-style-type: none"> <li><i>• Delphi you have to practice it so that you can understand it.</i></li> <li><i>• Cause that's what IT is all about: programming, starting your own programs.</i></li> <li><i>• You can make a lot of money from IT</i></li> </ul>	<ul style="list-style-type: none"> <li><i>• I think people that go into IT, if you start now, in the future you're really going to make big bucks.</i></li> </ul>

#### 4.5.2.2.4 Achievement in the subject IT

These three respondents believed that they can achieve higher marks in IT if they practice and put in more effort. Respondent 1 linked her achievements to her interest in the topics covered. Respondent 2 was satisfied with her marks, but respondent 4 obviously felt dissatisfied, although she believed she had the ability to perform well in IT. None of them reported a decline in their marks since pair programming was introduced.

R	Before pair programming	After pair programming
1	<ul style="list-style-type: none"> <li><i>• I feel that my marks could be at a better level</i></li> <li><i>• I do feel that with my ability to do IT I can put more effort into it, to higher my level mark.</i></li> </ul>	<ul style="list-style-type: none"> <li><i>• I do believe I can achieve higher than I am doing now</i></li> <li><i>• the topics that we're doing now. I don't have much interest in, that's why I don't really give as much effort as I did when we had creative topics, interesting topics</i></li> <li><i>• that I found was interesting for me and then my term mark ..., it was good</i></li> <li><i>• when I enjoy it the most, then it just raises above all the other subjects</i></li> </ul>
2	<ul style="list-style-type: none"> <li><i>• I am, but I do know I can, I can do better if I concentrate more, work harder, if I practice more cause I, I don't really practice a lot because I know what is going on but if I</i></li> </ul>	<ul style="list-style-type: none"> <li><i>• Ja, I would say so (my marks are a reflection of my abilities)</i></li> <li><i>• I can do better if I understand it better but I'm okay with it.</i></li> </ul>

	<p><i>practice more I think I can do better.</i></p> <ul style="list-style-type: none"> <li>• <i>A few subjects it is a little bit higher but like Bio and my IT marks are quite high.</i></li> </ul>	
4	<ul style="list-style-type: none"> <li>• <i>No, honestly no. I know I can do better but when it comes to actually doing it, I don't know, there is something wrong</i></li> <li>• <i>It's not that bad. But the thing is: I know I can do better. That's my only problem.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>No, honestly. I think I can do better and then my marks just aren't up to that level. I know I can do better but my marks don't reflect it. I don't know why.</i></li> <li>• <i>No, I think I'm holding myself back, for what reason I don't know, but I know I can do better. If I just have a lot of like extra classes or just concentrate on it, I know I can do better</i></li> </ul>

### 4.5.2.3. PROGRAMMING

#### 4.5.2.3.1 Enjoyment of programming

Respondents 1 and 4 enjoy the programming part more than the theory. Respondent 1 liked the creative part of programming in particular. Since pair programming had been introduced, respondent 4 felt that she enjoyed the fact that she learned alternative methods of and approaches to programming. Respondent 2 did not enjoy the programming that much, but it seems that once pair programming had been introduced, her understanding had improved.

R	Before pair programming	After pair programming
1	<ul style="list-style-type: none"> <li>• <i>I enjoy it a lot.</i></li> <li>• <i>Things that are more creative than just basic coding</i></li> <li>• <i>if there's something creative involved in that coding then it's more enjoyable to do.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>At times I do enjoy it</i></li> <li>• <i>more on the creative side</i></li> <li>• <i>but the short and quick ways I enjoy coding.</i></li> </ul>
2	<ul style="list-style-type: none"> <li>• <i>I do a little bit but not so much, because it's a little bit hard and confusing, but I do.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>No, because it's a little bit confusing but it's okay. It's easy to start understanding specific computing.</i></li> </ul>
4	<ul style="list-style-type: none"> <li>• <i>I like the programming part but when it comes to exams the theory part is the one I love most. But the programming I know I can do better down that department. But, programming has to get this one.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>Programming ... now that we're going in depth it's starting to open up to me, I'm learning new things. Another side to IT that I didn't really know about, that there's actually different ways. Because I use to think there's only one way of programming, ... but now sir's made it more open-minded. I'm thinking out of the box, like you can think other ways</i></li> </ul>

The respondents felt the typing out of long programs, sitting in front of a computer for long hours, is a part of programming that is not enjoyable. The detection and solving of errors continued to be a problem. Respondent 4 told how the comprehension of programming was a problem especially when they had to work on their own after the teacher had explained new topics to them.

R	Before pair programming	After pair programming
1	<ul style="list-style-type: none"> <li>• <i>And I think that's why I don't enjoy it so much because it's very long. And I'm sure there's shorter, just shorter ways [giggle] just to get that coding down to a point, yes.</i></li> <li>• <i>You don't always want to be cubed up in front of the computer.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>when there's long coding and ... there's no shortcuts then it's really, it's a mission to get through, so I don't really enjoy that.</i></li> <li>• <i>and when there's errors like then you don't know how to figure it out, it's really difficult.</i></li> <li>• <i>during an exam it's very stressful.</i></li> </ul>
2	<ul style="list-style-type: none"> <li>• <i>When I understand it nothing really but for now nothing. Because I understand it and know how to work with it, so it's okay now.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>Nothing specific, just, at first you can understand it but when it gets more complicated and more work, then it's just confusing again. So, ja, it's nothing specific that I don't like.</i></li> </ul>
4	<ul style="list-style-type: none"> <li>• <i>Well, the problem I have is actually when sir gives us coding and stuff we can actually understand it as soon as he gives it, but when we have to go and do it on our own it's like hard.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>The thing is, when he does it there I can do it but then just thinking for myself. I'm still in that phase of actually figuring it out for myself.</i></li> </ul>

#### 4.5.2.3.2 The importance of programming

Respondents 1 and 4 viewed programming as very important before and after pair programming was introduced, because it is the skill that is impressive and it is required in the workplace, whereas respondent 2 felt that it is a such a specific skill that it is only important for people who want to make IT a career.

R	Before pair programming	After pair programming
1	<ul style="list-style-type: none"> <li>• <i>Because your theory counts more than your coding, you feel like it's not that important but because it's IT, it's all about coding, that's what separates CAT from IT, it's the coding.</i></li> <li>• <i>that's what it's all about. Delphi and the coding.</i></li> <li>• <i>I feel coding is very important because, obviously if you go to a work place and you</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>The practical side, yes it's very important because you might one day want to go into programming. And then ... they are going to have to check ... how your ability is with that.</i></li> <li>• <i>And because it plays also a role in your report mark, like not just your theory, I think you should also like have that skill to be able to do coding as well.</i></li> </ul>

	<p>tell them that you took IT then they would: "Oh, you know how to do coding for games and so on, like making software and stuff like that".</p> <ul style="list-style-type: none"> <li>• It's part of that skill that you should have when you put down that you've taken IT.</li> </ul>	
2	<ul style="list-style-type: none"> <li>• It's important if you work a lot with computers but not so much if you just require it like for CAT if you just require it for jobs then it is important, otherwise.</li> </ul>	<ul style="list-style-type: none"> <li>• I don't think it's that important but what I think is important is knowing how to use computers and programs</li> </ul>
4	<ul style="list-style-type: none"> <li>• Ja, that's the one you really have to pay attention to because that's what IT is all about: Programming, ... That's the most important part for me. Theory, ja, you have to know about, computers ... but programming is the most important one.</li> </ul>	<ul style="list-style-type: none"> <li>• I think it goes both ways because you have to be balanced. You have to have the theory side on log down and your practical side because it doesn't help you only on the practical side if you don't know how that happened or the theoretical side. I think it has to be balanced.</li> </ul>

#### 4.5.2.3.3 Skills required for programming

Many higher-order skills were mentioned as requirements for being a good programmer. Creativity, thinking out of the box, logical thinking and problem-solving skills were mentioned more than once. Some personality traits, like being a positive learner, being a progressive person and having the will to learn new things, also stood out as part of the required skills for programming.

R	Before pair programming	After pair programming
1	<ul style="list-style-type: none"> <li>• creativity</li> <li>• know the basics of the computer</li> <li>• you have to think technical, and out of the box.</li> <li>• your way to interact with your teacher. Being able to ask him, is this correct</li> <li>• communicative skills</li> </ul>	<ul style="list-style-type: none"> <li>• you've got to think open-mindedly</li> <li>• you've got to be very creative to figure out different kind of ways</li> <li>• You've got to be technical</li> <li>• you've got to have logic</li> <li>• you've got to be a progressive person... and you've got to think forward</li> <li>• You've got to also be a positive learner</li> </ul>
2	<ul style="list-style-type: none"> <li>• To be able to solve problems</li> <li>• to be able to work well with computers, you know how they function.</li> </ul>	<ul style="list-style-type: none"> <li>• Good at solving problems</li> <li>• wide logical thinking.</li> </ul>
4	<ul style="list-style-type: none"> <li>• Thinking out of the box</li> <li>• Just the will to learn new things</li> </ul>	<ul style="list-style-type: none"> <li>• Creative mindset</li> <li>• think out of the box</li> </ul>

The three respondents covered the whole spectrum in terms of their personal programming skills. Respondent 1 felt fairly confident that she possessed the skills required, respondent 2 was more cautious and felt she possessed some of the skills, whereas it was obvious that respondent 4 found it hard to admit that she might not have the required skills although she is trying.

R	Before pair programming	After pair programming
1	<ul style="list-style-type: none"> <li>• <i>I believe I have most of those skills</i></li> <li>• <i>more on my creative side</i></li> <li>• <i>I change the ways and then I think it's wrong but at the end it gets my program working. That's also from my creative skill, that helps.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>But, ja my creative sense I'm really, I'm on the top of that, ja.</i></li> </ul>
2	<ul style="list-style-type: none"> <li>• <i>Not so much knowing how a computer functions but I'm getting to it now because of our theory but problem solving I think I can.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>Not very good but a little bit.</i></li> </ul>
4	<ul style="list-style-type: none"> <li>• <i>Hey, I'm getting there. Getting there. So far so good.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>I don't know what's happening but I used to have them but now maybe it's because of a lot of stress and stuff, but I'm getting there. I'm trying to get there because if I don't have an open mindset, creative mind, I don't think I will be able to get through it, but I'm getting there.</i></li> </ul>

#### 4.5.2.4 A CAREER IN IT

##### 4.5.2.4.1 The importance of a career in IT

The respondents in both interviews indicated that a career in IT is important mainly because of

- ❖ the financial gains,
- ❖ a shortage of people in IT, and
- ❖ the fact that technology is developing and countries need people in that field.

Respondent 1 stood out because it was so obvious that she felt a career in IT is important especially for women, but she insisted that as long as she is not stuck in an office, she would enjoy a career in IT.

R	Before pair programming	After pair programming
1	<ul style="list-style-type: none"> <li>• <i>Well, [giggle] they pay good. And I believe that's most of the reasons why, because not many people do it.</i></li> <li>• <i>I feel nowadays they're looking for women to</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>It could be because it does give reward, you get paid well if you go into that direction</i></li> <li>• <i>you're stuck in the office and you just do IT, it might not be so rewarding</i></li> </ul>

	<p>do it. ... because there are so many men that do it. Maybe it's just a bit of a spice, having a work place and letting a women sit there, and being in control of the meeting... That's how I feel like it would really change the workplace and make it a more funner thing.</p> <ul style="list-style-type: none"> <li>• It will be like one of my dreams to be able to show the men that I am capable of doing this and showing that independence. When you walk into a room and they can say: "Yo, that's an IT woman, she knows what she's doing".</li> </ul>	<ul style="list-style-type: none"> <li>• you might have such a passion for it but ... you want to have an open career, you want to teach, you want to go out there, and you want to teach people IT and how to program. I think that would be more enjoyable, like that side.</li> <li>• But if you're in the office, and you do IT and you're not really enjoying it, but it becomes more of like just a duty that you must follow, then I feel that it's not really that kind of career if ... you're not going to enjoy it.</li> </ul>
2	<ul style="list-style-type: none"> <li>• Yes, because again with development of technology in our country it's very valuable because when it develops then a lot of people will be using computers, so I think, yes.</li> </ul>	<ul style="list-style-type: none"> <li>• I think yes because again job opportunities.</li> <li>• we swopped partners so we know how other people work. So, if we do follow careers in IT we know how to work with other people and to get to know other people.</li> </ul>
4	<ul style="list-style-type: none"> <li>• Very. Money, money, money, money. You can make a lot of money from IT. A lot. Especially if you love it on top of that. That makes it even better.</li> </ul>	<ul style="list-style-type: none"> <li>• I think it's very valuable especially since what I said earlier about the technological phase. I think people that go into IT, if you start now, in the future you're really going to make big bucks.</li> </ul>

#### 4.5.2.4.2 Future plans

Respondents 1 and 2 planned on furthering their studies in IT as part of the preparation for their careers in graphic/game design and medicine respectively. Respondent 4 felt she wanted to follow a career in IT like her dad, but she felt a bit reluctant because, like respondent 1, she did not want to sit behind a computer for most of her working day. Once pair programming had been introduced, her perception had changed and she became aware of the fact that IT offers a variety of career opportunities.

R	Before pair programming	After pair programming
1	<ul style="list-style-type: none"> <li>• I love computers so that was my goal, to like; if I take IT it's going to open up my career in the computer industry because it's so wide.</li> <li>• I was thinking to go more into graphic designing and game designing so that all followed by my interests, and it's just opening up my careers in the computer industry</li> </ul>	<ul style="list-style-type: none"> <li>• I think because I'm going to go into graphic design, so I think it will play a good like background for me</li> <li>• I feel it will be a good thing that I can carry with me because of the practical side and just to open my mind to new ideas, to follow new sets of coding in that kind of sense</li> </ul>

	<p><i>because it is so wide, and I feel because I enjoy it so much I want to have a career one day that I will be able to enjoy as well.</i></p> <ul style="list-style-type: none"> <li>• <i>So, if I do follow on with IT, it will be because I enjoy it.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>And that it will be a good like experience that I'll take with me.</i></li> </ul>
2	<ul style="list-style-type: none"> <li>• <i>Study IT a little bit more in varsity but not a career in IT just a career that require IT subjects.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>I think I will study IT but I'm not sure of following a career in IT.</i></li> </ul>
4	<ul style="list-style-type: none"> <li>• <i>I plan on actually following my father's footsteps. Cause he is like my idle. He actually put me into this. ... I should actually consider IT. Considering the fact that I actually love computers. ... But I would love to go into IT.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>I plan on furthering my studies</i></li> <li>• <i>Ja, I'm busy researching now. I asked my dad to get some information on different fields of IT. Which fields I can go into, not just behind a desk. ... That's what he does, he's also an IT specialist</i></li> <li>• <i>But now that we're learning new things it's been opened up to me that they don't actually sit behind a desk all the time, typing and typing endless hours. That there's actually different kinds of IT specialists</i></li> </ul>

#### 4.5.2.5 PAIR PROGRAMMING

##### 4.5.2.5.1 Pairing versus Solo

As mentioned in 4.5, these three respondents were grouped together because they preferred pair programming to solo programming and they believed the rest of the class would also prefer pair programming.

R	Own preference	Class's preference
1	<ul style="list-style-type: none"> <li>• <i>Pairs definitely, but sometimes then I want to get the program right on my own because it really makes me feel good about myself.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>I think they would prefer to work in pairs, because you enjoy it more.</i></li> </ul>
2	<ul style="list-style-type: none"> <li>• <i>Yes, I do.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>I think pair</i></li> </ul>
4	<ul style="list-style-type: none"> <li>• <i>I prefer pair programming</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>We all prefer pair programming</i></li> </ul>

Respondents 1, 2 and 4 also made a number of comparisons between pair programming and solo programming which made it clear why they preferred pair programming. The following were typical comments:

- *when I was a single programmer, I didn't really put as much effort in because I felt it was just on me. Like, I didn't feel that much pressure, to like finish the coding, do it well.*



- *As individuals we struggled and then we weren't really, you know, like "let's just leave this" but as teams it becomes like a real competition.*
- *Sometimes we sit as an individual and we try to code but we also have that urge to like communicate with our partners.*
- *On my own I might not know some things that you know and then you can help me with the stuff that I don't know.*
- *when I was alone we had to think and type ... I was like figuring out the problem, actually try and type, you know, two things at a time.*
- *they work better with someone else than on their own. Having to solve the problem on their own: they don't know what they're doing; they don't know if what they're doing is right; they don't know if there are other ways to solve the problem.*
- *Because I was like: "I'm on my own, I can't figure out half of these stuff so what's the use of carrying on."*

#### 4.5.2.5.2 The enjoyment of pair programming

These respondents enjoyed the pair programming experience very much. The reasons they gave for their enjoyment of pair programming boiled down to:

- ❖ motivation, focus and encouragement
- ❖ communication and interaction
- ❖ help with errors and getting stuck less often
- ❖ comprehension

1	<ul style="list-style-type: none"> <li>• <i>I enjoyed it more as a pair</i></li> <li>• <i>I enjoyed that a lot. ... the encouragement from your partner and just everyone else.</i></li> <li>• <i>It's the competition of the teams, you know. As individuals we struggled and then we weren't really, you know, like: "let's just leave this" but as teams it becomes like a real competition. I enjoyed that a lot.</i></li> <li>• <i>I think they would prefer to work in pairs, because you enjoy it more.</i></li> <li>• <i>Sometimes we sit as an individual and we try to code but we also have that urge to like communicate with our partners, you know, and I feel doing both as pairs it's much more fun and you enjoy it more.</i></li> </ul>
2	<ul style="list-style-type: none"> <li>• <i>Yes, I do because I don't get stuck a lot and I get a lot of help and I can understand it better because what I don't know, my partner can explain it to me. So then I get a better understanding and then I know what I'm doing.</i></li> </ul>
4	<ul style="list-style-type: none"> <li>• <i>I prefer pair programming because I think I learn better with another person helping me, they're correcting my mistakes, myself also helping them with their mistakes.</i></li> <li>• <i>We all prefer pair programming because when we get in the class, sir gives us a program, the first thing we do: "Pair programming we want!"</i></li> </ul>

They responded to question 20 by saying in pair programming they particularly liked the feedback from their partner, sharing ideas, and sharing responsibility through swapping roles.

1	<ul style="list-style-type: none"> <li>• <i>Being able to judge your partner. I think when we had a rubric saying, out of 5 ... how would you mark your partner, how did he do in this? So, I felt, I really wanted to know how I did, you know, if I put the input there I should have if I help my partner. So, that really, that was fun to do, to mark and to judge how your partner did because you want to know.</i></li> </ul>
2	<ul style="list-style-type: none"> <li>• <i>I just like getting ideas from everyone else so that if I have to do like for our test, that I have to do then I can use some of those ideas and information for myself, just to help me.</i></li> </ul>
4	<ul style="list-style-type: none"> <li>• <i>The fact that we got to switch roles because when I was alone we had to think and type.</i></li> </ul>

#### 4.5.2.5.3 The advantages of pair programming

##### a) Pair learning and comprehension

The respondents agreed that they learned a lot from their partners, which led to improved comprehension.

- *It gives you a lot of knowledge too, in pairing. That's why you've got to hold it with you; all the stuff you did learn from the pairing in the activities.*
- *What I wouldn't know my partner could know or what my partner doesn't know I could know.*
- *I can understand it better because what I don't know, my partner can explain it to me. So then I get a better understanding and then I know what I'm doing.*
- *we swap roles so that we can get a chance for doing both; to make sure we understand and we know what we're doing*
- *I knew that that person actually helped me and taught me something I didn't know*

##### b) Pair trust

They worked in a collaborative fashion and got to know and trust their partners to get the work done. Some comments they made reflected their trust in their partners:

- *there's a lot of things I didn't know that my partner did*
- *Just help out each other, and just be there.*
- *your partner is there and he's also open-minded to what coding to set out*
- *together it's a mission*
- *When you're in a pair your partner doesn't give up*

### c) Help and negotiation

The two programmers worked together, helping each other to solve the problems. They had different prior experiences, which they utilised to negotiate a solution.

- *so you must help each other out.*
- *there's a lot of things I didn't know that my partner did*
- *You program together but use input from both individuals from the pair.*
- *What I wouldn't know my partner could know or what my partner doesn't know I could know. So, I can help her and then she can help me*
- *Pair programming is when the two of us get together and we share our ideas on the problem that has been given to us.*
- *But when you're with someone, that person can correct your mistakes, they can help you whenever you get stuck; introduce new ways, ways you didn't even know about.*

### d) Review and debugging

One of the comments made prior to the introduction of pair programming was:

- *when there's errors like then you don't know how to figure it out, it's really difficult*

Once pair programming had been used, the respondents felt they got stuck less often, and their partners helped them with the fixing of errors, which led to improved quality of programs. Even the shared responsibility regarding errors was seen as an advantage.

- *I don't get stuck a lot and I get a lot of help and I can understand it better because what I don't know, my partner can explain it to me.*
- *if you see a mistake has arised then you must both put your input and you must both help each other out.*
- *I know that I didn't do it myself and I had a partner with me, so we made a mistake together.*
- *But now that I had someone else with me ... the program actually worked better; I had less errors and stuff.*

### e) Pair pressure

The respondents perceived a natural pressure between the two partners, which resulted in them wanting to become involved, working hard at completing the task at hand and attempting to do it well. They also opined that the pairs competed against each other, which caused them to persevere and not to concentrate only on themselves.

- *I mean when I was a single programmer, I didn't really put as much effort in because I felt it was just on me. Like, I didn't feel that much pressure, to like finish the coding, do it well*
- *now with pair programming it's like, you want to help that other person out, you want to*

*become involved,*

- *now you would be there, doing your program, then you think of something else and then you just go off course. But then when you're with someone that person can actually get you back on track, onto what you guys were doing.*
- *it puts a lot of pressure and I feel that is good because, you know, it makes you want to complete the coding, it makes you want to do the job right.*
- *you're not doing it for your teacher, but also your partner.*
- *as an individual I didn't have that encouragement, like: "Come on, we can do this!"*
- *When you're in a pair your partner doesn't give up, you know. It's always that continuous run – "We're going to complete this program!" It really encourages me to do it.*
- *It's the competition of the teams. As individuals we struggled and then we weren't really, you know, like: "let's just leave this" but as teams you like, it becomes like a real competition.*
- *you uplift one another. It's not only about you.*

#### **f) Pair courage and confidence**

The respondents described how their partners encouraged them, which gave them confidence that:

- ❖ *their programs would work,*
- ❖ *they do have the ability to succeed and persevere – even on their own,*
- ❖ *they would not have to carry all the responsibility, and*
- ❖ *their knowledge and skills had improved.*

The following were typical responses:

- *Yes I was more confident because it was teamwork, I felt it should work, it should be able to come out. Because, as an individual I would have like doubt. "Did I really do well?" But as pairs I feel I should have this confidence that the program will work and that we did put enough effort in that we should.*
- *And I feel like that really boosts a person because they want to show their confidence and they want to show them that they can do it.*
- *it's more like you gain that pride because, you know, if we can do it as pairs then I should be able to do this.*
- *When you're in a pair your partner doesn't give up, you know. It's always that continuous run – We're going to complete this program! It really encourages me to do it, ja.*
- *they might feel that they're not good enough to be able to be in IT. That might lack their confidence, you know, like I'm not good enough. But if you encourage that person and you help them out then I feel like they'll enjoy it more.*
- *Yes(I was more confident) because then I know that I didn't do it myself and I had a partner with me, so we made a mistake together.*

- *you uplift one another*
- *Ja(I was more confident), because I knew that that person actually helped me and taught me something I didn't know.*
- *But now that the pair programming is introduced I'm like: there's still hope because I'm learning new things, someone is helping me on my mistakes, I'm not actually alone. They're helping me along the way, holding my hand, you know, just introducing me to new ways of actually doing it.*

#### **g) Program quality**

The respondents agreed that, as pairs, they produced programs of better quality, because they could share knowledge and they felt an urge to perform well.

- *we did that one practical activity that we had to do as pairs and I feel like I would have, no I DID do it better than I would have if I was like on my own, busy coding.*
- *I feel it does improve your quality of work because you want it to be correct and you just do it together, it really helps out*
- *Yes, because our ideas are different so if we can put our ideas together then we can have something better.*
- *But now that I had someone else with me, it actually worked better, the program actually worked better; I had less errors and stuff.*

#### **h) Social and career benefits**

The respondents felt that pair programming is useful in terms of learning how to work with other people, how to communicate effectively with another person, and how to coexist with other people. They also reported that they valued the fact that through pair programming they meet more people and they can build relationships.

- *you interact with your friends while you're coding.*
- *Sometimes we sit as an individual and we try to code but we also have that urge to like communicate with our partners*
- *nowadays when we pair up I want to do assignments because I want to have like that communication with my partner.*
- *I think it will work out because you get to work and you want to be able to work with other people.*
- *it's more exciting, you know, it's more teamwork.*

Respondent 2 commented that from a career preparation perspective, pair programming is a team-building exercise and something they might encounter when they enter the workforce.

- *we swopped partners so we know how other people work. So, if we do follow careers in IT we know how to work with other people and to get to know other people*

#### **i) Teacher consultation**

The respondents were unanimous in their opinion that they consulted their teacher less since they got help from their partners

- *So, it's not always: "Sir! Sir! Please come and help me!"*
- *I feel less because I, there's a lot of things I didn't know that my partner did.*
- *Less, I consulted him less. Because there used to be so much happening, and then I would get confused. ... But now that I'm with someone else she or he can help me on the problem that I had.*

#### **j) Program completion and submission**

The respondents reported how pair programming encouraged them to work on their programming assignments and complete the task at hand.

- *you get your assignment done because you just work together and you figure things out.*
- *As individuals we struggled and then we were really, like: "let's just leave this" but as teams you like, it becomes like a real competition.*
- *it makes you want to complete the coding, it makes you want to do the job right*
- *And every time I gave up, that was it, you know. When you're in a pair your partner doesn't give up, you know. It's always that continuous run – We're going to complete this program!*
- *And just being able to say we can do this and you can't leave the computer unless you two figure it out*
- *Instead of actually leaving us all alone because now you would be there, doing your program then you think of something else and then you just go off course. But then when you're with someone that person can actually get you back on track, onto what you guys were doing.*

#### **k) Time taken**

The respondents thought that pair programming equipped them to complete their programs in less time than would normally have been the case if they had to work on their own.

- *because of less problems and less complications with programs, less time*

#### **l) Going solo**

The respondents agreed that they could utilise everything they learned through pairing when they might have to work on their own, like in the exam, but one respondent expressed her concern that she would not have help when going solo.

- *you hold it with you, all the stuff you did learn from the pairing in the activities.*
- *If I could help out my partner then I can do it on my own*
- *But now that they were there with me, we can figure out the problem on our own. I think it's much better now.*

- *it just worries me a little bit for the exam because I won't be able to get help if I'm stuck. ... A little bit less prepared(for the exam) because you're getting help from another person but also benefits you because something that you didn't understand you now may know.*

#### **m) Persistence**

One respondent admitted she wanted to drop the subject, but she explained:

- *I was one of those girls that wanted to quit IT. Because I was like: I'm on my own, I can't figure out half of these stuff so what's the use of carrying on. But then sir luckily introduced this whole pair programming.*

#### **4.5.2.5.4 Difficulties of pair programming**

The respondents expressed their frustration at a partner who would not give their full cooperation.

- *if you have a lazy partner, that really frustrates me. Like, all the work is on you and the pressure as well.*
- *Only when somebody wouldn't participate, that really frustrates me. When you would be there, actually giving it your all, then that person will just be slacking and you know that person needs it or you guys need each other*
- *when your partner forgets their textbook and then you're the one just rushing through your textbook.*

The girls in this particular class felt the boys in their class were uncooperative partners.

- *I preferred it on the girls side because I must be honest some of the boys can be really lazy, you know. They can sit there, they can lay back and say: "You can read it and you can type it".*
- *But both girls they put equal effort in because it's not a sexual issue.*

The respondents reported that occasionally they would experience problems to negotiate a solution with their partners.

- *Just when you know you're right and then your partner might think you're wrong and just won't listen even though you would listen to them*
- *If you both don't know what's going on*

#### **4.5.2.5.5 Pair programming in the workplace**

These girls agreed that pair programming can be used effectively in the workplace, because problems can be solved by sharing knowledge, and it meets the social need of people to not be on their own.

- *Pair programming in the workplace? Isn't that what they do now? Like working together trying to solve the problem. Do they do that? Ja, that's what I think it actually works in the workplace. Then they can try and figure out the problem*
- *I think it will work out because you get to work and you want to be able to work with other people.*
- *It could work because if you're not doing your job correctly on your own, you have your workpeople to like help you out*
- *It might because of what different programmers know. They can put it together. So it could be better.*

Respondent 1 jokingly went as far as claiming that pair programming would be more effective in the workplace if females were involved.

- *I think it relies a lot on females and males in the workplace. So, if there's a lot of females[giggle], I think it would run more smoothly.*

#### **4.5.2.6 ATTRACTING GIRLS TO IT**

##### **4.5.2.6.1 Interests of girls**

In this section, attention will be focused on the interests of these three respondents, which again might be an indication of the interests of other girls.

More than one respondent made it clear that they feel restricted sitting behind a computer, working on their own.

- *I don't want to be just behind a desk. I feel restricted; I want to actually go out there.*

Respondent 1 had a lot to say about her interests which did not necessarily reflect the others' sentiments (especially: "create a program that you have to shoot that one guy and then he just knocks over")

- *I was a bit of a tomboy in my younger life and then I just started becoming more interested in what they (my male cousins) were interested in. That's why I related so much to them and all their enjoyment things.*
- *more on the creative side like if you are going to make an image move or displaying an inputbox on the screen. That's more, you know, interesting to find out how to code that*
- *Well, we did arrays now and I thought that was really interesting, ...because you can like link it globally to all your other set of coding*
- *Most of the girls enjoy things ...if there's something that's interesting to do. Like images and all that kind of stuff, that visual kind of stuff.*
- *So, if you have to create a program that you have to shoot that one guy and then he just knocks over. It would be entertaining, like little games like that. And I feel there should be more of that instead of just the same topics the whole time, you know, that's not that interesting.*



#### 4.5.2.6.2 Reasons and solutions for the shortage of girls in IT

The respondents gave a number of reasons for the shortage of girls in IT, which included that the subject is known to be difficult; girls lack confidence; they do not enjoy working on their own; the subject content does not speak to their interests; girls feel that IT is for boys; and the condition of a good Maths mark for acceptance into IT drives them away.

- *The reason why they didn't choose it was because people hearsay, saying: "No, IT is hard. When you get to matric it's going to be hard. You're not going to pass. So, why take a subject that you know you are not going to pass".*
- *they might feel that they're not good enough to be able to be in IT. That might lack their confidence, you know, like I'm not good enough*
- *the reason is because they don't enjoy it as much, as an individual.*
- *And IT, I feel it should be more creative, you know, especially for some of the girls. Most of the girls enjoy things that are, you know, more like... just you enjoy it more if there's something that's interesting to do. Like images and all that kind of stuff, that visual kind of stuff.*
- *girls don't think they can do it cause they think it is mostly for boys.*
- *Some of them actually wanted to take IT but now their Maths marks were too low, so they couldn't. They had to take Maths in order to take IT and they would be failing Maths but they really want to do IT*

These girls were very enthusiastic about promoting IT as a subject to girls. The following solutions to the shortage of girls were suggested:

- *Putting up signs and saying: "Girls for IT!"*
- *I feel that there should be something to welcome girls, to be able to say: "Come on girls, why aren't you taking IT? Come on, it's a subject many girls had passed through the years taking IT. So, we really need more IT, girls taking it".*
- *I know sir is the only teacher that can teach IT but just bringing women into the school to teach IT. I think that would also relate to mark the girls, the girl learners and say: "Hey, there's a women teaching IT". So, you know, that's also a hint that girls can, as well.*
- *And you've got to have like a very uplifting teacher. Like I feel he's got to be able to help out everyone and he's got to motivate you.*
- *I feel that kind of competitiveness, you know like: "Girls can do what guys can do", will encourage more girls to do IT*
- *I think, people can mention that girls can do anything, you just need to try or, teachers can ask girls that might be interested to like write a brief test on what they know about computers if they want to know more about computers.*
- *I think just letting them know ... You don't have to be a boy to work with computers and programs.*

- *Just, more positive stuff about IT. This whole negativity, they should leave it behind. ... It's just this whole: 'You're not going to pass thing'. That's what I think.*
- *we have to promote IT. I think we have to get, we have to tell these girls that: "You guys, this subject's actually nice. You can gain a lot from the subject money-wise" and it will definitely benefit them in the long run.*
- *should actually get a fun side to this. That would actually make them come. Well, I would come if I hadn't done it.*
- *If it's IT, the first thing you think about is guys. We should actually scrap that concept. No, you can get girls along.*

The respondents agreed that pair programming might be a solution to the shortage of girls in IT. They felt that pair programming would help girls to persist and they felt strongly that it should be used from Grade 10 – and not only in their school.

- *I think yes(introduce pair programming in Grade 10), it might help because boys and girls think differently.*
- *Ja, I think so (pair programming will cause less drop-outs) because they can get help from other people. So it might just help them improve marks as well.*
- *If they had introduced it earlier I think we would have had much more girls.*
- *If they know about this pair programming they would actually come because they know that someone would be there to help them along the way, they would help them with their mistakes, help them wherever they get stuck.*
- *I think they should actually introduce it into other schools.*

#### **4.5.2.6.3 Boys in the equation**

The respondents (especially respondent 1) felt adamant to show the boys what they as girls are capable of, and they were very proud of the fact that the girls in their class achieved better marks than the boys.

- *somehow the boys have this attitude that they are the only ones that can do it,*
- *The girls are doing better than the boys in our IT class. We are getting a higher average than they are. And then they still stand up with such confidence saying: No, boys are this and that.*
- *Let's show the guys.*
- *And I feel that kind of competitiveness, you know like: Girls can do what guys can do.*

The following were some comments made about the boys in the class:

- *They love gaming so much. The boys in our class play a lot of games so they're good with how you program games.*
- *boys and girls think differently.*
- *I must be honest some of the boys can be really lazy.*

### 4.5.3 RESPONDENTS 5 AND 6

Although they were grouped together based on their direct response to question 23 that they prefer solo programming to pair programming, the results for these two respondents below will show that they actually enjoyed pair programming.

#### 4.5.3.1 THE PERSONAL STORIES

##### 4.5.3.1.1 Respondent 5

Her brother is the “IT boffin” and uses computers the most in their family. When her brother took the subject IT when he was in Grade 10 and she was in Grade 8, she became interested in what he was doing. She thought:

- *Oh, that's cool. I also want to do that!*

Her brother and mother influenced her the most to take IT for the following strange reason:

- *because they told me there was no possible way that I could do it, so I proved them wrong and got 86% for my first term of doing IT so they all were surprised so I was really happy about that. {giggle}*

When asked why she took the subject IT, she answered the following:

- *First of all for the, how it looks on university entrance and how much points you get for it and second of all, it's a subject I really, really do enjoy and I don't want to do CAT, it's too easy for me so it's a bit of a challenge that I thought that I could get through.*

She admitted that she occasionally regretted the decision to take the subject, because the programming is complicated, but she continued by saying the teacher's assistance was important in overcoming her regrets.

##### 4.5.3.1.2 Respondent 6

In this respondent's house there are only 3 women, because her dad passed away and she described their computer use as follows:

- *we weren't all, you know, attacking the computer. Like it wasn't very popular in our house. But we just used it for like, doing homework, doing projects, typing down things.*

She often uses her computer for storing music and she does not play a lot of games. She plans on becoming an international commercial pilot and that is basically the reason why she decided to take IT as a subject.

- *responding to my career choice. As well it's an opportunity to challenge myself, to see what I'm capable of, how far my potential can actually go.*

- *I decided to take IT because I need the points for university to get through. So, IT offers a lot of points and it's a big bonus for me.*

Like in so many cases, it is a man who had influenced her the most to take IT and he has been motivating and supporting her ever since.

- *He's my co-pilot. ... he does a lot of this stuff and he helps me through it. He says to me: "You know what, it's going to put you in such a position where you are not going to have to go back and study all this stuff in order to do something". So, he's really helped me through it, motivated me to keep on doing it.*

She answered the following when asked if she ever had regretted her decision to take the subject IT:

- *I wouldn't say regret, but maybe have second thoughts. Yes, I think it was beginning of Gr.10, our first term and found it fairly easy, I was coping with it. Came second term and coding and I was like: "What am I doing?" So, over practicing and practicing I kind of got the hang of it and after then I was just fine with it. But there are some rough edges with me. But ja, practice helps.*

#### 4.5.3.2 THE SUBJECT IT

##### 4.5.3.2.1 Enjoyment of the subject IT

- *Who'd thought you can put enjoyable and IT in the same sentence, now I can!*

The above comment was made spontaneously by respondent 6 after pair programming was introduced and speaks for itself.

When asked specifically about their enjoyment of the subject IT, it was evident that they enjoy the subject and their responses seemed even more positive after pair programming.

R	Before pair programming	After pair programming
5	<ul style="list-style-type: none"> <li>• <i>it's a subject I really really do enjoy</i></li> <li>• <i>Yes I do, I really really like it a lot, it's just, it's complicated and I like configuring stuff and I like to think outside of the box, but it's really hard, honestly it's hard.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>I love IT. It's a really nice subject. IT, I have a passion for programming. I love practical things, I love things that are hands-on, things that have to be creative, you have to think a lot. I enjoy it, it's a nice subject.</i></li> </ul>
6	<ul style="list-style-type: none"> <li>• <i>I enjoy it. I enjoy more the practical than what I do the theory. I enjoy being on a computer, just figuring out, like solving problems and stuff like that. I enjoy sitting there and doing my own thing.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>Yes, I do very much. It's very challenging. But I try to put myself out there. ... So, I think it's really a subject that I didn't think I would choose in high school, but I'm glad I did, it's really fun.</i></li> </ul>

#### 4.5.3.2.2 The importance of the subject IT

They viewed IT as an important subject, before and after pair programming, especially for university entrance purposes and as a challenge to pupils who perform well academically.

R	Before pair programming	After pair programming
5	<ul style="list-style-type: none"> <li>• I do believe it is cause it like it really helps a lot. ... it's a challenge to people who have high maths and science marks. If they want to take a subject that's going to challenge them, they should try IT.</li> </ul>	<ul style="list-style-type: none"> <li>• It is important. It helps a lot when you go to university. Does help with your intellectual with the computer. And you start to learn how to use the computer better, you know how the programs work because you've been taught how to do programs like that.</li> </ul>
6	<ul style="list-style-type: none"> <li>• To me it's an important subject. I won't say it will to everyone else but for certain career choices I think IT becomes a big roll. Of course like becoming a pilot, I know I need the points for my IT. IT takes you a long way and it advances your brain and everything. So, I'd rate IT is a fairly important subject.</li> </ul>	<ul style="list-style-type: none"> <li>• Yes I think it's a very important subject to take especially with what I want to become</li> <li>• I know I need a certain amount of points for me to get into this and IT is boosting my marks like, goodness knows how much.</li> </ul>

#### 4.5.3.2.3 The attributes of the subject IT

Respondents 5 and 6 had a lot to say about the subject. They felt that it is a great challenge, especially the practical part of the subject. They commented that logical thinking and practice are important.

R	Before pair programming	After pair programming
5	<ul style="list-style-type: none"> <li>• The practical has to be logical</li> <li>• if they want to take a subject that's going to challenge them they should try IT.</li> <li>• programming it is really not something that's easy, you've got to click sure</li> <li>• it's complicated and I like configuring stuff and I like to think outside of the box but it's really hard, honestly it's hard.</li> <li>• better in the theory than I do in practical because it's just it's easier to remember things that you read and so on</li> </ul>	<ul style="list-style-type: none"> <li>• you have to think a lot</li> <li>• theory I can get from my own knowledge, I don't really have to be taught it. But with the programming it's something that you don't just have to get but you also have to get taught, you have to be taught the basics of how to do it.</li> </ul>
6	<ul style="list-style-type: none"> <li>• it could be an opportunity to challenge myself</li> <li>• I've realized that not many people actually go and take IT</li> </ul>	<ul style="list-style-type: none"> <li>• It's something you can challenge yourself with and you know that when you get good outcomes, you can feel proud about</li> </ul>

	<ul style="list-style-type: none"> <li>• <i>practice helps</i></li> <li>• <i>So, ja with a little bit of help, takes you quite a far way.</i></li> <li>• <i>IT takes you a long way and it advance your brain and everything</i></li> <li>• <i>if you don't have Maths, you can't take IT</i></li> <li>• <i>It's more difficult to pass than Mathematics</i></li> <li>• <i>I know it is difficult. Generally it is. It takes a lot of thinking. It takes a lot of practice.</i></li> </ul>	<p><i>yourself because it's something you weren't able to do, and you actually did it</i></p> <ul style="list-style-type: none"> <li>• <i>programming? Again that's more the challenging part</i></li> <li>• <i>I think IT can help with a lot of things not just, you know, working with the computer.</i></li> <li>• <i>studying for an IT practical isn't that easy because I mean you never know what to expect</i></li> </ul>
--	--	--

#### 4.5.3.2.4 Achievement in the subject IT

They did not feel too convinced that their marks were a reflection of their abilities, but both took responsibility in saying that they should try harder. With both of them, a more positive answer was given after pair programming had been introduced.

R	Before pair programming	After pair programming
5	<ul style="list-style-type: none"> <li>• <i>No, because I don't study hard enough</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>Sometimes. {giggle} It depends on how hard I studied for the test or how hard I actually tried.</i></li> </ul>
6	<ul style="list-style-type: none"> <li>• <i>If I push myself harder despite like of all my sports. If I pushed myself harder in a certain lifestyle, I think my marks can go up a lot more.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>I would think so yes. Because as I've grown my marks have become a lot higher and my ability has grown.</i></li> </ul>

A very definite change was detected with respondent 5 in terms of her satisfaction with her marks. She went from very negative before pair programming to very positive after pair programming. Respondent 6 was not completely satisfied, but commented that her marks had improved over time.

R	Before pair programming	After pair programming
5	<ul style="list-style-type: none"> <li>• <i>No, not at all.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>Yes, I'm very satisfied. Because I was told that I wasn't going to be able to do it, so then I went and I wanted to prove everybody wrong, and I did. I can do it.</i></li> </ul>
6	<ul style="list-style-type: none"> <li>• <i>With a bit improvement, a little more discipline in it. I think I can reach what I want to.</i></li> <li>• <i>But at this point I'm quite satisfied with my marks. I think they, I think they are okay. Getting me to where I wanted to be.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>I think I can push myself a lot more with my marks,... it can definitely improve... how I started have improved to where I am now. So, just a matter of time and grasping the concept. It takes time to get to point B {giggle}</i></li> </ul>

#### 4.5.3.3 PROGRAMMING

##### 4.5.3.3.1 Enjoyment of programming

Respondents 5 and 6 both enjoyed the programming part of IT, even before pair programming had been introduced, and preferred it to the theory, although both viewed the theory as the easier part of the subject.

R	Before pair programming	After pair programming
5	<ul style="list-style-type: none"> <li>• I do prefer the practical to the theory side but I do a lot better in the theory than I do in practical because it's just it's easier to remember things that you read and so on.</li> </ul>	<ul style="list-style-type: none"> <li>• Ja, I prefer the programming to the theory. Because I feel that like theory I can get from my own knowledge, I don't really have to be taught it. But with the programming it's something that you don't just have to get but you also have to get taught, you have to be taught the basics of how to do it.</li> </ul>
6	<ul style="list-style-type: none"> <li>• Yes I enjoy programming. ...It's like a competition for you, you have to like ... challenge yourself,</li> </ul>	<ul style="list-style-type: none"> <li>• Again that's more the challenging part, so I enjoy that, I enjoy challenging myself. And, ja I wouldn't say I preferred the theory because theory is more easier {giggle}.</li> </ul>

The respondents mentioned specific topics in programming that they find difficult, namely linking arrays and sorting. More generally, respondent 5 found it frustrating to fail in an exam to solve a problem which she knew she had mastered before. She also found lengthy pieces of programming frustrating especially once she had learned a shorter version. Respondent 6 did not like getting stuck and having to ask their teacher for assistance.

R	Before pair programming	After pair programming
5	<ul style="list-style-type: none"> <li>• I think what's frustrating is you learn something and you know for a fact that you know it, but as soon as you get into the exam, what I do is I get so like agitated with myself that I can't remember what to do.</li> <li>• the other thing is shortened versions. First you get taught the long version of how to do something and then your teacher comes the next week: "okay, this is how a simple way to do it". I hate that so much. I can not stand it{giggle}. I want the short version there and then and that's it, finish and klaar (completed).</li> </ul>	<ul style="list-style-type: none"> <li>• I don't like when you have to link the arrays together because it never ever wants to work, but otherwise it's fine, no problems at all.</li> </ul>

6	<ul style="list-style-type: none"> <li>• <i>Sorting. Sorting frustrates the daylight out of me.{giggle} I never know where to put something.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>fixing your problems. That's one thing that really gets me is that when like I write my program out and everything will be fine, the minute I want to play it I'll have one problem that I can never get rid of, ever.</i></li> <li>• <i>And I hate having to go to the teacher or going to someone above me to say: "Can you help me fix it?" It's things that you'd want to figure out for yourself.</i></li> </ul>
---	---	---

#### 4.5.3.3.2 The importance of programming

The 2 girls had thought right from the start that programming is an important skill to have, at least at school level and for opening doors in the future. One could detect from respondent 6's comments that it is a great desire of hers to acquire that skill.

R	Before pair programming	After pair programming
5	<ul style="list-style-type: none"> <li>• <i>I do think so because, programming, it makes you really know what the computer is working like and to actually see that okay, the computer is not doing everything for you. There was a person who sat there and they worked out how for this program to do it for you.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>Not necessarily. But I think maybe in the future it will be as technology starts to develop a lot more, but right now I don't know. ... But at school, ja I think it would be.</i></li> </ul>
6	<ul style="list-style-type: none"> <li>• <i>I think it is an important skill. It's a privilege to have that skill. Ja, I think it is fairly important. There's a lot you can do with it. There's so many doors that can open if you have that ability.</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>It's a lucky skill to have. Goodness, some people are born with it, some people aren't. Yes I think it's a lucky skill to have. It shows you're capable of a lot more things, you can push yourself and extend your mind into creating things like programs and stuff.</i></li> </ul>

#### 4.5.3.3.3 Skills required for programming

The girls opined that logical thinking, thinking out of the box, knowledge of the operation of computers, memory, concentration and mathematical skills are important skills required for programming. Respondent 6 also felt that personality traits, like being a motivated and steady person, are important.

R	Before pair programming	After pair programming
5	<ul style="list-style-type: none"> <li>• <i>Definitely logic and knowledge about the way computers work, and memory</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>You have to be very logic. You have to be able to think out of the box. If you have a</i></li> </ul>



		<i>problem, you are going to have to be able to like think straight there, how you're going to fix it.</i>
6	<ul style="list-style-type: none"> <li>• obviously you need to have mathematical skills</li> <li>• Understand how a computer works and usually a computer works basically with numbers</li> </ul>	<ul style="list-style-type: none"> <li>• A lot of concentration.</li> <li>• I think you have to be very motivated.</li> <li>• You have to be a very steady person. Like, you start something you know you are going to finish it.</li> </ul>

Respondent 5 was quite confident that her performance in the subject proved that she possessed the necessary skills, but again it was evident that respondent 6 desired those skills, but she thought she was progressing.

R	Before pair programming	After pair programming
5	<ul style="list-style-type: none"> <li>• I believe I do, when I open up programs and I see okay, input box. Ok, I learned how to do this, I can do this. And I do see the different components that I used in programs and I really do believe that I can do the subject.</li> </ul>	<ul style="list-style-type: none"> <li>• I should hope so{giggle} because I'm doing the subject and I'm doing quite well, so I think I do.</li> </ul>
6	<ul style="list-style-type: none"> <li>• Shoo, I always do try! I hope to think I do.</li> <li>• I won't say I've got all the skills but I'm definitely trying on them</li> </ul>	<ul style="list-style-type: none"> <li>• I think I'm growing to it. I won't say I'm a 100% there but it takes a lot to actually get yourself to sit down, work through it. ... If I don't know how to do something I usually push it aside and that's it for me.</li> </ul>

#### 4.5.3.4 A CAREER IN IT

##### 4.5.3.4.1 The importance of a career in IT

All the girls felt that a career in IT is important, because you get paid well, there is a shortage of people in that field, and there are numerous job opportunities. The advancements in technology call for more IT people in South Africa.

R	Before pair programming	After pair programming
5	<ul style="list-style-type: none"> <li>• Yes, because we are in the technological age so, computers are everywhere and it's just going to get even bigger and better ... So, computers are very, very important.</li> </ul>	<ul style="list-style-type: none"> <li>• I think so. You get paid a lot of money and nowadays a lot of things are getting taken over by computers. ... There's a lot of job opportunities out there for programming.</li> </ul>
6	<ul style="list-style-type: none"> <li>• Ja, there's a lot of money involved.{giggle} ... I know in South Africa we short a lot of them. ... especially females.</li> </ul>	<ul style="list-style-type: none"> <li>• Yes it's very. Goodness you make lots of money with being an IT technician. I mean we lack quite a few of those especially in SA. ...</li> </ul>

		<i>SA will have a better economy at the end of the day. So I think it's a very good career choice to go.</i>
--	--	--

#### 4.5.3.4.2 Future plans

Both respondents had decided on a career that involves selection, but their future studies would involve IT. Respondent 6 said right from the start that an IT career is her next option in the event of her not being selected to become a pilot and she seemed positive if that happened to be the case. Respondent 5 initially felt she would not follow a career in IT but after pair programming had been introduced, she strongly felt that it was her next option if she did not get into medical school.

R	Before pair programming	After pair programming
5	<ul style="list-style-type: none"> <li><i>No, I don't think so. I think, cause I have my mind set on what I want to become and that doesn't involve IT ... So, I think IT will just help me in what I want to become but I wouldn't do something with IT.</i></li> </ul>	<ul style="list-style-type: none"> <li><i>It's not my first choice but it definitely is my second choice. ... if I don't get in for my first choice ... then I'll definitely go into IT. Something like that maybe on the mine, when they do the programming for like, how much water has gone in and how much water has gone out. I would really like doing that. I think it would be quite nice.</i></li> </ul>
6	<ul style="list-style-type: none"> <li><i>Definitely, if again being a pilot, it does concern IT and I will be studying it further and going into university I'm going to be taking IT as well. But of course if I don't make it becoming a pilot ... I obviously want to fall back to an IT technician and I then could make a great career for myself.</i></li> </ul>	<ul style="list-style-type: none"> <li><i>Well, it's my fallback subject. I want to be an IT technician if I don't get through with my being a pilot.</i></li> <li><i>If I don't really make it in that kind of career choice then I'd love to like go into plane engineering. With plane engineering you need to know how a plane works, how to operate it using computers.</i></li> </ul>

#### 4.5.3.5 PAIR PROGRAMMING

##### 4.5.3.5.1 Pairing versus Solo

As mentioned in 4.5, these two respondents were grouped together because their response to question 18 was that they preferred to rather program on their own.

Respondent 5 admitted she would rather do her own thing, but she felt that the rest of the class would prefer to work in pairs. It is noticeable that although she preferred to work on her own, she immediately added that pair programming had helped her and improved her programming skills.

R	Own preference	Class's preference
5	<ul style="list-style-type: none"> <li>• I'm the type of person who like to be on my own. I don't like people chopping in on my ideas and stuff. So, I prefer single programming. But I must say the pair programming has helped me a lot. I'm a lot better at programming now, than I was.</li> </ul>	<ul style="list-style-type: none"> <li>• They definitely prefer pairing.</li> </ul>

Respondent 6 started off by saying, more than once, that she preferred solo programming but later on in the interview she contradicted herself by referring to others in the class who prefer solo programming in contrast to her who prefers pair programming. She claimed that the girls mostly prefer pair programming, unlike the boys who prefer solo programming.

R	Own preference	Class's preference
6	<ul style="list-style-type: none"> <li>• I enjoy sitting there and doing my own thing</li> <li>• I enjoy working alone, I thoroughly do</li> <li>• I always used to be a very individual person if we had a group class I end up doing it, sharing with all my friends and then they can give it in.</li> </ul> <p>*****</p> <ul style="list-style-type: none"> <li>• there are a few girls like myself and there are a few guys that actually, you know, enjoy (pair programming)</li> <li>• But I mean I think there are quite a few people like myself, that actually enjoy the pair programming. It's cool. I like it.</li> </ul>	<ul style="list-style-type: none"> <li>• I know a few that like working individually, quite a few. Not many of the girls that I know would work individually but I mean, most of the guys. They really like working individually, they like doing their own work.</li> </ul>

A number of comparisons between pair programming and solo programming were made by respondents 5 and 6. They claimed with solo programming they did not finish their work, they would get stuck, they often had to call for assistance, and they would not persevere.

- in the beginning (before pair programming) we would do programs where we wouldn't even finish.
- Sometimes I would get like a little bit stuck because I wouldn't have anybody there
- when I started IT I was very scared to think about, you know, sitting in class and doing IT and having to call sir the entire time
- Whereas when I was doing it by myself I'd actually just end up giving up because I didn't know how to do it, what to do.

#### 4.5.3.5.2 The enjoyment of pair programming

The word *fun* was used to describe pair programming six times by the two respondents. The following were typical comments, but especially the first quotation speaks for itself

- *Who'd thought you can put enjoyable and IT in the same sentence, now I can!*
- *It's fun ... It's very nice.*
- *they would say that ... it's more fun*
- *And then you guys making a mistake, you laugh about it, stuff like that.*
- *I thought it was fun, it was enjoyable*
- *It was a good experience*
- *I really thoroughly enjoyed it*
- *I think there are quite a few people like myself, that actually enjoy pair programming. It's cool. I like it.*

They particularly liked learning from their partner, the encouragement they received, having to consult the teacher less, helping their partner, and persevering to the end.

- *you learn so much more things and I didn't think I'd learn that from my peers.*
- *I also liked to have somebody to speak to because when you're on your own it's like a bit frustrating, it gets like upsetting that you can't do it and then the person next to you will say: "Just calm down, this is how you do it"*
- *I liked the fact that I didn't have to ask the teacher anything*
- *I mean I actually helped someone. I didn't think I'd be able to, you know, say: Oh but I did it like that, maybe that's right and it actually ends up being right*
- *So, I think that is the one thing I liked about it is that I actually sat there, myself and my peer friend and we got through it, we did it .... You feel better about yourself, more confident about how you program and stuff.*

#### 4.5.3.5.3 The advantages of pair programming

##### a) Pair learning and comprehension

The respondents thought that they learned a lot from each classmate with whom they paired. It led to improved comprehension, which would have benefitted them during exams.

- *I know a lot more now from learning from the people that I've worked with.*
- *you tend to grasp concepts and you tend to understand things better.*
- *I can now approach my exam knowing that I know how to do this, I learned from my friend*
- *not only you're sitting with this one person who now you know the knowledge they have and they know the knowledge you have, but you're actually changing with the other people in the class and actually getting knowledge from each and every individual.*
- *you learn a lot more things, you learn a lot quicker.*
- *With the pair programming, I must say I think it was a really, really good idea. Like I really myself actually started understanding more things about IT.*

#### **b) Pair trust**

They trusted their partner to fill in the gaps in their knowledge:

- *it gets like upsetting that you can't do it and then the person next to you will say: "Just calm down, this is how you do it."*
- *And then like out of the blue it's like: "Hey, I've done this before. My friend taught me how to do this, my friend showed me how to do this. I can do it".*

#### **c) Help and negotiation**

The respondents described how they worked together, helping each other to create programs. They shared knowledge and opinions and made suggestions to reach improved solutions.

- *If they get stuck they can help each other out, so it's a very good program to follow.*
- *they definitely do prefer it because it's easier and you help each other a lot.*
- *I think a good way of describing it is helping each other to better in creating a program.*
- *Getting more opinions, getting more suggestions to how to do things, you know, it helps, you can 10 to 1 find a solution.*
- *I actually helped someone. I didn't think I'd be able to*
- *It's not that bad. I shouldn't be so scared. This one knows just as much as I do, I mean I can help him, he can help me, so why not?*

#### **d) Review and debugging**

As mentioned in 4.5.3.3.1, getting stuck, fixing problems and having to ask for assistance were seen as particular frustrations with programming, but the respondents explained how they quickly fixed problems with the help of their partner or by swapping partners.

- *you come along a problem where you both don't know something because last time you guys had a problem with it and again you have another problem with it. So, then you swap partners and then you swap roles and then it works very, very well.*
- *you finish your program in that period and then when it's done there's like 3 mistakes, fix it quickly then your program's working. So, it is a lot better.*
- *Where I got stuck, one would help me, where one got stuck, I would help because I would know the answer where he didn't*
- *me and my friend actually go through it, I follow what happens, I get the problem, I know the problem, I fix the problem and I remember it.*

#### **e) Pair pressure**

The respondents told how they felt obliged to put effort into finishing the programs within the limited time allocated.

- *it makes you want to do it better*

- *I would say it's better trying to solve a problem together than doing it by yourself. Because you have, obviously the same amount of time but more effort in trying to solve it.*
- *But obviously the one motivates the other and says: "We got to go through it because we can do it. Just carry on going through it. Let's try and see if this works, try and see if that works".*

#### **f) Pair courage and confidence**

The respondents felt confident in their work and got encouragement from their partners to solve problems they might not have solved when working alone.

- *it did make me more confident in my assignments.*
- *But obviously the one motivates the other and says, you know: We got to go through it because we can do it. Just carry on going through it*
- *Truth – it boosts up your confidence so much and like, I know when I started IT I was very scared to think about sitting in class and doing IT and having to call sir the entire time.*
- *I felt more motivated to actually want to solve problems*

#### **g) Program quality**

The respondents agreed that, as pairs, they produced programs of better quality, because they learned from their partner.

- *in the beginning we would do programs where we wouldn't even finish. Where as when you're navigating and you're driving at the same time, it's going a lot faster and you're thinking a lot better.*
- *Yes, I think it's the way you try to solve the problem. For instance if I get a problem and I'm not too sure how to approach it, my friend can help me. ... the teacher would come to us and he'd ... fix the problem and then I'd not know what he did. So, when I sit there myself and me and my friend actually go through it, I follow what happens, I get the problem, I know the problem, I fix the problem and I remember it.*

#### **h) Time taken**

The respondents thought that pair programming equipped them to complete their programs in less time than it would normally have taken them if they had to work on their own.

- *in the beginning we would do programs where we wouldn't even finish. Where as when you're navigating and you're driving at the same time, it's going a lot faster and you're thinking a lot better. So, you finish your program in that period*

#### **i) Learner achievement**

Respondent 5 told how she improved in her programming skills:

- *But I must say the pair programming has helped me a lot. I'm a lot better at programming now, than I was.*

#### **j) Social benefits**

The respondents thought that pair programming gave them an opportunity to communicate and work with other people and they claimed that it is something especially girls like.

- *And I also liked to have somebody to speak to because when you're on your own it's like a bit frustrating, it gets like upsetting that you can't do it*
- *They would say it's much better because of the way you can communicate with each other.*
- *Girls they like to work with other people and they like to talk all the time*
- *I'm learning social skills in IT. I mean I can learn how to communicate not with just the computer but with my partner telling him what to do*
- *You actually start doing this, get more communication with each other, start chatting with each other and you become friends!*

#### **k) Teacher consultation**

Respondent 6 felt the decrease in teacher consultation was one of the best things about pair programming.

- *A lot less. Definitely a lot less. ... myself and my friend can sit doing the program, not having to ask the teacher anything because we helped each other through it.*
- *I liked the fact that I didn't have to ask the teacher anything. I really liked that.*
- *not having to say: Sir! all the time. So, obviously he gets a better perspective on you.*
- *I mean, whereas I called sir most of the time saying: Please help me with this, please help me with that, please explain this to me, I don't understand it. Now, just you know, basically calling sir saying: I'm finished!*

#### **4.5.3.5.4 Difficulties of pair programming**

The respondents mentioned some difficulties of pair programming namely:

- ❖ experience and skills conflict, and
- ❖ partners being absent.

They described their problems as follows:

- *Sometimes when I got paired up with a person who didn't really know their subject too well and then I'd feel like I was doing all the work.*
- *if you put someone who's really slow and doesn't understand the concept and you take someone really high that does it really well, they don't get along too well because one's rushing the other, the other one's too slow, the other one's too fast.*

- *people tend to be absent and we have a fixed amount of people in our class. I think we're 12. Meaning 2, 2, 2 through the whole entire thing. And then when one's absent, you'd sit there and you're like: "Uhhh, okay my friend would know how to do this and I don't".*

#### **4.5.3.5.5 Pair programming in the workplace**

The respondents felt that pair programming can work well in the workplace.

- *I think it would because as far as I know for IT if you want to develop like a big program, you do work in different groups and things like that. So, if we learn how to do it, ja then we are going to be able to get into the workplace better. So, I think it does work. Because you can't go and say: make Microsoft WORD, and then you, just one programmer making MS Word. It's impossible; you have to do it with pairs.*
- *Again I would say yes, IF, everyone who was put into IT knew what they were doing ... If it's coming out well at school, I mean, think how it would come out when you start working or going to university or something like that. I think it could really, really help.*

#### **4.5.3.6 ATTRACTING GIRLS TO IT**

##### **4.5.3.6.1 Interests of girls**

Both girls admitted that they do not play computer games and both of them said they liked the challenge that IT provides to them.

- *No, I don't play games. I don't enjoy computer games{giggle}, I'm not very good at them.*
- *if they want to take a subject that's going to challenge them they should try IT.*

##### **4.5.3.6.2 Reasons and solutions for the shortage of girls in IT**

The girls felt that the perception exists that the focus of IT is games and that the subject and a career in IT are masculine. The following was a typical response:

- *Well I know people tend to think IT's all gaming, IT's all doing, you know, guys stuff. Because they make it a very male type of subject or male type of career choice.*

The respondents also felt that the word that goes around is that IT is a difficult subject and not all learners can afford a computer.

- *I know with IT I was, I was very scared to take IT. ... When I was in Grade 9 going into Grade 10, I spoke to the grade in front of me and I spoke to my friends, and I was like: "What's IT like?" They said: "IT's so hard, IT's so difficult". You know. It's more difficult to pass than Mathematics, and I 'sukkel'(struggle) with my Mathematics as it is!*
- *in the beginning of the year their requirement was that we had to have our own computer, and I know a lot of people that wanted to take IT but they didn't have their own computers so they said they couldn't do it.*



Respondents 5 and 6 claimed that another reason for girls not being interested in IT, is the boys scaring them off and, according to them, the boys are actually the incompetent ones.

- *Please try and get more girls {giggle} in the class because it's just, it's hard to work with boys especially when they don't know what they're doing. It is hard.*
- *Guys like to scare the girls away, I know that. Especially in our school. They'd tell them, you know: "IT's so difficult. IT is this, IT is that".*

Some solutions to the problem of a shortage of girls in IT were given:

Like the other respondents, they felt that the subject should be promoted based on the fact that the girls in IT achieve well, IT presents a lot of opportunities for the future, it is not only for boys, and it is not as difficult as it is made out to be.

- *I think just the biggest thing is to actually show that the girls in my class they are doing better than the boys*
- *So if girls actually get the inside of it and see how many things you can do with it, I think more girls would come into it*
- *So, attracting girls: I think it's just to make them believe that they can do it and that it's not just only for boys.*
- *we need to inform them and say to them: "You know what hey, it's not a fact of being difficult, it's a fact of actually pushing yourself to an extent where you need to use this, you need to have it in order to do what you want to do".*

A teacher that encourages and motivates the girls seemed to be important to these girls.

- *What our teacher does is he goes, okay, girls can do the subject. It's not only boys and he believes in us and he thinks that we can do it and he sees and he helps us a lot with the work and if we don't understand he goes over it again and all of that stuff.*

These respondents agreed that pair programming might be a solution to the shortage and lack of persistence of girls in IT. Both of them felt that pair programming should have been used right from when they started programming.

- *I think starting it in Grade 10 with us it would have helped me a lot more, because then I would have understand more today. I thought it was a very good idea.*
- *I think it could. Girls they like to work with other people and they like to talk all the time so, I think if you do introduce pair programming with everything, it will definitely work. Right from the beginning, start with it immediately.*
- *a lot of people have left because, they just couldn't grasp it or anything. So, I think pairing and programming with IT, like generally through, all IT pairing, I think would really, really help*

Respondent 6 brought up the idea that pair programming will bring about a critical mass (see 2.5.1), in other words the more girls there are, the more girls will join.

- *I know it will attract more people. You can approach IT knowing you're not alone in it. I know when I came to IT in Grade 11 I thought: "look how little people there are!". I'd sit there by myself thinking: "if I come lowest in the class, what if I come highest in the class, or middle in the class, it's pointed out". So they know who you are. But if you're working as a team, and you're working with someone you can actually think: "It's not that bad. I shouldn't be so scared. This one knows just as much as I do, I can help him, he can help me, so why not?" I think girls would actually be a bit attracted to it. Ja, the more people that come, not just getting guys, getting girls as well.*

#### **4.5.3.6.3 Boys in the equation**

Like the other respondents, these two were quite proud of the fact that the girls in their class perform better than the boys. They did not have too many kind words for the boys in their class and described them as lazy, the ones that struggle, and the ones that might be asked to leave the subject because of poor performance.

- *with like our class there's girls who are actually doing better in IT than it is than the boys so that is more of a confident booster they said: "No, girls can't do this" but actually we can do it and actually we can even be better at it than the boys.*
- *he is going to tell half of those boys that they cannot do IT and they have to drop the subject. And he's not going to tell the girls because the girls are actually trying hard, and they are studying and they are getting somewhere.*
- *The boys, you don't want to get paired with them because they don't know their subject and they're lazy, boys are very, very lazy.*
- *they came into Grade 10 when we started taking the subject with the attitude saying: "No man, there's no girls in the year above us so we are obviously ten times better na na na na". The results came back, girls were far higher than the boys and we actually shut them down. Sir was really proud of us. The best programmer in our class is not a boy, it's a girl!*
- *Guys like to scare the girls away, I know that. Especially in our school.*
- *you get the ones, mostly the guys, quite funny, who'd actually sit there, and they're still busy with their program, they're stuck, they're lost with this.*

#### 4.5.4 MORE RESULTS

Although each respondent's two interviews were treated as a single hermeneutic unit and the results were reported in three groupings, there were a few arching themes that so strongly emerged from each respondent's data and related to the research aims, that the researcher thought that instead of reporting three times on the same theme, these themes will be discussed as a unit in the next section.

##### 4.5.4.1 Exams and tests

Each and every respondent mentioned that they found practical exams and tests very daunting and for most of them that was the answer they had given to the question of whether they had ever regretted their decision to take IT, which in effect caused them to enjoy the subject IT less.

- *during an exam it's very stressful*
- *And there was, you know, the practical exam that I went to, I just said: Ag, why did I have take the subject.*
- *when it comes to tests because sometimes they ask the questions a little bit confusing.*
- *It was after writing my June exam. I wasn't too happy with my results.*
- *When it comes to exams. ... is this actually the one for me?*
- *think what's frustrating is you learn something and you know for a fact that you know it, but as soon as you get into the exam, ... I can't remember what to do*
- *if you give me the program to do in class, I can do it, ... and as soon as I get into the exam and the marks that actually count, I cannot do the work.*
- *but as soon as I get into the exam it just, something goes wrong, always*

The fact that they had to write exams on their own after they had used pair programming in class, was not a problem for them, because they felt they had learned a lot that they could apply in the exam through pair programming.

- *I think it's prepared me a bit better for the exam.*
- *I know a lot more now from learning from the people that I've worked with. And if you don't remember what they've taught you then that's just your own stupidity. If you now (in the exam) going to have a problem, you can do it with 2 people but then you can't do it on your own, then you definitely weren't paying attention while you were doing the pair work together.*
- *I just like getting ideas from everyone else so that if I have to do like for our test, that I have to do then I can use some of those ideas and information for myself, just to help me.*
- *When I get to the exam I would be doing the same mistakes that I would have done all on my own. But now that they were there with me, we figured out the problem*

- *I think it's prepared me a bit better for the exam. I mean obviously things I learned from my friends, the things I can now do that I didn't know how to do. I can now approach my exam knowing that I know how to do this that I learned from my friend.*

#### **4.5.4.2 Technological era**

All of the girls were very much aware of the technological era we live in and the demands and opportunities it presents to them. The following were typical remarks and related strongly to their view of the importance of the subject and a career in IT:

- *The world around us is evolving and it's evolving in such a way that it involves a lot of technology, developments, advancements etc. and IT is becoming a huge part of that as well.*
- *You'll see, every work place they have computers, everyone is working on the computers.*
- *no matter what kind of business you want to go into, it's going to be involving administration or anything like that with computers.*
- *the world is revolving and if we don't have people in those categories or willing to take a chance and do something in IT there's no way we can develop further and we'll basically just deteriorate instead of progressing.*
- *we are in the technological age so, computers are everywhere and it's just going to get even bigger and better. I mean, the cell phones that are coming out nowadays, the touch screen and all of that, it's just everything is just going to improve and we're gona start having robots and stuff. So, computers are very, very important.*

#### **4.5.4.3 Different methods and solutions**

The girls felt that they used to only apply what they were taught in class, but the introduction of pair programming was beneficial to them, because it made them aware of, and taught them, different approaches and methods to solve a problem.

- *you get taught one way and you think that's the only way that you can use that coding meanwhile there's so many ways you can get it,*
- *Think of various methods in solving a problem, not just stick to one solution*
- *You don't know if there are other methods of doing it. You don't know if you can go another route to something. But when you're with someone, that person can correct your mistakes, they can help you whenever you get stuck; introduce new ways you didn't even know about.*
- *Or try different ways of solving the problem. We don't have to go the same route that has been taught to us, thinking out of the box, thinking of:"Instead of doing it this way, we can try this way. If this way doesn't work, we can go another route".*

#### 4.5.4.4 The teacher

It was clear from the interviews that these teenage girls looked up to their teacher and they valued his opinion and assistance. One in particular was thankful that their teacher introduced pair programming in their class.

- *But then sir luckily introduced this whole pair programming*
- *sir would say to my mom: She is capable of doing IT. That pushes my confidence up*
- *What our teacher does is he goes, ok, girls can do the subject. It's not only boys and he believes in us and he thinks that we can do it and he sees and he helps us a lot with the work and if we don't understand he goes over it again*
- *When sir does it with us, it like, it clicks*
- *Sir was really proud of us*
- *the programming help you so much with your friend and your confidence went up, your knowledge about it went up, not having to say: Sir! all the time. So, obviously he gets a better perspective on you.*

#### 4.5.4.5 A computer scientist

*Describe a computer scientist in terms of what the person looks like and what the person does.*

The above question relates to their view of the importance of a career in IT and was planned for both interviews to determine if pair programming would change the girls' perception of a computer scientist. The following were typical responses in the first interview:

- *A very old male, like you know with white hair and he's got glasses*
- *He's just stuck in his cubicle his whole work time*
- *I think it looks like weary men, like a white coat, with glasses, like little, like very nerdy or dorky and works like solving lots of, like a brainiac*
- *Nine times out of ten it's a male and he has glasses(giggle) and he's tall and thin and they are like computer boffins, they play games 24/7 they, are always on the computer, they know everything they repeat their maths and stuff like that. They fix computers, they program new programs, games and they spend hours and hours on thinking about what they're going to do*
- *I see him 24/7 behind a computer, coding games, making games for little kids to play.*

From the responses of the first interviews, it was gathered that most of the girls got stuck on their perception of a typical "scientist" often portrayed in the media as an old man wearing glasses and a white coat. The few who got past the "scientist" part and described a "computer" scientist were painting a picture of a nerdy male figure stuck behind his computer.

In the second interview, the question was changed to describing a *computer programmer*. The following were typical responses:

- *I think a programmer is more open to new ideas, but a scientist keeps to the way things are, like he keeps to the same methods and I think he's not open to new ideas and new changes towards.*
- *I picture a programmer much different because he thinks differently, is more open-mindedly than a scientist.*
- *He interacts with people ... he wants to make up his own ideas and make up his own plans.*
- *Always at a computer, like typing fast, with glasses and coat, ja*
- *Okay, a normal person. Someone who's very creative, not afraid to express what they think no matter what their surroundings are. Someone who's very confident and who is able to try new things and not follow a fixed routine.*
- *What I used to think actually, is a person that sits behind a computer, codes every single day, day in and day out. But now that we're learning new things it's been opened up to me that they don't actually sit behind a desk all the time, typing and typing endless hours.*
- *I would see them like a tall guy, kind of thin with like glasses{giggle}. Somebody who's like always behind their computer, they've always got their laptop with them*
- *usually they're sitting in front of the computer 24/7, programming these things. So, I think by means of looks, he'd look more of a geek {giggle}, if it's a vague way to put it, but very wealthy, earning a lot of money.*
- *I think they are going to be probably around 20 to 30 years old that actually sit down and do these programming. Basically doing games, PlayStations, X-box and stuff like that.*

The interview after pair programming showed that pair programming had changed some of the girls' perceptions of a computer scientist and programmer, but for some the male figure with glasses still stuck in their minds.

#### **4.6 SUMMARY**

The results of the qualitative investigation were reported in this chapter. In Chapter 5, the results will be discussed, conclusions will be drawn and finally, recommendations will be made.

## **Chapter 5**

### **Discussion of findings, conclusions and recommendations**

#### **5.1 SYNOPSIS OF STUDY**

The research report comprised five chapters and a synopsis of each chapter will firstly be given, followed by a discussion of the findings, conclusions and recommendations.

Chapter 1 was devoted to an orientation in terms of the shortage of girls in the IT class and the need for measures to increase enrolments in the field of technology. A further orientation was given of pair programming as an intervention that shows several promising properties for educational purposes. These gave rise to the problem statement, the research question, the aim of the research and the research design.

In Chapter 2 a literature review was given of girls and the computer environment. Attention was given to girls' attitudes towards computers and the factors that have an effect on their attitudes and involvement in the computer environment.

Chapter 3 was devoted to a literature review of pair programming. Attention was paid to the advantages of pair programming in general and to girls in particular. It was clear that pair programming held several promising benefits for the intended empirical study.

Chapter 4 contains the empirical study and describes the whole process when the six Grade 11 girls were investigated. A complete report of the results is presented.

This chapter contains a discussion of the research results emerging from the literature study and the empirical investigation in the light of the research question set for this study. Conclusions arising from the results are drawn, followed by recommendations resulting from the conclusions.

#### **5.2 DISCUSSION OF THE FINDINGS OF THIS STUDY**

The interpretation and discussion of the research results occur in light of the subquestions in Table 1.2, as derived from the research question (see 1.4).

The discussion focuses more explicitly on the empirical findings than on the literature, because the literature studied in Chapters 2 and 3 formed a foundation for data collection

and analysis in the empirical study. The literature is therefore implicitly and intrinsically part of the study and its findings.

A synopsis of the findings of the literature study is firstly given.

## **5.2.1 SYNOPSIS OF THE LITERATURE STUDY**

### **5.2.1.1 Girls and the computer environment**

In view of the growing role of technology in the world at the beginning of the 21<sup>st</sup> century, women's low and decreasing representation in IT is a major concern. Women's participation in university-level computing degrees is typically low worldwide, and as a result women's participation in computing professions is equally low (see 2.1).

Girls (especially those aged around 14) have more negative feelings about the computer than boys and enjoy working with computers less (see 2.3.1). They do not enjoy the competitive and anti-social environment of computers but rather like collaboration, completion and relevance to the real world. Girls are attracted when they recognise computing as a form of communication, a means of creative self-expression, or as a path to a helping career (see 2.2).

Girls do view computers as important and useful – but they are not necessarily interested in computers and its related subjects and careers (see 2.3.2). Another problem is the persistence of women in computer science – the ratio of women to men involved in computing shrinks dramatically from secondary school on through university and into IT careers. The scientific heart of computer science is *lost in translation* at the secondary school level, and as a result the field continues to lose the participation and interest of a broad layer of students, especially females (see 2.3.2).

A number of general factors were identified that have an effect on girls' attitudes towards computers and involvement in the computer environment. The main factors applicable to this study are age, experience, parents, the male culture of IT, and games and other software (see 2.4). A number of classroom factors that have an effect on girls' attitudes were also identified. They are peers, the teacher, teaching strategies, the IT curriculum, the role of programming in IT, and the academic achievement of girls in IT (see 2.5).

Several interventions and strategies were recommended to increase the number of girls in the IT class and pair programming is an intervention that shows promise (see 2.6).



#### **5.2.1.2 Pair programming in the IT class**

Traditional introductory programming courses generally require that learners work individually on their programming assignments. This approach teaches learners that software development is an individual activity, potentially giving learners the mistaken impression that programming is an isolating and lonely career. IT educators must create a collaborative, socially-engaging environment with clearly defined boundaries that appeals to the current generation of students and that paints a more realistic picture of the collaborative nature of professional IT careers (see 3.1).

Pair programming, whereby two programmers develop software side-by-side at one computer, seems to be an approach that will address several significant factors that limit girls' participation in IT, although the benefits associated with pair programming extend to both boys and girls (see 3.1).

The software industry has practiced pair programming with great success for years, which sparked the interest of researchers to experiment with industry's pair-programming model in the IT classroom (see 3.2).

A number of factors that influence the success of pair programming have been identified. The incorporation of the five principles of cooperative learning into pair programming is important to ensure an equal contribution of both members of the pair. The facilitator in the programming class must, when pairing the learners, pay attention to compatibility factors such as perceived skill and actual skill of the partners, and personality types (see 3.4.1).

There are benefits for the educator who incorporates pair programming into the classroom, such as a reduced workload, because the students have a peer to turn to for help, and because of peer pressure, the number of cheating cases having to be dealt with by facilitators, is greatly reduced (see 3.4).

Several advantages of pair programming to all learners, and specifically to girls, have been identified (see 3.5 and 3.7.1). The advantages which have the greatest significance for this study are:

- Increased enjoyment of programming
- Positive attitudes towards the pair-programming experience
- Greater confidence, increased interest and greater persistence in computers and its related subjects and careers

- Social and career benefits – girls realise that programming is not as solitary as they imagined it to be
- Program quality improves and program completion and submission increase
- A decrease in bugs and getting stuck, resulting in a decrease in teacher consultation and time taken
- An increase in learning and comprehension resulting in improved academic achievement
- Improved communication, motivation and focus

Some difficulties with pair programming have also been identified, such as conflict in experience, reliability and effort; personality clashes resulting in disagreements; distractions; bad communication; co-dependence and the resistance to the implementation of pair programming. However, by managing the pair-programming experience through the application of the five principles of cooperative learning, the problems can be reduced significantly (see 3.6).

A discussion of the findings arising from the empirical study follows.

## 5.2.2 DISCUSSION OF THE FINDINGS OF THE EMPIRICAL STUDY

### 5.2.2.1 Response to Subquestion 1:

**How does pair programming shape secondary school girls' experience with regard to their enjoyment of programming? (see Table 1.2)**

It was found that most of the respondents always enjoyed the programming part of IT more than the theory, but one respondent enjoyed the theory more than the programming before pair programming was introduced (see 4.5.1.3.1). Once pair programming was introduced, her view changed and she started enjoying programming more than the theory. An issue that seems somewhat contradicting is that they find the theory part of the subject easier, but they prefer the practical. Pair programming had such an impact on them that although their marks in the theory might be higher (see 4.5.1.3.1 and 4.5.3.3.1) and they find the theory easier, they still prefer the programming part.

The increased enjoyment of programming was attributed to a number of factors:

- *Improved comprehension* (see 4.5.1.3.1 and 4.5.2.5.2). The respondents all commented on how pair programming improved learning and comprehension, because each partner had knowledge and skills to offer which they then learned from each other (see 4.5.1.5.3.(a); 4.5.2.5.3.(a) and 4.5.3.5.3.(a)). The fact that they learn from each other agrees with findings of Williams and Upchurch (2001:330) and Cockburn and Williams (2000:2).
- *Different methods and solutions*. Through pair programming the girls learned that there are different methods and solutions to a problem (see 4.5.2.3.1). All of the respondents actually mentioned how they learned new methods to solve a problem from their partners, and they realised that there are different correct solutions to a problem (see 4.5.4.3). These girls' comments concurred with studies that showed that girls have far less computer experience than boys and they usually stick to what they were taught in class, not exploring further, like boys (see 2.4.3). Pair programming taught them more than the one method or solution the teacher could manage in class time.
- *Fewer errors*. The girls had fewer problems with errors once pair programming was used (see 4.5.1.3.1, 4.5.2.5.2 and 3.5.1.8). Furthermore, getting stuck and the struggle to fix errors, are factors that negatively influence the enjoyment of programming (see 4.5.1.3.1, 4.5.2.3.1 and 4.5.3.3.1), but with the help of a partner in

pair programming, they got stuck less and fixed problems quickly and effectively (see 4.5.1.5.3.(d); 4.5.2.5.3.(d), 4.5.3.5.3.(d) and 3.5.1.15).

The programs written in pairs are of a better quality because thinking skills improve and ideas can be shared (see 4.5.2.5.3.(g), 4.5.3.5.3.(g) and 3.5.1.5). Even the one girl that prefers solo-programming, pointed out in the very same sentence that her programming had improved a lot since they started using pair programming (see 4.5.3.5.3.(i)).

The girls also pointed out several of the advantages of pair programming coupled with the behaviours that support those advantages as described by Williams and Kessler (2002:21–31)(see 3.5.2).

- *Help and negotiation.* The girls worked together with their partners and helped each other to create effective programs. They shared knowledge and opinions and made suggestions to come to improved solutions (see 4.5.1.5.3.(c), 4.5.2.5.3.(c) and 4.5.3.5.3.(c)). The fact that they could help their partners was singled out as one of the most enjoyable things about pair programming (see 4.5.3.5.2).
- *Pair courage and confidence.* The girls also felt confident that the quality of their programs had improved when they were working in pairs. They described how the partners encouraged each other to solve problems they might not have attempted when working alone. Pair programming was even described as a *confidence booster* (see 4.5.1.5.3.(f), 4.5.2.5.3.(f) and 4.5.3.5.3.(f)).

*they might feel that they're not good enough to be able to be in IT. That might lack their confidence, you know, like I'm not good enough. But if you encourage that person and you help them out then I feel like they'll enjoy it more.*

Several studies had shown that girls lack confidence in their abilities and pair programming proved to be one strategy to improve their confidence (see 2.3.3.2 and 3.5.1.3).

- *Pair trust.* The girls described how they got to know and trust their partners to solve the problems. They trusted their partners to fill in the gaps in their knowledge (see 4.5.1.5.3.(b), 4.5.2.5.3.(b) and 4.5.3.5.3.(b)).
- *Pair pressure.* The girls perceived a natural pressure between the two partners, which resulted in them wanting to become involved, working hard at completing the task at hand and attempting to do it well (see 4.5.1.5.3.(e), 4.5.2.5.3.(e) and 4.5.3.5.3.(e)). The one girl said more than once that they enjoyed the fact that, instead of quickly giving up when they were working alone in the past, pair programming changed the classroom scene into a competition of the pairs (see 4.5.2.5.3.(e)).

A suppressor to the enjoyment of programming is that long sets of coding is time-consuming and becomes frustrating (see 4.5.2.3.1 and 4.5.3.3.1) to a point where the individuals want to give up and programs are not completed. When working in pairs, it is less frustrating (see 4.5.1.5.3.(g)) and time-consuming (see 4.5.2.5.3.(k), 4.5.3.5.3.(h) and 3.5.1.7) and the partners encourage and motivate each other not to give up and complete the task at hand (see 4.5.2.5.3.(j), 4.5.3.5.3.(f) and 3.5.1.6). Since girls prefer collaboration and completion in the computer environment (see 2.2), it is obvious that pair programming addresses that preference.

Another suppressor to the enjoyment of programming is having to consult the teacher when getting stuck, but with pair programming the girls had to consult their teacher far less than before (see 4.5.1.5.3.(i); 4.5.2.5.3.(i), 4.5.3.5.3.(k) and 3.5.1.10). The fact that they had to consult their teacher less was described as one of the best things about pair programming (see 4.5.3.5.2) and it resulted in the teacher getting an improved view of the girls (see 4.5.3.5.3.(k)). Girls value their teacher's opinion and teachers are often unintentionally discriminatory towards the girls in their class (see 2.5.2 and 4.5.4.4).

All of the girls found practical exams very daunting and before pair programming was introduced, they commented how exams made them consider dropping the subject. Once pair programming was introduced, they felt that they had learned so much through pair programming that writing the exam individually did not pose a threat to them (see 4.5.4.1).

Very few negative aspects of pair programming were brought up, like occasional personality clashes, conflict in effort and experience conflicts (see 3.6.1, 3.6.4 and 3.6.5). The girls mainly complained about the boys in the class who were described as incompetent and causing the few difficulties with pair programming (see 4.5.1.5.4, 4.5.2.5.4 and 4.5.3.5.4). However, the enjoyment and advantages of pair programming totally outweighed the difficulties of pair programming.

#### 5.2.2.2 Response to Subquestion 2:

**How does pair programming shape secondary school girls' experience with regard to their enjoyment of the subject IT? (see Table 1.2)**

***Who'd thought you can put enjoyable and IT in the same sentence, now I can!***

From the above quote one can come to the conclusion that pair programming changed at least one girl's perception of the enjoyment of the subject IT (see 4.5.3.2.1), but it was not only the case for this one girl. Most of the girls enjoyed the subject IT before pair programming was introduced, but once pair programming was used, an even more positive attitude was reflected. Words like *fun*, *really nice*, *I love it* were generously used once pair programming was introduced (see 4.5.1.2.1; 4.5.2.2.1 and 4.5.3.2.1).

From the literature it was seen that the pair-programming experience makes the subject more enjoyable (see 3.5.1.1). Pair programming was repeatedly described as *fun* and the word *more* ("*enjoyed it more*", "*much more fun*", "*more exciting*") was used repeatedly (see 4.5.1.5.2, 4.5.2.5.2 and 4.5.3.5.2). The two girls who claimed that they prefer solo programming, described pair programming as "It's cool. I like it", "a good experience", "it's very nice", "really thoroughly enjoyed it" and "a very good program to follow", which is a very strong indication that even the smaller proportion of a class who prefer solo programming, definitely enjoyed the pair-programming experience and saw the benefits they could gain from it (see 4.5.3.5.2).

A critically related issue to the enjoyment of a subject is the achievement in a subject (see 2.5.6). Half of the six respondents reported an improvement in their marks since pair programming had been introduced (see 4.5.1.2.4 and 4.5.3.2.4) and the other half did not report a decline or greater dissatisfaction with their marks (see 4.5.2.2.4). These findings concur with other studies reporting that pair programming improves learner achievement (see 3.5.1.12). The girls opined that pair programming should be used in IT classes because marks will improve and they reported how even the boys' marks had improved (see 4.5.2.6.3 and 4.5.1.6.3).

Secondary school girls are very sociable, they like collaboration and they have the *urge* to communicate (see 4.5.2.5.1, 4.5.3.5.3 and 4.6.3.6.2) – pair programming addresses that need according to the literature (see 2.2 and 3.5.1.17). The fact that they can communicate and interact with their friends while they are learning, was described as one of the most

enjoyable parts of the subject brought about by pair programming (4.5.1.5.3.(h), 4.5.2.5.3.(h) and 4.5.3.5.3.(j)).

Girls do not enjoy the fact that there are so few of them in the IT class (see 4.5.3.6.2). Pair programming will bring about a critical mass, in other words the more girls there are in the IT class, the more girls will join (see 2.5.1 and 4.5.3.6.2). The one girl also opined that if pair programming was introduced earlier, there would have been more girls taking the subject IT (see 4.5.2.6.2).

When the girls were asked in question 16 (see Table 4.1) what could be done to attract more girls to IT, the girls all spoke as if from one mouth that pair programming should have been used when they were introduced to programming in Grade 10 (see 4.5.1.6.2, 4.5.2.6.2 and 4.5.3.6.2). This corresponds with numerous studies that recommend pair programming for introductory programming courses (3.5.1.20). It was also recommended that pair programming be used in other secondary schools (see 4.5.2.6.2). The following quote wraps up the girls' feelings regarding attracting girls to the subject IT:

*If they know about this pair programming they would actually come because they know that someone would be there to help them along the way, they would help them with their mistakes, help them wherever they get stuck.*

### **5.2.2.3 Response to Subquestion 3:**

**How does pair programming shape secondary school girls' experience with regard to their view of the importance of programming and the subject IT? (see Table 1.2)**

As mentioned earlier (see 4.4.4), the Grade 11 girls used in the empirical investigation have already come a long way with IT. Not only have they chosen the subject, but they have also persevered for more than a year. It was therefore no surprise that their view of the importance of programming and the subject IT had not changed from before pair programming to after pair programming was used – the subject IT and the skill of programming remained important to them (see 4.5.1.2.2; 4.5.1.3.2, 4.5.2.2.2; 4.5.2.3.2; 4.5.3.2.2 and 4.5.3.3.2). These findings correspond with literature findings that girls do view computers and computer subjects as important (see 2.3.2).

However, some of them considered quitting because they found the programming difficult (see 4.5.2.1.1; 4.5.2.1.2 and 4.5.3.1.2), but once pair programming was introduced, it made them persist (see 4.5.2.5.3.(m)). The girls all agreed that pair programming would help girls in the IT class to persist and would therefore cause less drop-outs (see 4.5.2.6.2 and 4.5.3.6.2). This phenomenon of girls quitting is referred to as the *pipeline shrinkage problem*

(see 2.3.2), but studies on pair programming also found that the use of pair programming in the IT class causes students that might otherwise have dropped the course, to complete the course, and consequently pass it (see 3.5.1.4).

#### **5.2.2.4 Response to Subquestion 4:**

##### **How does pair programming shape secondary school girls' experience with regard to their view of the importance of a career in IT? (see Table 1.2)**

A career in IT was seen as important before and after pair programming for reasons such as financial gains, the shortage of people (especially women) in IT in South Africa and the rapid advancements in the field of technology (see 4.5.1.4.1, 4.5.2.4.1 and 4.5.3.4.1). This corresponds with the findings of Seymour *et al.* (2005:103) in South Africa that showed that secondary school learners do not know what a degree in IT entails, but they are convinced that remuneration and occupational benefits are positive factors in the pursuing of an IT career (see 2.3.2).

Although the respondents felt a career in IT is important, it was an obvious concern before pair programming was introduced, that a career in IT is synonymous with being stuck in an office, working on the computer with no social communication and interaction (see 4.5.1.4.2, 4.5.2.4.1 and 4.5.2.6.1). Girls are attracted when they recognise computing as a form of communication, a means of creative self-expression, or as a path to a helping occupation since they do not like the competitive and anti-social environment of computers (see 2.2). Once pair programming was introduced, that perception of an IT career being a solitary occupation, had changed (see 4.5.2.4.2). The girls commented after pair programming was introduced that when they follow a career in IT, they will work in a team and get to know other people (4.5.2.4.1 and 4.5.2.5.3.(h)). Pair programming in the IT class had prepared them for a career in IT, because they had learned social skills (to communicate and work with other people) in the process of having a partner and swapping partners (4.5.2.5.3.(h) and 4.5.3.5.3.(j)). This corresponds with findings from the literature (see 3.5.1.23).

The girls unanimously agreed that pair programming would work well in the workplace, mentioning reasons such as 'more minds are better than one' and 'the workers' needs for social interaction are met' (see 4.5.1.5.5, 4.5.2.5.5 and 4.5.3.5.5). They noticed the success of pair programming in the IT class and predicted success in the workplace as well (see 4.5.3.5.5). One girl even felt that the involvement of women would make pair programming more effective in the workplace, which says a lot about the preference of girls for an IT career (see 4.5.2.5.5). The one girl even felt that programming in the workplace is impossible without pairing:



*you can't go and say: "make (create) Microsoft Word", and then you, just one programmer making(creating) MS Word. It's impossible; you have to do it with pairs.*

The girls' view of a computer scientist, before pair programming was introduced, had the characteristics typical of a scientist portrayed in the media – an old man with a white coat wearing glasses (see 2.4.1.2 and 2.4.1.3). After pair programming was introduced and they were asked to describe a computer programmer, a definite change in view was detected with some of the girls. A computer programmer is a more progressive person, interacting with people, creative, open-minded and wealthy, and not necessarily stuck behind a computer. However, for some girls the male figure with glasses still stuck in their minds (see 4.5.4.5).

A strong indicator that pair programming had changed the girls' perception towards the importance of a career in IT, is their future plans regarding a career in IT. One girl did not even consider a career in IT before pair programming was introduced. Once pair programming was used in the IT class, she suddenly reported how she considers a career in IT in the event of her not being selected for her first career option (see 4.5.3.4.2). Another girl planned a career in IT before pair programming was introduced, but the concern of being stuck behind a computer was quite evident. Once pair programming had been introduced, a greater enthusiasm about her career in IT was detectable. She explained how she came to the realisation that IT offers a variety of career opportunities and does not necessarily entail being stuck behind a desk (see 4.5.2.4.2). Students value the practical applications of social interaction offered by pair programming (see 3.5.1.23).

### **5.3 CONCLUSIONS ARISING FROM THIS STUDY**

The following conclusions can be made in answering the research questions based on the discussion of the findings of the research done on the 6 girls:

*Subquestion 1* Pair programming positively shaped these secondary school girls' enjoyment of programming.

*Subquestion 2* Pair programming shaped the girls' enjoyment of the subject IT to a *more fun* experience – even for the girls who prefer solo programming.

*Subquestion 3* The girls thought programming and the subject IT to be important, but pair programming shaped the girls' view of the importance of programming and the subject IT in the sense that they persisted and did not quit.

*Subquestion 4* The girls did think that a career in IT is important, but pair programming shaped their view of the importance of a career in IT in the sense that it

changed their view of IT being a solitary occupation and it caused them to more positively consider a career in IT.

#### 5.4 RECOMMENDATIONS ARISING FROM THIS STUDY

##### ***Pair programming we want!*** (see 4.5.2.5.2)

The main recommendation is voiced in the afore-mentioned quote uttered by one of the girls in the empirical study.

Several reasons were given to believe that pair programming has the potential to increase girls' computer interest and consequently increase their participation and persistence in IT. Pair programming increases the enjoyment of programming and of the subject IT. Furthermore, pair programming increases girls' views of the importance of a career in IT and it make them persist. Pair programming challenges the belief that computing is a solitary activity and answers to the need of girls for a more social class environment.

The IT teacher should implement pair programming right at the beginning of a programming course. When implemented correctly, not only the learners, especially the girls, benefit from pair programming, but also the IT teacher.

It is important to bear in mind that pair programming does not simply entail making learners write programs in pairs. The teacher must facilitate and control the pair-programming experience and continuously apply the five principles of cooperative learning.

IT teachers are often unaware of what causes girls to enjoy their subject and what chases them away. IT teachers should familiarise themselves with the preferences of girls. Pair programming is an untapped resource worth considering in any programming class, but especially to attract more girls to IT – and retain them.

The recommendations are concluded with the following quote of respondent 4:

*But now that the pair programming is introduced I'm like: there's still hope because I'm learning new things, someone is helping me on my mistakes, I'm not actually alone. They're helping me along the way, holding my hand, you know, just introducing me to new ways of actually doing it (see 4.5.2.5.3.(f)).*

## **5.5 SHORTCOMINGS IN THE EMPIRICAL STUDY**

A few shortcomings were identified in the empirical study. These shortcomings were:

- The researcher was never present when the girls worked in pairs. In a way it was good in terms of objectivity, because the girls did not even realise that this study actually caused pair programming to be implemented in their class – they thought it was their teacher's initiative. On the other hand, the observation of the girls while working in pairs could have added another dimension to the study, especially in terms of their enjoyment of programming. One problem is that if an outsider observes a programming class, the learners often do not act naturally. In addition, the practical feasibility of observing the girls would have been a problem, because it would have required that the girls be observed prior to the introduction of pair programming too.
- The researcher did not have any control over whether the IT class teacher correctly implemented pair programming in his class. However, the teacher had a 2-hour training session on the implementation of pair programming (see 4.4.4). The teacher was also supplied with a manual on the implementation of pair programming and a poster on the pair-programming rules for secondary learners was given to him (see 3.4.2).
- Although this group was the biggest group of Grade 11 girls in the North-West Province that could participate, the population of only 6 girls from one school with a specific teacher, is not necessarily representative of all Grade 11 girls in IT classes in South Africa.

## **5.6 RECOMMENDATIONS FOR FURTHER RESEARCH**

- A similar study on more girls than the six in this study
- A quantitative study on a sample of IT girls with the same research questions can be very helpful for generalisation purposes
- A comparative study to compare girls' and boys' enjoyment of programming, pair programming and the subject IT can shed some light on concurrences and differences between the two genders in the IT class
- A similar investigation can be done, but with a control group of girls not having used pair programming to compare the enjoyment and achievement of the girls in the study group with the control group
- An investigation into the impact of a well-planned and aggressive campaign to promote IT and inform girls about the subject IT; a model for such a campaign can also be developed

- An investigation into the experiences of teachers regarding the implementation of pair programming in the IT class
- A cross-cultural comparative study to compare girls from different races' enjoyment of programming, pair programming and the subject IT.

## 5.7 FINAL REMARKS

Although this study did not focus on the teacher in the IT class, it is worth mentioning that the teacher reported that he noticed such great benefits from using pair programming in his Grade 11 class, that he quickly implemented it in his Grade 10 class as well.

It was so evident that the term *computer scientist* has such a negative connotation, that even naming the subject and careers in IT should receive careful consideration.

It is hoped that the research done in this study and further research that might develop from this study, will make a difference to the number of girls taking IT as a subject and their persistence in the subject. It is also hoped that this study will convince IT teachers to implement pair programming in their classes, not only to benefit the girls in their class, but all the learners. May the implementation of pair programming make a contribution to fulfilling at least one girl's dream:

*It will be like one of my dreams to be able to show the men that I am capable of doing this and showing that independence. When you walk into a room and they can say: "Yo, that's an IT woman, she knows what she's doing".*

## Bibliography

AAUW (American Association of University Women) educational foundation: commission on technology, gender and teacher education. 1998. Gender gaps: where schools still fail our children. Washington, WA.

AAUW (American Association of University Women) educational foundation: commission on technology, gender and teacher education. 2000. Tech-Savvy: educating girls in the new computer age. Washington, WA.

AAUW (American Association of University Women) educational foundation: commission on technology, gender and teacher education. 2004. Under the microscope: a decade of gender equity projects in the sciences. Washington, WA.

ACM-W (Association for Computing Machinery – committee on women in computing). 2006. <http://www.acm.org/women> Date of access: 30 Jun. 2008.

ALLY, M., DARROCH, F. & TOLEMAN, M. 2005. A framework for understanding the factors influencing pair programming success. (In Proceedings of the 6th International Conference of XP held in Sheffield, UK from June 18 to 23, 2005. p.1–10.)

ANDERSON, A., BEATTIE, R. & BECK, K. 1998. Chrysler goes to "Extremes". *Distributed computing*:24–28, Oct.

ARSLAN, B. 2003. Dealing with difficulties in pair and group work activities. <http://www.metu.edu.tr/home/wwwmld/newsletter3.0/Classroom%20Tips/Berna%20Arslan.htm> Date of access: 21 Feb. 2008.

ATLI, G. 2006. Critical personality traits in successful pair programming. Bowling Green State University. (Thesis – M.Sc.)

BALCITA, A.M., CARVER, D.L. & SOFFA, M.L. 2002. Shortchanging the future of information technology: the untapped resource. *SIGCSE Bulletin*, 34(2):32–35, Jun.

BALOĞLU, M. & ÇEVİK, V. 2008. Multivariate effects of gender, ownership, and the frequency of use on computer anxiety among high school students. *Computers in human behavior*, 24(6):2639–2648, Sep.

BECK, K. 2000. *Extreme programming explained: embrace change*. Reading, MA.: Addison-Wesley. <http://books.google.co.za/books?id=G8EL4H4vf7UC&printsec=frontcover&dq=extreme+programming+explained+inauthor:Beck> Date of access: 6 Aug. 2008.

BECKERS, J., WICHERTS, J. & SCHMIDT, H. 2007. Computer anxiety: “trait” or “state”? *Computers in human behavior*, 23:2851–2862.

BEGEL, A. & NAGAPPAN, N. 2008. Pair programming: what's in it for me? (*In Proceedings of the Second ACM-IEEE International Symposium, ESEM'08 held in Kaiserslautern, Germany on 9 and 10 October, 2008. p.120–128.*)

BENNETT, D.T., BRUNNER, C., MCDERMOTT, M. & GREENE, L. 2006. Designing for diversity: investigating electronic games as pathways for girls into information technology professions. (*In The National Science Foundation's ITWF & ITR/EDF principal investigator conference. Hosted by Department of Computer Science North Carolina State University, April 2–4 2006 in Raleigh, NC. p. 65–68.*)

BERENSON, S.B., SLATEN, K.M., WILLIAMS, L. & HO, C. 2004. Voices of women in a software engineering course: reflections on collaboration. *ACM journal on educational resources in computing*, 4:11–35, Mar. [ftp://ftp.ncsu.edu/pub/unity/lockers/ftp/csc\\_anon/tech/2004/TR-2004-31.pdf](ftp://ftp.ncsu.edu/pub/unity/lockers/ftp/csc_anon/tech/2004/TR-2004-31.pdf) Date of access: 26 May 2008.

BEVAN, J., WERNER, L. & MCDOWELL, C. 2002. Guidelines for the use of pair programming in a freshman programming class. (*In Proceedings of the 15th Conference on Software Engineering, CSEET'02 held in Kentucky. p. 100–107.*) <http://doi.ieeecomputersociety.org/10.1109/CSEE.2002.995202> Date of access: 21 Feb. 2008.

BEYER, S. 2006. Women in computer science or management information systems majors: results of three longitudinal studies. (*In The National Science Foundation's ITWF & ITR/EDF principal investigator conference. Hosted by Department of Computer Science North Carolina State University 2 to 4 April 2006 in Raleigh, NC. p. 12–15.*)

BHARGAVA, A., KIROVA-PETROVNA, A. & MCNAIR, S. 2002. Gender bias in computer software programs: a checklist for teachers. *Information technology in childhood education annual*, 2002(1):205–218.

BIPP, T., LEPPER, A. & SCHMEDDING, D. 2008. Pair programming in software development teams – an empirical study of its benefits. *Information & software technology*, 50(3):231–240, Feb.

BISHOP-CLARK, C., COURTE, J. & HOWARD, E.V. 2006. Programming in pairs with Alice to improve confidence, enjoyment, and achievement. *Journal of educational computing research*, 34(2):213–228.

BLUM, L. 2001. Expanding the pipeline: transforming the culture of computing at Carnegie Mellon. *Computing research news*, 13(5):2–9, Nov.

BLUM, L. & FRIEZE, C. 2005. In a more balanced computer science environment, similarity is the difference and computer science is the winner. *Computing research news*, 17(3), May. <http://www.cra.org/CRN/articles/may05/blum.frieze.html> Date of access: 5 Aug. 2008.

BOVÉE, C., VOOGT, J. & MEELISSEN, M. 2007. Computer attitudes of primary and secondary students in South Africa. *Computers in human behavior*, 23:1762–1776.

BOYD, J. 2000. Firms work to keep women. *Internetweek*, 839:90, Nov.

BRAUGHT, G., EBY, L.M. & WAHLS, T. 2008. The effects of pair-programming on individual programming skill. (*In* Proceedings of the 39th SIGCSE technical symposium on computer science education SIGCSE'08, held in Portland, Oregon, USA, March 12–15. p. 200–204.)

BRYSON, M. & DE CASTELL, S. 1998. New technologies and the cultural ecology of primary schooling: imagining teachers as Luddites in/deed. *Educational policy*, 12(5):542–567.

CAMP, T. 1997. The incredible shrinking pipeline. *Communications of the ACM*, 40(10):103–110, Oct.

CANADA, K & BRUSCA, F. 1991. The technological gender gap: evidence and recommendations for educators and computer-based instruction designers. *Educational technology research and development*, 39(2):43–51.

CARLSON, S. 2006. Colleges work to attract and support women in technology majors. *The Chronicle: Information Technology*, 52(19):235. <http://chronicle.com> Date of access: 5 Jun. 2006.

CHAN, V., STAFFORD, K., KLAWE, M. & CHEN, G. 2000. Gender differences in Vancouver secondary students' interests related to information technology careers. Canadian Coalition of Women in Engineering, Science, Trades, and Technology. <http://www.mun.ca/cwse/Chan.Vania.pdf> Date of access: 29 Jul. 2008.

CHEN, M. 1986. Gender and computers: the beneficial effects of experience on attitudes. *Journal of educational computing*, 2(3):265–281.

CHOBOKOANE, N & BUDLENDER, D. 2002. A day in the life of a South African teenager. <http://www.statssa.gov.za/publications/TechPaperTeenagers/TechPaperTeenagers.pdf> Date of access: 29 Jul. 2008.

CHOI, K.S., DEEK, F.P. & IM, I. 2008. Exploring the underlying aspects of pair programming: the impact of personality. *Information & software technology*, 50(11):1114–1126, Oct.

CHOU, C. & TSAI, M. 2007. Gender differences in Taiwan high school students' computer game playing. *Computers in human behavior*, 23(1):812–824, Jan.

CHRISTENSEN, R. & KNEZEK, G. 2000. Internal consistency reliabilities for 14 computer attitude scales. *Journal of technology and teacher education*, 8(4):327–336.

CHRISTENSEN, R., KNEZEK, G. & OVERALL, T. 2005. Transition points for the gender gap in computer enjoyment. *Journal of research on technology in education*, 38(1):23–37, Fall.

CLARKE, M.C. & FINNIE, G.R. 1998. Changes in entry-level university students' attitudes to computers for 1985–1997. *South African computer journal*, 21:26–32, Aug.



CLIBURN, D. 2003. Experiences with pair programming at a small college. *Journal of computing in small colleges*, 19(1):20–29.

COCKBURN, A. & WILLIAMS, L. 2000. The costs and benefits of pair programming. (In Proceedings of the first international conference on eXtreme programming and flexible processes in software engineering, XP2000 held in Cagliari, Sardinia, Italy. p. 1–11.)

COHOON, J. MCGRATH. 2001. Toward improving female retention in the computer science major. *Communications of the ACM*, 44(5):108–114.

COLLEY, A. & COMBER, C. 2003. Age and gender differences in computer use and attitudes among secondary school students: what has changed? *Educational research*, 45(2):155–165, Summer.

COLLIS, B.A. 1985. Psychosocial implications of sex differences in attitudes towards computers: results of a survey. *International journal of women's studies*, 8(3):207–213.

COLLIS, B.A. & WILLIAMS, R.L. 1987. Cross-cultural comparison of gender differences in adolescents' attitudes toward computer and selected school subjects. *Journal of educational research*, 81(1):17–27, Sep/Oct.

COMBER, C., COLLEY, A., HARGREAVES, D.J. & DORN, L. 1997. The effects of age, gender and computer experience upon computer attitudes. *Educational research*, 39(2):123–133, Summer.

DAVIES, R. 2006. Pair programming explained. *Overload*, 73:4–6, Jun.

DENNER, J., WERNER, L., BEAN, S. & CAMPE, S. 2005. The girls creating games program: strategies for engaging middle-school girls in information technology. *Frontiers*, 26(1):90–98.

DEPARTMENT OF EDUCATION **see** SOUTH AFRICA. Department of Education.

DEPARTMENT OF SCIENCE AND TECHNOLOGY **see** SOUTH AFRICA. Department of Science and Technology.

- DRYBURGH, H. 2000. Under-representation of girls and women in computer science: classification of 1990s research. *Journal of educational computing research*, 23(2):181–202.
- DURNDELL, A., GLISSOV, P. & SIANN, G. 1995. Gender and computing: persisting differences. *Educational research*, 37(3):219–227.
- ELKJAER, B. 1992. Girls and information technology in Denmark: an account of a socially constructed problem. *Gender and education*, 4(1/2):25–40.
- FISHER, A., MARGOLIS, J. & MILLER, F. 1997. Undergraduate women in computer science: experience, motivation and culture. (In Paper presented at the SIGCSE Technical Symposium. San Jose, CA. p. 106–110.)
- FLOR, N.V. 2006. Globally distributed software development and pair programming. *Communications of the ACM*, 49(10):57–58, Oct.
- FOX, R. 2000. Girls and computers. *Communications of the ACM*, 43(7):10, Jul.
- FRIEZE, C. 2007. The critical role of culture and environment as determinants of women's participation in computer science. Pittsburgh: Carnegie Mellon University. (Thesis – D.Phil.)
- GALPIN, V. 1992. Gender and computer science education. Johannesburg: University of the Witwatersrand (Report – M.Sc.)
- GALPIN, V. 2002. Women in computing around the world. *ACM SIGCSE Bulletin*, 34(2):94–100, Jun.
- GALPIN, V. 2004. A South African perspective on women in computing professions. (In Martin, U. & Peltu, M., eds. Women in computing professions: will the Internet make a difference? Position papers for an OII policy forum. p. 13–15.)
- GALPIN, V. & SANDERS, I. 1993. Gender imbalances in computer science at the University of the Witwatersrand. *ACM SIGCSE Bulletin*, 25(4):2–4, Dec.

GALPIN, V., SANDERS, I., TURNER, H & VENTER, B. 2003. Gender and educational background and their effect on computer self-efficacy and perceptions. Technical Report TR-Wits-CS-2003-0, School of Computer Science, University of the Witwatersrand, Apr. <http://www.cs.wits.ac.za/> Date of access: 27 Jul. 2008.

GARCIA, O.N. & GILES, R. 2000. Research foundations on successful participation of underrepresented minorities in information technology. Final report from a cyber conference. <http://www.cise.nsf.gov/itminorities.html> Date of access: 20 Aug. 2008.

GOODE, J., ESTRELLA, R. & MARGOLIS, J. 2006. Lost in translation: Gender and high school computer science. (In Aspray, W. & Cohoon, J., eds. *Women in IT: reasons on the under-representation*:89–114. Cambridge, MA.: MIT Press)

GORRIZ, Z. & MEDINA, C. 2000. Engaging girls with computers through software games. *Communications of the ACM*, 43(1):42–49, Jan.

GORSKI, P.C. 2002. How computers still cheat girls. *Education digest*, 68(3):23–27, Nov.

GOVENDER, I. 2006. Learning to program, learning to teach programming: pre- and in-service teachers' experiences of an object-oriented language. Pretoria: UNISA. (Thesis – D.Phil)

GUARDIAN. 2008. A-level results 2008. <http://www.guardian.co.uk/education/table/2008/aug/14/alevels.schooltables> Date of access: 25 Aug. 2008.

GUNN, C. 2003. Dominant or different? Gender issues in computer supported learning. *Journal of asynchronous learning networks*, 7(1):14–30, Feb.

GÜRER, D & CAMP, T. 1998. Investigating the incredible shrinking pipeline for women in computer science. <http://women.acm.org/documents/finalreport.pdf?searchterm=discovery> Date of access: 19 Aug. 2008.

GÜRER, D & CAMP, T. 2002. An ACM-W literature review on women in computing. *ACM SIGCSE Bulletin*, 34(2):121–127.

HAHN, J.H. 2007. Paarassessering teenoor individuele assessering in rekenaarprogrammering. Potchefstroom: NWU. (Verhandeling – M.Ed.)

HANKS, B. 2005. Empirical studies of distributed pair programming. Santa Cruz: University of California. (Thesis – D.Phil.)

HANKS, B. 2008. Problems encountered by novice pair programmers. *Journal on educational resources in computing*, 7(4)art 2:1–13, Jan.

HANKS, B., MCDOWELL, C., DRAPER, D. & KRNJAJIC, M. 2004. Program quality with pair programming in CS1. (*In Proceedings of the 9<sup>th</sup> Annual Conference on Innovation and Technology in Computer Science Education ITiCSE'04 held in Leeds, United Kingdom on 28, 29 and 30 June 2004. p. 176–180.*)

HENSON, V. 2002. How to encourage women in Linux. <http://tldp.org/HOWTO/Encourage-Women-Linux-HOWTO/x106.html> Date of access: 5 Jun. 2006.

HERBERT, T.S. 2000. Women role models in Computer Science at the University of the Witwatersrand. Masters research report, Technical Report TRWitsCS200016, Johannesburg: Department of Computer Science, University of the Witwatersrand, Oct. <http://www.cs.wits.ac.za/~sawic/ref.html#journal> Date of access: 6 Mar. 2006.

HO, C., SLATEN, K., WILLIAMS, L. & BERENSON, S. 2004. Examining the impact of pair programming on female students. NCSU CSC Technical Report [ftp://ftp.ncsu.edu/pub/unity/lockers/ftp/csc\\_anon/tech/2004/TR-2004-20.pdf](ftp://ftp.ncsu.edu/pub/unity/lockers/ftp/csc_anon/tech/2004/TR-2004-20.pdf) Date of access: 6 Feb. 2008.

HOWARD, E.V. 2006. Attitudes on using pair-programming. *Journal of educational technology systems*, 35(1):89–103.

HUFF, C. 2002. Gender, software design, and occupational equity. *SIGCSE Bulletin*, 34(2):112–115.

HUHLWANE, H. 2006. South African women in computing. Statistics. <http://www.gaon.net/vashti/sawic/stat.html> Date of access: 5 Aug. 2008.

HULKKO, H. & ABRAHAMSSON, P. 2005. A multiple case study on the impact of pair programming on product quality. (*In Proceedings of the 27<sup>th</sup> Conference on Software Engineering ICSE'05 held in St. Louis, Missouri, USA on 15 to 21 May 2005. p. 495–504.*)

JACOBSON, N. & SCHAEFER, S.K. 2008. Pair programming in CS1: overcoming objections to its adoption. *Inroads — SIGCSE Bulletin*, 40(2):93–96, Jun.

JAMES, T., LEINONEN, E., SMITH, R. & HAATAJA, M.L. 2006. A manual of possible interventions to improve the situation of women in high-level ICTs in South Africa. Meraka Institute, CSIR, Pretoria, Jul. <http://women-in-ict.meraka.csir.co.za> Date of access: 5 Aug. 2008.

JENSEN, R.W. 2005. A pair programming experience. *Overload*, 65:22–25, Feb.

JOHNSON, D.W. & JOHNSON, R.T. 2009. *Joining Together: group theory and group skills*. 10th ed. Upper Saddle River, N.J.: Pearson.

KAISER, J. 2000. Getting girls to like computers. *Science now*, 4:3, Apr.

KATIRA, N., WILLIAMS, L. & OSBORNE, J. 2005. Towards increasing the compatibility of student pair programmers. (*In Proceedings of the 27<sup>th</sup> International Conference on Software Engineering ICSE'05 held from 15 to 21 May 2005.*)

KATIRA, N., WILLIAMS, L., WIEBE, E., MILLER, C., BALIK, S. & GEHRINGER, E. 2004. On understanding compatibility of student pair programmers. (*In Proceedings of the 35<sup>th</sup> SIGCSE technical symposium on computer science education, SIGCSE'04 held in Norfolk, Virginia, USA from 3 to 7 March 2004. p. 7–11.*)

KEKELIS, L.S., ANCHETA, R.W. & HEBER, E. 2005a. Hurdles in the pipeline: girls and technology careers. *Frontiers: a journal of women studies*, 26(1):99–109.

KEKELIS, L.S., ANCHETA, R. W., HEBER, E. & COUNTRYMAN, J. 2005b. Role models make a difference: a recipe for success. *AWIS Magazine*, 34(3):17–24.

KING, J., BOND, T. & BLANDFORD, S. 2002. An investigation of computer anxiety by gender and grade. *Computers in human behavior*, 18(1):69–84, Jan.

KLAWE, M. 2001. Refreshing the nerds. *Communications of the ACM*, 44(7):67–68, Jul.

KLAWE, M. 2002. Girls, boys, and computers. *Inroads*, 34(2):16–17, Jun.

KNEZEK, G. & CHRISTENSEN, R. 1996. Validating the computer attitude questionnaire. (In Paper presented at the annual meeting of the Southwest Educational Research Association held in New Orleans, LA, January 1996.)

KNUPFER, N. 1997. Gendered by design. *Educational technology*, 37(2):31–37.

KNUPFER, N. 1998. Gender divisions across technology advertisements and the WWW: implications for educational equity. *Theory into practice*, 37(1):54–63.

KOOHANG, A.A. 1989. A study of attitudes towards computers: anxiety, confidence, liking and perception of usefulness. *Journal of research on computing in education*, 22:214–288.

LEE, A.C.K. 2003. Undergraduate students' gender differences in IT skills and attitudes. *Journal of computer assisted learning*. 19:488–500.

LEEDY, P.D. & ORMROD, J.E. 2005. Practical research. Planning and design. 8th ed. Upper Saddle River, NJ: Prentice-Hall.

LEVINE, T. & DONITSA-SCHMIDT, S. 1998. Computer use, confidence, attitudes and knowledge: a causal analysis. *Computers in human behavior*, 14(1):125–146.

LITTLETON, K. & HOYLES, C. 2002. The gendering of information technology. (In Yelland, N. & Rubin, A., eds. Ghosts in the machine: women's voices in research with technology. New York: Peter Lang Publishing. p. 3–32.)

LUI, K.M., CHAN, K.C. & NOSEK, J.T. 2008. The effect of pairs in program design tasks. *IEEE transactions on software engineering*, 34(2):197–211, Mar/Apr.

MAKRAKIS, V. 1993. Gender and computing in schools in Japan: the "we can, I can't" paradox. *Computers and education*, 20(2):191–198.

MARGOLIS, J. & FISHER, A. 2000. Women in computer science: closing the gender gap in higher education. Final report to the Sloan Foundation. <http://www.cs.cmu.edu/~gendergap>  
Date of access: 28 Sep. 2008.

MARGOLIS, J. & FISHER, A. 2002. Unlocking the clubhouse: women in computing. Cambridge, MA: MIT Press.

MARGOLIS, J., FISHER, A. & MILLER, F. 1999. Caring about connection: gender and computing. *IEEE technology and society magazine*, 18(4):13–20. <http://www.cs.cmu.edu/afs/cs.cmu.edu/project/gendergap/www/papers/IEEE99.pdf> Date of access: 29 Jul. 2008.

MARGOLIS, J., FISHER, A. & MILLER, F. 2000a. The anatomy of interest: women in undergraduate computer science. *Women's studies quarterly*, 28(1/2):104–127. [http://www.journalism.wisc.edu/~gdowney/courses/lis810-labor-2004-09/PDF/Margolis%20J%20et%20al%20\(WSQ\)%202000.pdf](http://www.journalism.wisc.edu/~gdowney/courses/lis810-labor-2004-09/PDF/Margolis%20J%20et%20al%20(WSQ)%202000.pdf) Date of access: 29 Jul. 2008.

MARGOLIS, J., FISHER, A. & MILLER, F. 2000b. Living among the "programming gods": the nexus of confidence and interest for undergraduate women in computer science. <http://www.cs.cmu.edu/~gendergap> Date of access: 28 Sep. 2008.

MARTINEAU, R. 1998. Science and technology: where are the women? *Agenda*, 38:12–18.

MCDOWELL, C., HANKS, B. & WERNER, L. 2003a. Experimenting with pair programming in the classroom. (*In* Proceedings of the 8<sup>th</sup> Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE'03 held on June 30 to July 2 2003. *ACM SIGCSE Bulletin*, 35(3):60–64.)

MCDOWELL, C., WERNER, L., BULLOCK, H.E. & FERNALD, J. 2002. The effects of pair programming on performance in an introductory programming course. (*In* Proceedings of the 33rd SIGCSE technical symposium on computer science education, SIGCSE'02. p. 38–42.)

MCDOWELL, C., WERNER, L., BULLOCK, H.E. & FERNALD, J. 2003b. The impact of pair programming on student performance, perception and persistence. (*In* Proceedings of the 25<sup>th</sup> international conference on software engineering, ICSE'03 held in Portland, Oregon, USA. p. 602–607.)

MCDOWELL, C., WERNER, L., BULLOCK, H.E. & FERNALD, J. 2006. Pair programming improves student retention, confidence and program quality. *Communications of the ACM*, 49(8):90–95.

MCILROY, D., BUNTING, B., TIERNEY, K. & GORDON, M. 2001. The relation of gender and background experience to self-reported computing anxieties and cognitions. *Computers in human behaviour*, 17(1):21–33.

MCKINNEY, V.R., WILSON, D.D., BROOKS, N., O'LEARY-KELLY, A. & HARDGRAVE, B. 2008. Women and men in the IT profession. *Communications of the ACM*, 51(2):81–84, Feb.

MCMILLAN, J.H. 2000. Educational research: fundamentals for the consumer. 3<sup>rd</sup> ed. USA: Addison Wesley Longman.

MELNIK, G. & MAURER, F. 2003. Introducing agile methods in learning environments: lessons learned. <http://sern.ucalgary.ca/~milos/papers/2003/MelnikMaurer.pdf> Date of access: 29 Apr. 2008.

MENDELS, P. 2000. Technology is still a male bastion. [http://www.businessweek.com/careers/content/jun2000/ca20000626\\_426.htm](http://www.businessweek.com/careers/content/jun2000/ca20000626_426.htm) Date of access: 5 Aug. 2008.

MENTZ, E. & GOOSEN, L. 2007. Are groups working in the Information Technology class? *South African journal of education*, 27(2):329–343.

MENTZ, E., VAN DER WALT, J. L. & GOOSEN, L. 2008. The effect of incorporating cooperative learning principles in pair programming for student teachers. *Computer science education*, 18(4):247–260, Dec.

MERRIAM, S.B. 1998. Qualitative research and case study applications in education. 2<sup>nd</sup> ed. San Francisco, CA: Jossey-Bass.

MILLER, L., CHAIKA, M. & GROPE, L. 1996. Girls' preferences in software design: insights from a focus group. *Interpersonal computing and technology*, 4(2):27–36.



MOUTON, J. 2001. How to succeed in your master's & doctoral studies: a South African guide and resource book. Pretoria: Van Schaik.

MUHR, T. 2004. User's manual for ATLAS.ti 5.0. 2<sup>nd</sup> ed. <http://www.atlasti.com/downloads/atlman.pdf> Date of access: 5 May 2007.

MÜLLER, M.M. 2007. Do programmer pairs make different mistakes than solo programmers? *Journal of systems & software*, 80(9):1460–1471, Sep.

NAGAPPAN, N., WILLIAMS, L., WIEBE, E., MILLER, C., BALIK, S., FERZLI, M. & PETLICK, J. 2003. Pair learning: with an eye towards future success. <http://www.springerlink.com/content/bv5gtav53uqmg8hk> Date of access: 6 Feb. 2008.

NASS, C., MOON, Y. & GREEN, N. 1997. Are computers genderneutral? Gender stereotypic responses to computers. *Journal of applied social psychology*, 27(10):864–876.

NELSON, L.J., WEISE, G.M. & COOPER, J. 1991. Getting started with computers: experience, anxiety, and relational style. *Computers in human behavior*, 7:185–202.

PANDOR, N. 2006. Science, teachers, and textbooks. Address by the Minister of Education, at a Maths and Science professional development programme celebration ceremony, Engen Court, Cape Town. [http:// www.info.gov.za/](http://www.info.gov.za/) Date of access: 27 Feb. 2008.

POWELL, R. 2008. Improving the persistence of first-year undergraduate women in computer science. (In Proceedings of the 39th SIGCSE technical symposium on computer science education SIGCSE'08 held in Portland, Oregon, USA, March 12–15.)

PRYOR, J. 1995. Gender issues in groupwork – a case study involving work with computers. *British educational research journal*, 21(3):277–288, Jun.

REINEN, I.J. & PLOMP, T. 1993. Some gender issues in educational computer use: results of an international comparative survey. *Computers and education*, 20(4):353–365.

REINEN, I.J. & PLOMP, T. 1997. Information technology and gender equality: a contradiction in terminis? *Computers and education*, 28(2):65–78.

RUBIN, A., MURRAY, M., O'NEIL, K. & ASHLEY, J. 1997. What kind of educational computer games would girls like? (*In American Educational Research Association. Through the glass wall. Paper presented at the AERA held in Chicago in April 1997.*) <http://mathequity.terc.edu/gw/html/MITpaper.html> Date of access: 12 Aug. 2008.

SALLEH, N. 2008. A systematic review of pair programming research – initial results. (*In Proceedings of NZCSRSC 2008 held in Christchurch, New Zealand in April 2008. p. 151–158.*)

SANDERS, D. 2001. Student perceptions of the suitability of extreme and pair programming. (*In Proceedings of 2001 XP Agile Universe.*)

SANDERS, J. 1997. Teacher education and gender equity. *ERIC Digest*. <http://www.eric.ed.gov> Date of access: 6 Aug. 2008.

SANDERS, J. 2002. Snatching defeat from the jaws of victory: when good projects go bad. Girls and computer science. (*In American Educational Research Association. Paper presented at the annual meeting of the AERA held in New Orleans, LA on 1 to 5 April 2002. p.1–16.*)

SANDERS, J. 2005a. Gender and technology in education: a research review. <http://www.josanders.com/resources> Date of access: 6 Feb. 2008.

SANDERS, J. 2005b. Lessons I've learned in 22 years of working with teachers about girls in IT. <http://www.josanders.com/pdf/Oxford.pdf> Date of access: 6 Aug. 2008.

SEYMOUR, L., HART, M., HARALAMBOUS, P., NATHA, T. & WENG, C. 2005. Inclination of scholars to major in Information Systems or Computer Science. *South African computer journal*, 35:97–106.

SHASHAANI, L. 1993. Gender-based differences in attitudes toward computers. *Computers and education*, 20(2):169–181.

SHASHAANI, L. 1994. Socioeconomic status, parents' sex role stereotypes, and the gender gap in computing. *Journal of research on computing in education*, 26(4):433–451.

SHASHAANI, L. 1997. Gender differences in computer attitudes and use among college students. *Journal of educational computing research*, 16(1):37–51.

SHASHAANI, L. & KHALILI, A. 2001. Gender and computers: similarities and differences in Iranian college students' attitudes toward computers. *Computers and education*, 37(3–4):363–375, Nov/Dec.

SHORE, J. 2006. Successful software: pair programming. [http://www.jamesshore.com/Agile-Book/pair\\_programming.html](http://www.jamesshore.com/Agile-Book/pair_programming.html) Date of access: 27 Oct. 2006.

SHOTICK, J. & STEPHENS, P.R. 2006. Gender inequities of self-efficacy on task-specific computer applications in business. *Journal of education for business*, 81(5):269–273, May/Jun.

SIMON, B. & HANKS, B. 2007. First year students' impressions of pair programming in CS1. (*In Proceedings of the International Computing Education Research workshop, ICER'07 held in Atlanta, Georgia, USA on 15 and 16 September 2007. p. 73–85.*)

SOUTH AFRICA. Department of Education. 1995. White paper on education and training. <http://www.info.gov.za/whitepapers/1995/education.htm> Date of access: 6 Mar. 2008.

SOUTH AFRICA. Department of Education. 2001. National plan for higher education. <http://www.polity.org.za/html/govdocs/misc/higheredu1.htm> Date of access: 10 Mar. 2008.

SOUTH AFRICA. Department of Education. 2003. National curriculum statement. Grades 10–12 (General) Information Technology. Pretoria: Shumani Printers.

SOUTH AFRICA. Department of Education. 2008a. National curriculum statement. Grades 10–12 (General) Learning programme guidelines. Information Technology. 50p. <http://www.education.gov.za/Curriculum/LPGs2008/LPG%20INFORMATION%20TECHNOLOGY.pdf> Date of access: 2 Jun. 2008.

SOUTH AFRICA. Department of Education. 2008b. Subject assessment guidelines: Information Technology. 28p. <http://www.education.gov.za/Curriculum/SAGs2008/SAG%20INFORMATION%20TECHNOLOGY.pdf> Date of access: 2 Jun. 2008.

SOUTH AFRICA. Department of Science and Technology. 2004. Facing the facts: women's participation in science, engineering and technology. Pretoria: NACI(National Advisory Council on Innovation) and DST (Department of Science and Technology).

STARR, L. 2000. An Education World e-interview with Sherry Turkle: is technology just for boys? [http://www.education-world.com/a\\_curr/curr228.shtml](http://www.education-world.com/a_curr/curr228.shtml) Date of access: 6 Aug. 2008.

TEAGUE, J. & CLARKE, V. 1991. Fiction and fact: students' and professionals' perceptions of women in computer science. (*In Proceedings of the Conference on women, work and computerization.* p. 377–390.)

TEO, T. 2007. Perceived importance, enjoyment, and anxiety as correlates of computer attitudes. *Psychological reports*, 100(1):127–135, Feb.

THOMAS, L., RATCLIFFE, M. & ROBERTSON, A. 2003. Code warriors and code-a-phobes: A study in attitude and pair programming. (*In Proceedings of the 34th SIGCSE technical symposium on Computer science education. SIGCSE'03 held in Reno, Nevada, USA from 19 to 23 February 2003.* p. 363–367.)

TILLBERG, H.K. & COHOON, J. MCGRATH. 2005. Attracting Women to the CS major. *Frontiers*, 26(1):126–140.

TOCCI, C.M. & ENGELHARD, G. 1991. Achievement, parental support and gender differences in attitudes toward mathematics. *Journal of educational research*, 84(5):280–286, May/Jun.

TRAUTH, E.M., QUESENBERRY, J.L. & HUANG, H. 2008. A multicultural analysis of factors influencing career choice for women in the information technology workforce. *Journal of global information management*, 16(4):1–23, Oct–Dec.

UNIVERSITY OF JOHANNESBURG. 2008. Akademie vir Inligtingstegnologie. *Beeld–Jip*: 12, 27 Oct.

VANDEGRIFT, T. 2004. Coupling pair programming and writing: learning about students' perceptions and processes. *SIGCSE Bulletin*, 36(1):2–6, Mar.

VANHANEN, J. & LASSENIUS, C. 2007. Effects of pair programming at the development team level: an experiment. (*In Proceedings of the 33rd EUROMICRO Conference on Software Engineering and Advanced Applications.* p. 211–218.)

VARMA, R. 2002. Undergraduate minorities in IT-related fields: findings from a case study in a minority-serving institution. (*In Proceedings of the 2002 American Society for Engineering Education Annual Conference & Exposition held at the University of Louisiana at Lafayette from 20 to March 2002.*)

VEGSO, J. 2005. Interest in CS as a major drops among incoming freshmen. *Computing research news*, 17(3). <http://www.cra.org/CRN/articles/may05/vegso> Date of access: 6 Aug. 2008.

WAKER, P. 2008. Computer olympiad. <http://www.olympiad.org.za/> Date of access: 20 Aug. 2008.

WARSCHAUER, M. 2007. A teacher's place in the digital divide. *Yearbook of the National Society for the Study of Education*, 106(2):147–166, Oct.

WASBURN, M.H. & MILLER, S.G. 2005. Still a chilly climate for women students in technology: a case study. (*In Rosser, S.V. & Fox, M.F., eds. Women, gender and technology. Urbana-Champaign: University of Illinois Press. p. 60–79.*) <http://books.google.com/books?id=nf1E3EFqoXAC&pg=PA60&dq=wasburn+women#PPA61,M1> Date of access: 15 Aug. 2008.

WBD (World Book Dictionary). 2003a. Enjoyment. Chicago, Ill.:World Book, Inc.

WBD (World Book Dictionary). 2003b. Importance. Chicago, Ill.:World Book, Inc.

WERNER, L., DENNER, J. & BEAN, S. 2004a. Pair programming strategies for middle school girls. (*In Proceedings of the Seventh IASTED International Conference on computers and advanced technology in education held from 16 to 18 August 2004.* p. 161–166.)

WERNER, L., HANKS, B. & MCDOWELL, C. 2004b. Pair programming helps female computer science students. *ACM Journal of educational resources in computing*, 4(1):1–8, Mar.

WERNER, L., HANKS, B. & MCDOWELL, C. 2005a. Female computer science students who pair program persist. *ACM Journal of educational resources in computing*. <http://www.cse.ucsc.edu/~charlie/pubs/jeric2005.pdf> Date of access: 22 Apr. 2008.

WERNER, L., HANKS, B., MCDOWELL, C., BULLOCK, H. & FERNALD, J. 2005b. Want to increase retention of your female students? *Computing research news*. <http://www.cse.ucsc.edu/~charlie/projects/pairprogramming/Pipeline.pdf> Date of access: 21 Feb. 2008.

WERNER, L., MCDOWELL, C. & HANKS, B. 2004c. Retaining women in computer science: the impact of pair programming. Project update ITWF/ITR workforce PI's meeting, Nov. [http://www.cse.ucsc.edu/~charlie/pubs/nsf\\_summary\\_nov\\_2004.pdf](http://www.cse.ucsc.edu/~charlie/pubs/nsf_summary_nov_2004.pdf) Date of access 5 Jun. 2008.

WHITLEY, B.E.Jr. 1996. Gender differences in computer-related attitudes: it depends on what you ask. *Computers in human behavior*, 12(2):275–289.

WHITLEY, B.E.Jr. 1997. Gender differences in computer-related attitudes and behaviour: a meta-analysis. *Computers in human behavior*, 13(1):1–22.

WILLIAMS, L, ALEXANDER, W., BERENSON, S., KNIGHT, V., KOSTER, B., OSBORNE, J., VOUK, M. & YOON, S. 2006a. Collaboration through agile software development practices: a means for improvement in quality and retention of it workers. (*In The National Science Foundation's ITWF & ITR/EFW principal investigator conference. Hosted by Department of Computer Science North Carolina State University 2 to 4 April 2006 in Raleigh, NC. p. 182–185.*)

WILLIAMS, L. & KESSLER, R. 2000a. All I really need to know about pair programming I learned in kindergarten. *Communications of the ACM*, 43(5):108–114, May.

WILLIAMS, L. & KESSLER, R. 2000b. The effects of "pair-pressure" and "pair-learning" on software engineering education. (*In Proceedings of the 13th Conference on Software Engineering Education and Training. IEEE Computer Society held in Austin, TX. p. 59–65.*) <http://collaboration.csc.ncsu.edu/laurie/Papers/CSEET.PDF> Date of access: 8 May. 2008.

WILLIAMS, L. & KESSLER, R. 2001. Experiments with industry's "pair-programming" model in the Computer Science classroom. *Computer science education*, 11(1):7–20.

WILLIAMS, L. & KESSLER, R. 2002. Pair programming illuminated. Reading, Massachusetts Addison-Wesley.

WILLIAMS, L., KESSLER, R., CUNNINGHAM, W. & JEFFRIES, R. 2000. Strengthening the case for pair programming. *IEEE Software*, 17(4):19–25, Jul/Aug.

WILLIAMS, L., LAYMAN, L., OSBORNE, J. & KATIRA, N. 2006b. Examining the compatibility of student pair programmers. (*In Proceedings of the Agile 2006 International Conference Agile'06*, IEEE Computer Society held in Minneapolis, MN. p. 411–420.)

WILLIAMS, L., LAYMAN, L., SLATEN, K., BERENSON, S & SEAMAN, C. 2007. On the impact of a collaborative pedagogy on African American millennial students in software engineering. (*In Proceedings of the 29th International Conference on Software Engineering, ICSE'07*)

WILLIAMS, L., MCCRICKARD, D.S., LAYMAN, L. & HUSSEIN, K. 2008. Eleven guidelines for implementing pair programming in the classroom. (*In Proceedings of the Agile 2008 Conference*. p. 445–452.)

WILLIAMS, L., MCDOWELL, C., NAGAPPAN, N., FERNALD, J. & WERNER, L.L. 2003. Building pair programming knowledge through a family of experiments. (*In Proceedings of the IEEE International Symposium on Empirical Software Engineering held in Rome, Italy, 2003*. p. 143–153.)

WILLIAMS, L. & UPCHURCH, R.L. 2001. In support of student pair-programming. (*In Proceedings of the 32nd SIGCSE technical symposium on computer science education, ACM SIGCSE Bulletin*, 33(1):327–331.)

WILLIAMS, L., WIEBE, E., YANG, K., FERZLI, M. & MILLER, C. 2002. In support of pair programming in the introductory computer science course. *Computer science education*, 12(3): 197–212, Sep.

WITI (Women in Technology International). 1997. The ENIAC programmers.  
<http://www.witi.com/center/witimuseum/halloffame/1997/eniac.php> Date of access: 6 Aug. 2008.

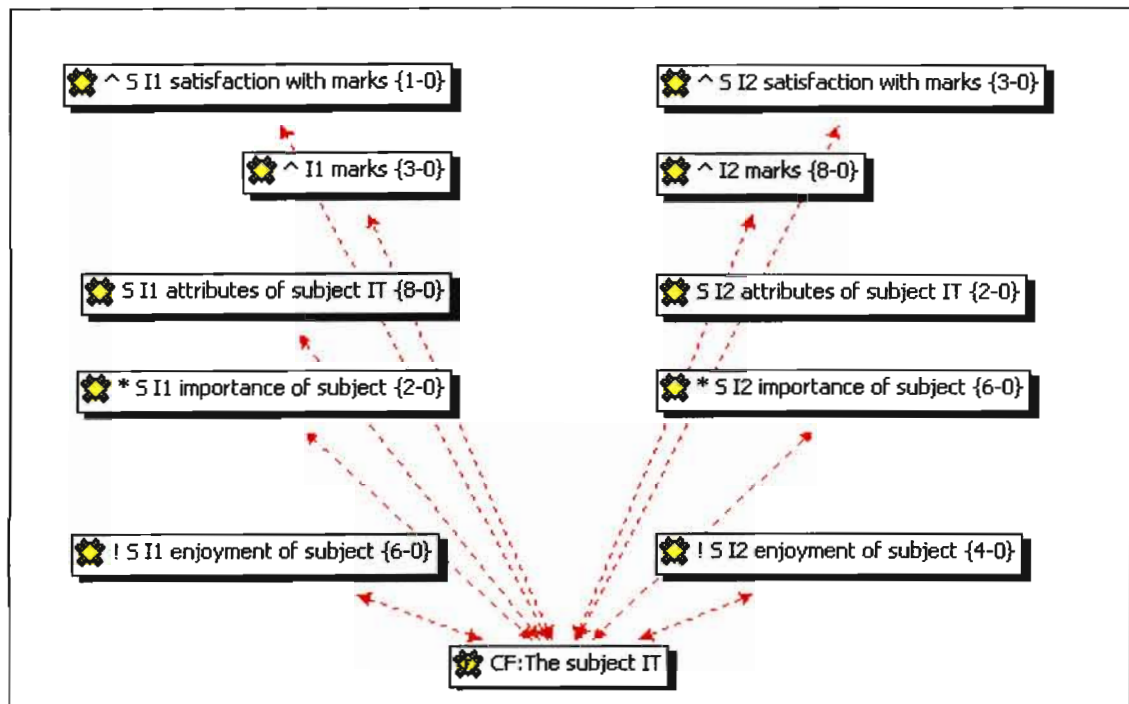
WOODROW, J. 1991. A comparison of four computer attitude scales. *Journal of educational computing research*, 7(2):165–187.



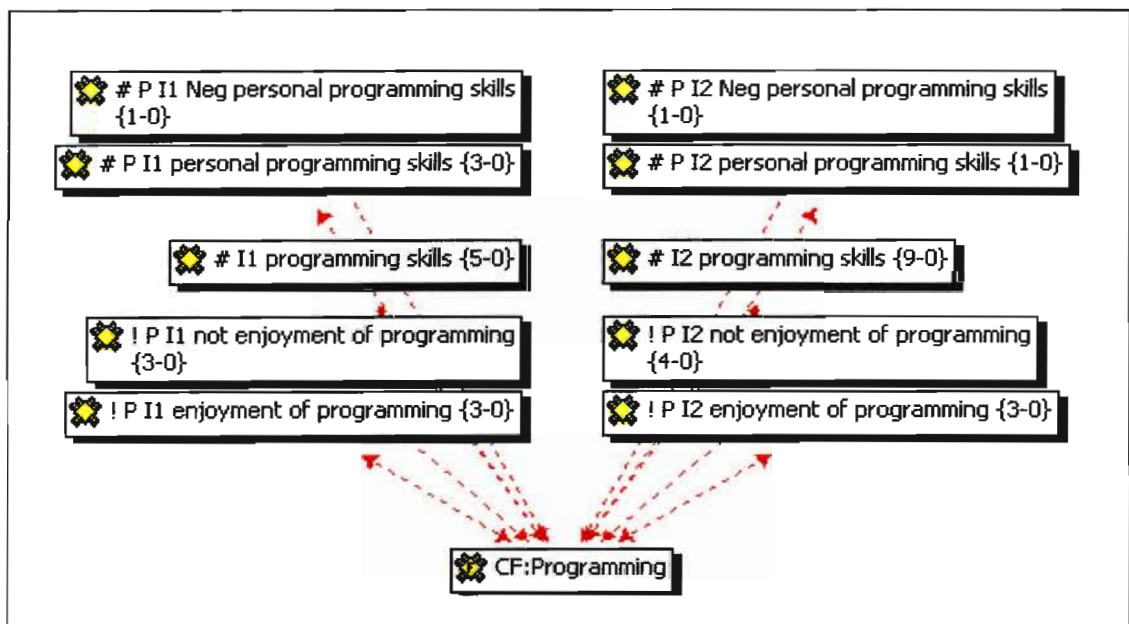
## APPENDIX A

### Example: Networks of Code Families for respondent 1

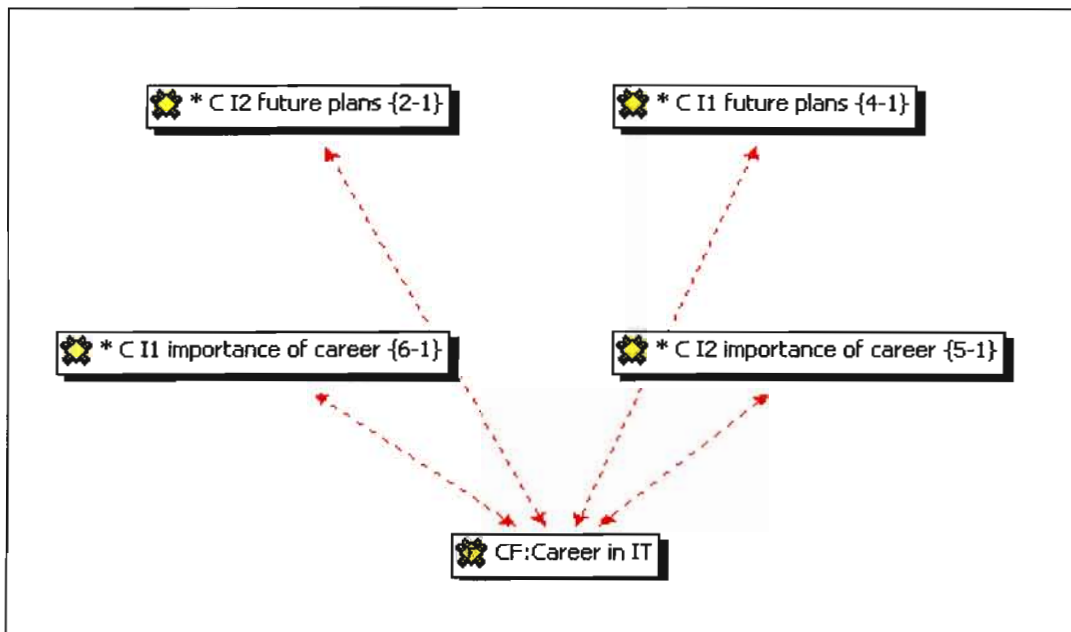
#### CODE FAMILY: The subject IT



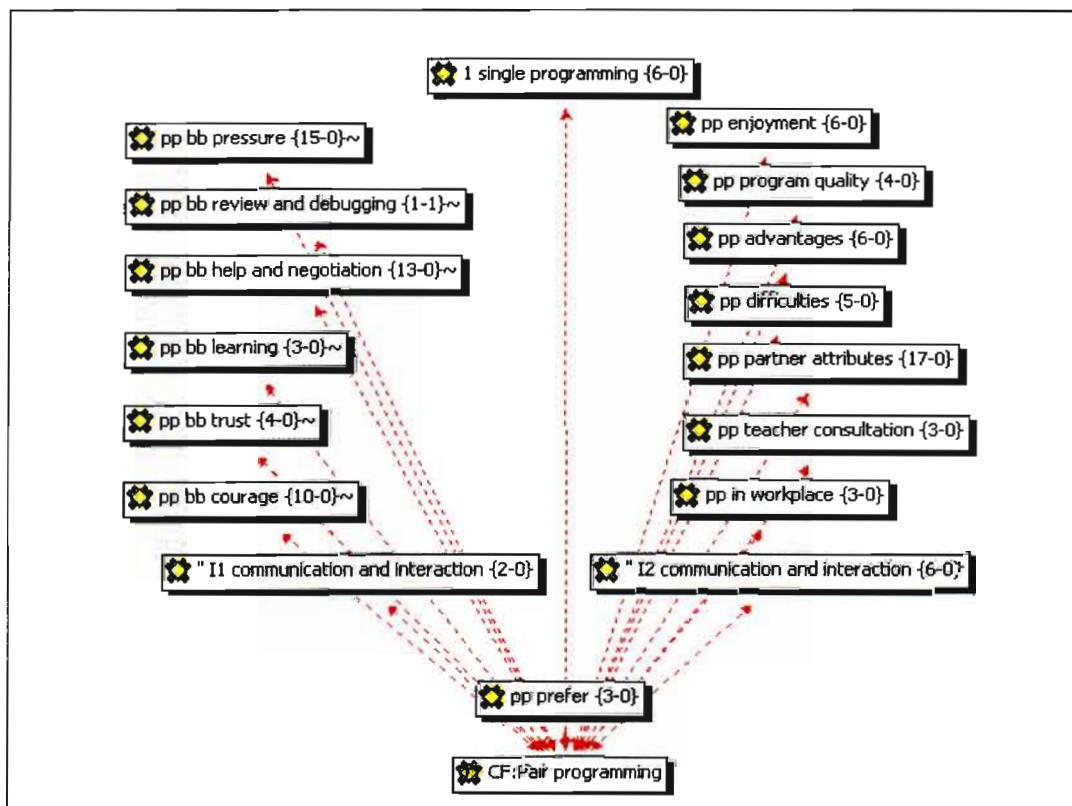
#### CODE FAMILY: Programming



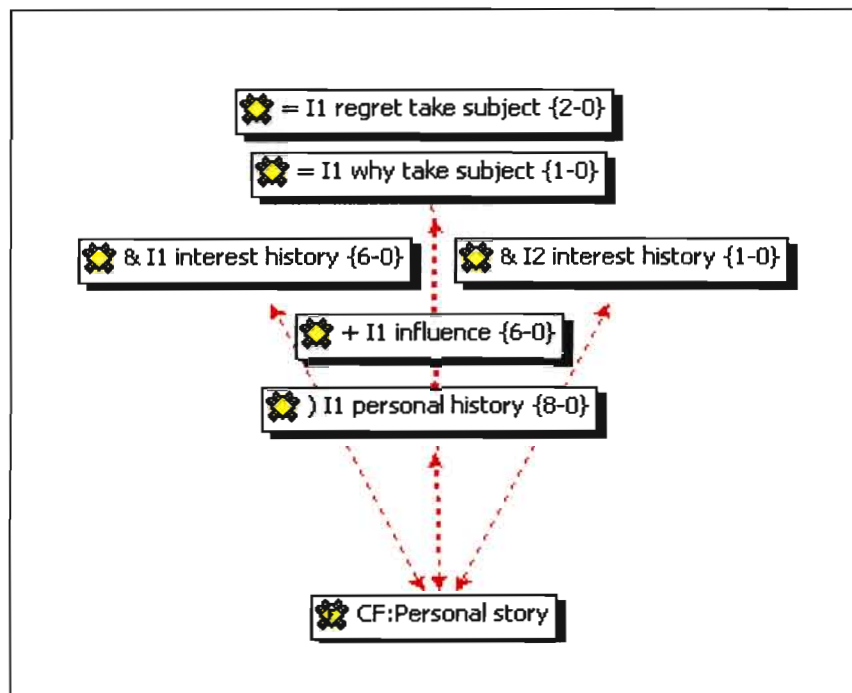
### CODE FAMILY: Career in IT



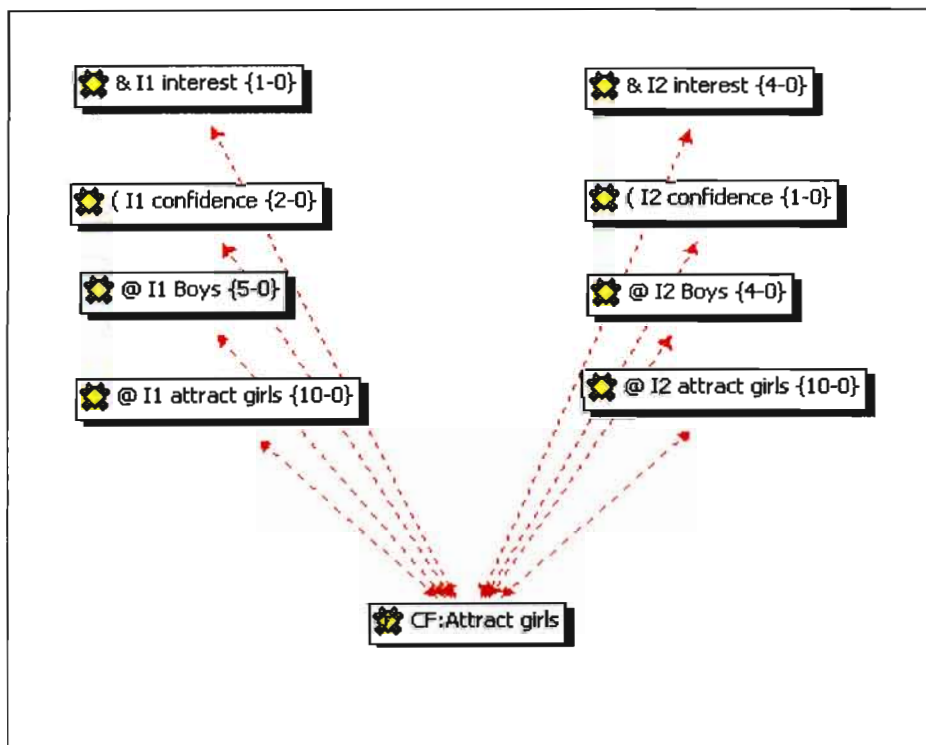
### CODE FAMILY: Pair programming



### CODE FAMILY: Personal story



### CODE FAMILY: Attract girls



## APPENDIX B



NORTH-WEST UNIVERSITY  
YUNIBESITHI YA BOKONE-BOPHIRIMA  
NOORDWES-UNIVERSITEIT  
POTCHEFSTROOMKAMPUS

Privatebag X6001, Potchefstroom  
South-Africa, 2520

Tel: (018) 299-1111/2222  
Web: <http://www.nwu.ac.za>

Mrs. Janet Liebenberg

Tel: (018) 290 7130  
Fax: (018) 2907130  
E-pos: [liebenbergcj@telkomsa.net](mailto:liebenbergcj@telkomsa.net)

Mr H.M. Mwel

Office of the Superintendent-General

3 March 2008

Dear mr Mwel

### PERMISSION TO CONDUCT A PILOT STUDY PROJECT IN NORTH-WEST SCHOOLS ON PAIR PROGRAMMING

I am a masters degree student at the North-West University. I am part of a research project which is sponsored by the NRF concerning the effectiveness of the proposed model for the use of collaborative group work when teaching programming skills in Computer science/Information Technology.

Professor Elsa Mentz is my study leader and she already got permission from your office to conduct the research in six schools (see attached).

I hereby request permission to conduct research with a pretest/post-test approach where learners, before pair programming is implemented in their IT class, will complete a questionnaire on their perception of computers and programming in the IT class. After a period of 6 months of pair programming, learners will again complete the questionnaire to determine if a change in their perception of computers and programming has occurred. Selected learners will also be interviewed to have a more detailed description of their thoughts, experiences and problems.

The results of this research will be made available on request.

Thank you for your attention.

Yours sincerely,

A handwritten signature in black ink, appearing to read 'J. Liebenberg'.  
Me Janet Liebenberg

Student no 10088873

A handwritten signature in black ink, appearing to read 'E. Mentz'.  
Prof E Mentz

(Study leader)

## APPENDIX C



### education

Lefapha la Thuto  
Onderwys Departement  
Department of Education  
**NORTH WEST PROVINCE**

First Floor,  
Garona Building  
Private Bag X2044,  
Mmabatho 2735  
Tel.: (018) 367-3429  
Fax: (018) 367-3430  
e-mail: [ptyatya@nwpg.gov.za](mailto:ptyatya@nwpg.gov.za)

### OFFICE OF THE SUPERINTENDENT-GENERAL

Enquiries: Mpiliso Tyatya  
Tel: 018 367 3429  
Fax: 018 367 3430  
E-mail: [sqedu@nwpg.gov.za](mailto:sqedu@nwpg.gov.za)

05 March 2008

To: University of the North West  
Prof. Elsa Mentz

Attention: Me. Janet Liebenburg

From: Mr. H.M. Mweli  
Superintendent-General

#### PERMISSION TO CONDUCT A PILOT STUDY PROJECT IN NORTH WEST SCHOOLS ON PAIR PROGRAMMING

Reference is made to your letter regarding the above matter. The content is noted and accordingly, approval is granted to your kind self to conduct the pilot study as per your request, subject to the following provisions: -

- That you notify the relevant Regional Managers about your request and this subsequent letter of approval.
- That the onus to notify the Principals of your target schools about your intended visit and purpose thereof rests with your good self.
- That participation in your project will be voluntary.
- That, as far as possible, the general programme of learning and teaching should not be interfered with.
- That the findings of your research must be made available to the Education Department upon request.

With my best wishes

MR. H.M. MWELI  
SUPERINTENDENT GENERAL

**"STAND UP, TEAM UP AND REACH OUT"**  
**"A PORTRAIT OF EXCELLENCE"**

## APPENDIX D



NORTH-WEST UNIVERSITY  
YUNIBESITHI YA BOKONE-BOPHIRIMA  
NOORDWES-UNIVERSITEIT  
POTCHEFSTROOMKAMPUS

Privaatsak X6001, Potchefstroom  
Suid-Afrika, 2520  
Prof E Mentz  
Skool vir Onderwysersopleiding  
Tel: (018) 2991858  
Faks (018) 2994238  
E-pos: elsa.mentz@nwu.ac.za  
Janet Liebenberg  
Tel: (018) 2907130 Sel: 073 071 4396  
Faks: (018) 2907130  
E-pos: liebenbergcj@telkomsa.net

2 March 2009

The Principal  
Fields College

Dear ms Holding,

### RESEARCH: PAIR PROGRAMMING IN THE IT-CLASS

I am a masters degree student at the North-West University and I am part of a research project on pair programming in the Information Technology class. The National Research Foundation (NRF) awarded us research funds to investigate pair programming in schools with the assistance of IT teachers. It has been proven that the implementation of pair programming in the teaching of programming skills delivers improved results compared to individual teaching and learning. The NWU applies this principal with success in the training of teachers and a successful pilot study has already been conducted in 2008 at a number of schools in the North-West province.

Professor Elsa Mentz is my study leader and she is also the project leader of the NRF-project from which my study originates. The North-West Education Department has already given permission for this research in schools in the province (see enclosed). My specific interest is the shortage of girls in the IT-class and how they can benefit from pair programming. Fields College was identified as one of the schools in North-West with a larger number of girls in the IT-class.

Hereby I would like to extend a kind request to involve mr. Homan and your gr.11 IT-girls in the project that will take place from March to June 2009. It involves initial interviews with the gr.11 IT-girls about the subject IT and programming as well as the training of the teacher for about an hour, where the teacher will be equipped to apply pair programming in the IT-class. Mr. Homan will be requested to apply pair programming as teaching strategy in the gr.11 class. After the gr.11-

learners have programmed in pairs for a few months, a second set of interviews will be conducted with the gr.11 IT-girls to determine their perceptions regarding the subject IT, programming and pair programming.

If your school is prepared to partake in the project, I will make an appointment with mr. Homan myself at a time convenient to him. The ethical procedures of the university will be followed, which includes getting written permission from the parents of the girls involved in the project. The findings of the project will also be made available to you.

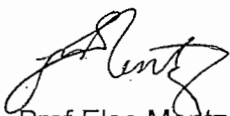
We are convinced that this is an opportunity where the university and school can take hands to do practice specific research that can consequently contribute to the improvement of the teaching and learning in our schools.

In case you need any further information, you are welcome to contact me at the numbers above. If you prefer, I can come and see you personally as well. Further I would appreciate it if you can fax the attached sheet back to me.

Kind regards,

A handwritten signature in black ink, appearing to read 'Janet Liebenberg', with a stylized, flowing script.

Janet Liebenberg

A handwritten signature in black ink, appearing to read 'Elsa Mentz', with a stylized, flowing script.

Prof Elsa Mentz

Study leader

## APPENDIX E



NORTH-WEST UNIVERSITY  
UNIBESITHI YA BOKONE-BOPHIRIMA  
NOORDWES-UNIVERSITEIT

Private Bag X6001, Potchefstroom  
South Africa 2520

Tel: (018) 299-4900  
Faks: (018) 299-4910  
Web: <http://www.nwu.ac.za>

### Ethics Committee

Tel +27 18 299 1366  
Fax +27 18 297 5308  
Email [Ethics@nwu.ac.za](mailto:Ethics@nwu.ac.za)

28 August 2007

Dear Prof E Mentz (*project leader*)

### ETHICS APPROVAL OF PROJECT

The North-West University Ethics Committee (NWU-EC) hereby approves your project as indicated below. This implies that the NWU-EC grants its permission that, provided the special conditions specified below are met and pending any other authorisation that may be necessary, the project may be initiated, using the ethics number below.

Project title: Group work in Computer Science Education									
Ethics number	NWU-EC-10-10-11-2-10-7-1A-2								
Approval date	3 August 2007				Expiry date	3 August 2012			

Special conditions of the approval (if any):

#### General conditions:

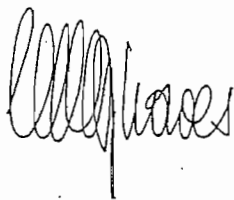
While this ethics approval is subject to all declarations, undertakings and agreements incorporated and signed in the application form, please note the following:

- The project leader (principle investigator) must report in the prescribed format to the NWU-EC:
  - annually (or as otherwise requested) on the progress of the project,
  - without any delay in case of any adverse event (or any matter that interrupts sound ethical principles) during the course of the project.
- The approval applies strictly to the protocol as stipulated in the application form. Would any changes to the protocol be deemed necessary during the course of the project, the project leader must apply for approval of these changes at the NWU-EC. Would there be deviation from the project protocol without the necessary approval of such changes, the ethics approval is immediately and automatically forfeited.
- The date of approval indicates the first date that the project may be started. Would the project have to continue after the expiry date, a new application must be made to the NWU-EC and new approval received before or on the expiry date.
- In the interest of ethical responsibility the NWU-EC retains the right to:
  - request access to any information or data at any time during the course or after completion of the project;
  - withdraw or postpone approval if:
    - any unethical principles or practices of the project are revealed or suspected,
    - it becomes apparent that any relevant information was withheld from the NWU-EC or that information has been false or misrepresented,
    - the required annual report and reporting of adverse events was not done timely and accurately,
    - new institutional rules, national legislation or international conventions deem it necessary.



The Ethics Committee would like to remain at your service as scientist and researcher, and wishes you well with your project. Please do not hesitate to contact the Ethics Committee for any further enquiries or requests for assistance.

Yours sincerely

A handwritten signature in black ink, appearing to read 'M Lowes', with a stylized, cursive script.

Prof M Lowes  
(chair NWU Ethics Committee)

## APPENDIX F

Jackie Viljoen  
Algemene Taalpraktisyn  
Bergzicht Gardens 16  
Fijnbosslot  
STRAND 7140

Lid van die Suid-Afrikaanse Vertalersinstituut

☎ +27+21-854 5095    📠 082 783 0263    📠 +27+21-853 5944

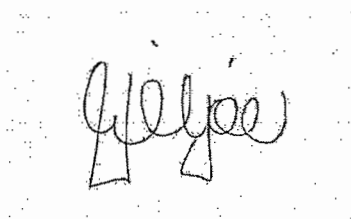
Posadres: Bergzicht Gardens 16, Fijnbosslot, STRAND 7140, Suid-Afrika

---

### VERKLARING

Ek verklaar hiermee dat die M-tesis van Janet Liebenberg behoorlik deur my taalversorg is.

Titel van tesis: Secondary school girls' experiences of pair-programming in Information Technology



JACKIE VILJOEN  
STRAND  
Suid-Afrika  
17 November 2009

Jackie Viljoen  
Algemene Taalpraktisyn  
Bergzicht Gardens 16  
Fijnbosslot  
STRAND 7140

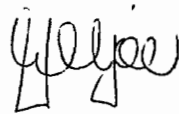
Geakkrediteerde lid van die Suid-Afrikaanse Vertalersinstituut  
Nommer APSTrans 1000017

☎ +27+21-854 5095    📠 082 783 0263    📠 +27+21-853 5944  
Posadres: Bergzicht Gardens 16, Fijnbosslot, STRAND 7140, Suid-Afrika

---

### VERKLARING

Ek verklaar hiermee dat die bronnelys van die M-tesis van JANET LIEBENBERG  
behoorlik deur my nagegaan is met inagneming van die riglyne van die NWU.



JACKIE VILJOEN  
STRAND  
Suid-Afrika  
23 November 2009

## Form for Informed consent

Part 1: General Project Information

Part 2: General Principles

Part 3: Consent

### Part 1: General Project Information

The part below provides you as participant in the project with more information, so that you can make an informed decision about your voluntary participation or not.

1. *Title of the Project:* Secondary school girls' experiences of pair-programming in IT.
2. *Institution / School / Subject group:* North-West University / School of education / Computer science education
3. *Names & contact details of Project Staff:*

*(These persons are your first line of contact for enquiries, help and complaints related to the project or your participation in the project. If you need any help, feel uncertain or have any questions regarding the project, or if you experience any unwanted effects of the project interventions, feel victimised or have any other complaints related to the project, you may contact these persons at any time.)*

	Contact person	Project head
Title, name & surname	Mrs Janet Liebenberg.....	Prof Elsa Mentz.
Full names	Janet Adri.....	Elsa.....
Function in Project	Student.....	Study leader.....
Qualifications	BSc (HOD) Bed Hons.....	BA (HOD) BSc Hons, MSc, PhD.
Telephone (home)	018 290 7130.....	018 290 5053.....
Telephone (work)	.....	018 299 1858.....
Cellphone	073 071 4396.....	0836607181.....
Postal address	PO Box 20844, Noordbrug, 2522	Private Bag X6001, Potchefstroom, 2520.....

4. You were approached to take part in this project and may now have the following questions:
  - 4.1. *The set requirements that persons must meet to be able to take part in the project:* You must be a girl in Gr 11 and take IT as a subject.
  - 4.2. *Reason why you were chosen:* Your school has a greater number of Gr 11 girls that take IT as a subject than other schools in the province and we want to use a class with as many girls as possible for the research.

- 4.3. *The purpose of this project:* To obtain more information regarding girls' perceptions regarding the subject IT and to find a possible solution for the shortage of girls in the IT classroom.
- 4.4. *What is expected of you as a participant:* You will be asked to participate in two interviews where you will have to give your honest opinion regarding the subject IT and programming. The first interview will be early in your Gr 11 year and the second will be about four months later.
- 4.5. *How long I am expected to be involved in the project:*

Visit	Date	Duration
.....1.....	March 2009..	.....½ hour....
.....2.....	June 2009....	.....½ hour....

- 4.6 *Potential general benefits for the broader community, which may arise from the project:* We hope that with this research we can address the shortage of girls in IT and to give guidelines to teachers to address girls' interests and attitudes in the IT class.
- 4.7 *How the findings of the project will be made available or conveyed to you:* The results will be made available to the school and we plan to publish it in a accredited scientific magazine.
- 4.8 *Measures that have been taken to guarantee the confidentiality of the data:* The information from the interviews will at all times be treated **vertroulik** and anonymously. The analysis of the data will be treated and reported anonymously by the researcher. The interview data will not be made available to any other person.

As Project Head, I confirm to participants that the above information is complete and correct.

2	0	0	9		0	3		0	6
c	c	y	y		m	m		D	d

Signature of Project Head

Date

Signed at .....

City/Town

## **PART 2: General Principles**

### **To the signatory of the consent contained in Part 3 of this document:**

You are invited to take part in the research project as described in Part 1 of this informed consent form. It is important that you also read and understand the following general principles, which are applicable to all participants in our research projects:

1. Participation in the project is completely voluntary and no pressure, however subtle, may be placed on you to take part.
2. It is possible that you may not derive any benefit personally from your participation in the project, although the knowledge that may be gained by means of the project, may benefit other persons or communities. You may not be bribed to participate.
3. You are free to withdraw from the project at any time, without stating reasons, and you will in no way be harmed by so doing. You may also request that your data no longer be used in the project. However, you are kindly requested not to withdraw from the project without careful consideration, since it may have a detrimental effect on, inter alia, the statistical reliability of the project.
4. By agreeing to take part in the project, you are also giving consent for the data that will be generated to be used by the researchers for scientific purposes as they see fit, with the caveat that it will be confidential and that your name will not be linked to any of the data without your consent.
5. The NWU Ethics Committee and/or a Court of Law may request access to information to ensure/inspect the ethical responsibility of practices, in the interest of participants and the public.
6. You will be given access to your own data upon request.
7. A summary of the nature of the project and the benefits that can be expected by you as participant, are set out for you in Part 1 hereof.
8. You are encouraged to ask the Project Head or co-workers any questions you may have regarding the project and the related procedures at any stage. They will gladly answer your queries. They will also discuss the project with you in detail.
9. If you are a minor, the written consent of your parent or legal guardian is required before you participate in this project, as well as (in writing if possible) your voluntary assent to take part – no coercion may be placed on you.
10. The project objectives are always secondary to your well-being and actions taken will always place your interests above those of the project.

## PART 3: Consent

Title of the Project: Secondary school girls' experiences of pair-programming in IT.

### Legal Parent / Guardian

For all research and interventions on participants under the age of 21 years, the written proxy consent of the parent or legal guardian is required.

I,

\_\_\_\_\_  
**Full names & Surname**  
(parent or legal guardian)

\_\_\_\_\_  
**Relationship**  
(parent or legal guardian)

of the participant mentioned above, hereby give consent for her to take part in this project and I hereby exempt the University, as well as any employee or student of the University, from any liability from any detrimental effect that may arise in the course of the project, unless such injury, damage or death is caused by the negligence of the University, its staff and/or its students. I furthermore declare that I have read the preceding premises in connection with the project, as discussed in **Part 1** and **Part 2** of this informed consent form, and have also heard the oral version thereof and I declare that I understand it. I have also initialled every page of **Part 1** and **Part 2**. I was given the opportunity to discuss relevant aspects of the project with the Project Head.

2 0 - -  
c c y y m m d d

\_\_\_\_\_  
**Signature of Legal Parent/Guardian**

**Date**

Signed at

\_\_\_\_\_  
**Place of Signature**

I,

\_\_\_\_\_  
**Full names & Surname**  
(minor participant)

\_\_\_\_\_  
**Relationship**  
(participant)

minor child of the above parent/guardian, hereby give my voluntary assent to take part in the project and declare that I understand what the participation involves.

2 0 - -  
c c y y m m d d

\_\_\_\_\_  
**Signature of Minor Participant**

**Date**

Signed at

\_\_\_\_\_  
**Place of Signature**

**Witnesses**

20

-

-

c

c

y

y

m

m

d

d

Signature of Witness 1

Date

Signed at

Place of Signature

20

-

-

c

c

y

y

m

m

d

d

Signature of Witness 2

Date

Signed at

Place of Signature