Pair programming and secondary school girls' enjoyment of programming and the subject Information Technology (IT)

Janet Liebenberg , Elsa Mentz & Betty Breed

Routledge
Taylor & Francis Group

Pair programming and secondary school girls' enjoyment of programming and the subject Information Technology (IT)

Janet Liebenberg*, Elsa Mentz and Betty Breed

Faculty of Education Sciences, North-West University, Private Bag x6000, Potchefstroom, 2520, South Africa

This paper reports on a qualitative study that examined how pair programming shapes the experience of secondary school girls taking IT as a subject, with respect to their enjoyment of programming and the subject itself. The study involved six Grade 11 girls who were doing solo programming in Grade 10 and pair programming in their following Grade. The results showed that the girls enjoyed the subject more when programming in pairs due to improved comprehension of the task. They especially enjoyed the socialization and communication brought about by pair programming. The assistance, support, motivation, focus and encouragement they received from partners when stuck or while fixing errors made the programming experience more enjoyable for them. The increased enjoyment brought about by pair programming resulted in the perception of greater learning in the subject IT and also to greater interest in it. It also led to greater persistence in dealing with problems. Pair programming should be implemented right from the start of Grade 10 since it may lead to greater enjoyment of programming and the subject IT in general. The approach may also lead to more girls being attracted to the subject.

Keywords: cooperative/collaborative learning; gender studies; pair programming; secondary education; teaching/learning strategies

Introduction

In a time of great technological advancement where the computer plays an ever increasing role, and in which women increasingly take up positions in the labor market, the low number of girls in the secondary school phase taking Information Technology (IT)[1] is conspicuous. In the United Kingdom, the National GCE A-level results of June 2010 showed that of the 4065 learners who had written the Computing examination,

*Corresponding author. Email: janet.liebenberg@nwu.ac.za

only 361 (8.9%) were girls (Oliver, 2010). Statistics of the North-West Province (South Africa), where this study was conducted, showed that in 2007 and 2008 the females (27.2%) taking IT as a subject were far fewer than the males (72.8%).

Another cause for concern is the low retention rate of learners in IT, especially girls. For example, statistics of the North-West Province schools showed that the number of girls taking IT as a subject had nearly halved from 2007 to 2008. In 2008 there were 44 Grade 11 girls opposed to the 80 who had started the subject in Grade 10 in 2007. 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 from taking computer courses in secondary school on through university and into IT careers (Gürer & Camp, 2002). Goode, Estrella, and Margolis (2006) are convinced that the scientific heart of computer science is "lost in translation" at the secondary school level, resulting in the field continuing to lose the participation and interest of a broad layer of students, especially females. According to McKinney, Wilson, Brooks, O'Leary-Kelly and Hardgrave (2008), the IT industry faces more of an input problem than a throughput problem. In other words, the under-representation problem is rather caused by fewer women entering IT and not so much by large numbers of women leaving. This implies that greater attention should be paid to supply-side issues such as helping young girls understand the importance of computer careers and to identify with the subject at a tender age.

A closely related issue is "critical mass" of girls taking the subject. It is not so much the presence of boys at the computers that discourages girls from participating, but rather the absence of the girlfriends of the girls taking the subject (Sanders, 2005). Cohoon (2001) found that this "critical mass" of other women present correlated more strongly than any other factor with women's retention in computer science majors in USA universities.

In view of the above, the 2009 NIC (New Image for Computing) Report of the WGBH Educational Foundation and the Association for Computing Machinery (ACM) decided to concentrate on girls as a special target group because of their observation of a "significant gender gap" in young people's attitudes toward computer science. "As long as teenagers believe that computer science is boring, difficult, antisocial, or doesn't have much impact on solving the world's problems, they're unlikely to choose it for their future" (ACM, WGBH, 2009).

In South Africa, provision was made for school learners to prepare for further study and a career in Computer Science by offering them the subject IT at high or secondary school level. One of the four learning outcomes of the subject focuses on the design and development of

appropriate computer-based solutions to specific problems through using programming in Delphi. The weight allocated to this learning outcome is 60%, which means that learners should be busy with programming for more than half of their time in the IT class. This explains why this study focused on the programming component of the subject IT.

Williams and Upchurch (2001) suggested that pair programming, where two programmers work at one computer on the same programming task, shows several promising properties for educational purposes. A number of research studies on pair programming in tertiary institutions have since been done, and to a lesser extent also studies on women and pair programming. Very few studies have so far focused on girls in the secondary school IT (elsewhere known as Computer Science – CS) classroom (Goode et al., 2006; Kelleher, Pausch, & Kiesler, 2007). No studies conducted in the South African IT classroom could be found. The purpose of this study was to determine whether pair programming indeed improved girls' enjoyment of programming and of the subject IT in general.

Conceptual and theoretical framework

Girls and the computer environment

Girls (especially those aged around 14) have more negative feelings about the computer than boys and enjoy working with computers less than boys (Christensen, Knezek, & Overall, 2005; Sanders, 2005; Shashaani, 1997). They do not enjoy the competitive and anti-social environment associated with computers but prefer collaboration, completion and relevance to the real world (Chou & Tsai, 2007; Frieze, 2007). They are attracted by computing as a form of communication, a means of creative self-expression, or as a path to a helping career (Shotick & Stephens, 2006; Tillberg & Cohoon, 2005).

Most countries do not provide a diverse, interesting curriculum for CS in secondary schools. The standard computer curriculum focuses exclusively on programming or emphasizes basic skills as opposed to problem solving (Goode et al., 2006). Goode et al. (2006) furthermore 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 clear that the continued under-representation of girls in IT/CS underscores the need for strategies in order for girls to rediscover the passion, beauty, joy, and awe of computing and experience the IT class as fun (Garcia, Chapman, Hazzan, Johnson, & Sudol, 2010). Hazzan (in: Garcia et al., 2010) believes that in order to generate greater excitement among potential IT/CS (henceforth CS) students, it should be highlighted in the CS class room that human aspects, in general, and teamwork in particular, are at the heart of the actual work in CS.

Pair programming whereby two programmers develop software side-by-side at a single computer appears to be a strategy for effecting this. It touches on many issues that women face in the CS field, such as the issues surrounding socialization, image, and confidence (Balcita, Carver, & Soffa, 2002; McDowell, Werner, Bullock, & Fernald, 2003). Pair programming seems advantageous to education in general and might lead to retention in the number of female university students enrolled in CS courses. The question still remains whether pair programming could have an effect on secondary school girls' enjoyment of programming and the subject IT, and whether it could possibly result in an improvement in their enrollment and retention rate in IT/CS at school and university level.

Pair programming in the IT/CS class

Traditional introductory programming CS courses generally require that learners work individually on their programming assignments. This approach teaches learners that software development is an individual activity, potentially giving them the mistaken impression that programming is a lonely career. On the contrary, pair programming seems to have a positive effect in general on CS students of both genders at universities (Bishop-Clark, Courte, & Howard, 2006; Simon & Hanks, 2008; Williams & Kessler, 2002), on female CS students (Ho, Slaten, Williams, & Berenson, 2004; Werner, Hanks, & McDowell, 2004a; Werner, Hanks, McDowell, Bullock, & Fernald, 2005), specifically in terms of enjoyment (Ho et al., 2004; McDowell, Werner, L., Bullock, & Fernald, 2003; Werner, McDowell, & Hanks, 2004b).

While the positive effects of pair programming in general are a fairly settled matter, differences in the effects of pair programming by gender are not. Braught, Wahls, and Eby (2011) found no significant differences by gender in the effect that pair programming had on a number of measures and they suggest that further investigation is required to determine if pair programming affects the performance and/or attitudes of men and women differently.

The following advantages of pair programming specifically focused on females were discovered by researchers in a tertiary education context:

- Female students working in pairs enjoy the programming process (Werner et al., 2004b); the enjoyment comes from the usefulness of the program and teamwork (Ho et al., 2004). These findings were significant for the study reported in this paper, since the aim of the study was to determine whether pair programming had an effect on secondary school girls' enjoyment of the subject IT.
- Typically, female CS 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; Werner et al., 2004b).

- 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., 2003; Werner et al., 2004a).
- The collaborative nature of pair programming teaches female students that programming is not the competitive, socially isolating activity that they imagined (Werner et al., 2004a).
- 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 when all course grades including exam scores and programming assignment scores were taken into consideration (Werner et al., 2004b).
- 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, Slaten, Williams, & Ho, 2004).

It is clear that pair programming can be beneficial to female programmers because it might address factors that potentially limit their participation in CS. It was therefore assumed that research on secondary school girls' experiences could provide valuable information about how to attract more female students to CS at tertiary level.

Research method
Research design
A basic qualitative design (Merriam, 2009) was used, aiming at assisting the researcher to gain understanding of how people make sense of their lives and their experiences. Computer Science research is traditionally not done with a qualitative design but for this study it was essential to work qualitatively to understand how the girls interpreted their experiences with pair programming and what meaning they attributed to their experiences.

Participants
The very reason for this study (i.e. the shortage of girls in CS/IT in South Africa) restricted the number of participants to only six ($n = 6$). Statistics

provided by the North-West Department of Education (South Africa) of 14 schools in the province showed that this school had the largest IT class of Grade 11 girls in the North-West Province available. The other schools only had one or two Grade 11 IT girls which did not make the study feasible to be conducted in those schools. The school where the study was conducted is multi-cultural and has pupils from an average to high socio-economic background. These six female learners were used because they had had experience in solo-programming in their Grade 10 year and were amenable to being subjected to pair programming in Grade 11. Nine girls started with the subject in their Grade 10 year, but three dropped out.

Contrary to quantitative researchers who strive to collect large amounts of data through random selection methods in order to generalize, the qualitative researcher has the responsibility to provide enough description about the context of the sample in order to allow others to adequately judge whether the findings apply to their situation (Byrne, 2001).

Data collection and instruments

Two sets of semi-structured interviews were conducted based on interview schedules (see Appendix 1). Permission to conduct the study had been obtained from the superintendent-general of the North-West Department of Education. Permission from the principal of the target school in the North-West Province was obtained and the teacher of the IT class agreed to participate and implement pair programming as a teaching strategy in his class. The learners and their parents signed an informed consent form and they were informed that confidentiality and anonymity would be ensured. The researcher indicated on the informed consent form and at the first interview that the researcher is interested in girls and their enjoyment of programming and the subject IT. All interviews were recorded.

The IT teacher was trained by the researchers in a 2 hour session on the implementation of pair programming and the principles of cooperative learning. 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 by the researcher.

At the first meeting, interviews of about 20 minutes each were conducted by the researcher to determine the girls' perceptions of and attitudes toward programming and the subject IT. After the first interview, all the Grade 11 IT learners were trained by their class teacher in the application of pair programming. They were informed that pair programming was to be used for all programming assignments during the term – both the boys and the girls were doing pair programming.

After three months, now that the teacher had used pair programming as a teaching strategy in his class and the learners had completed several

paired assignments, the girls were interviewed again for approximately half an hour each by the researcher to again determine the girls' perceptions of and attitudes towards programming and the subject IT.

In the first interview, a few warm-up questions were asked to put the girls at ease. The rest of the questions used in the first interview were also used in the second interview to qualitatively determine if there was any change in the girls' enjoyment of programming and of the subject IT. In the second interview, the girls' experiences during pair programming were also explored. In order for the girls to give objective responses in the interviews, the researcher only asked them about pair programming in the second interview from question 17 onwards (see Appendix 1). The girls were under the impression that the introduction of pair programming was an initiative of their teacher.

Data analysis

By means of the ATLAS.ti 5.2 computer program, each participant's transcribed data from the two interviews were coded into themes and subthemes and analyzed for dominant themes. The question central to this analysis was, ''How does pair programming shape secondary school girls' experience with regard to their enjoyment of programming and of the subject IT?'' Information that participants offered in response to other questions was also analyzed, such as: What can be done to attract more girls to IT? From the data analysis, an overall description of their experiences and feelings about pair programming and of the subject IT were described as the girls experienced it. Since the product of qualitative research is richly descriptive and the purpose is not to generalize (Merriam, 2009) the results of the study can now be presented in the form of quotes from the participant interviews. Although validity and reliability are issues that are hard to control and measure in qualitative research, the researcher used peer reviews to promote the trustworthiness of the study through discussions with other researchers familiar with pair programming regarding the process of the study and tentative interpretations.

Results and discussion

The following themes emerged from the data analysis.

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

From the above quote it can be concluded that pair programming had changed at least one girl's perception of the enjoyment of the subject IT, but she was not unique in this. Since this study focused on girls who had already been involved in the computer environment by taking the subject

IT, it was not surprising that most of the girls had enjoyed the subject IT even before pair programming was introduced, but once pair programming had been used, they reflected an even more positive attitude to the subject and pair programming. Words like "fun", "really nice", "I love it" were used once pair programming had been introduced. The following response was given to the question: Do you enjoy the subject IT? after pair programming had been introduced

> 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 better … So, I think we're enjoying it a bit more because we understand it better now. (The participants responded in English, which in some cases was their second language.)

The literature showed that the pair-programming experience makes the subject more enjoyable to university students. This study with secondary school girls confirmed this finding. They repeatedly described pair programming as "fun" and the word "more" ("enjoyed it more", "much more fun", "more exciting") was used repeatedly. The participants were asked whether they preferred pair-programming or solo-programming. Four participants said they preferred pair programming but the two girls who claimed that they preferred 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". These are all strong indications that even the smaller proportion of the class who preferred solo programming also enjoyed the pair-programming experience and saw the benefit of doing it.

The girls described taking the subject IT prior to pair programming as frustrating and stressful but once pair programming had been introduced the situation changed:

> 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.

One respondent enjoyed the theory part of the subject more than the programming before pair programming had been introduced. Once pair programming had been introduced, her view changed:

> Before I didn't, but now that I understand it and know exactly how to fix my mistakes I do enjoy it (programming in Delphi), much more than the theory. Whereas before, I enjoyed the theory more than the Delphi.

Analysis of the data brought forward a number of themes relating to the girls' enjoyment of programming and IT as a subject:

Socialization and Communication

Secondary school girls are very sociable; they like collaboration:

> Sometimes we sit as an individual and we try to code but we also have that urge to like communicate with our partners.

> Girls they like to work with other people and they like to talk all the time.

The fact that they could communicate and interact with their friends while learning was described as one of the most enjoyable parts of the subject brought about by pair programming. In the first interview, before pair programming was introduced, one of the participants 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 had been introduced, when asked what she particularly liked about pair programming, she responded as follows:

> 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.

Improved comprehension

All the participants commented on how they felt that pair programming improved learning and comprehension, because each partner had knowledge and skills to offer to the others.

> Now that I understand it and know exactly how to fix my mistakes I do enjoy it (programming in Delphi)

> Yes, I do (enjoy programming) because . . . . 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.

The programs written in pairs are of a better quality because thinking skills improve and ideas can be shared.

> 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.

> But now that I had someone else with me . . . . the program actually worked better

Even the one of the girls who said that she preferred solo-programming pointed out in the very same sentence that

her programming had improved since they started using pair programming.

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

The girls reported that through pair programming they learned that there were different methods and solutions to a problem. All of the participants mentioned how they learned from their partners new methods to solve a problem, and that they realized that there were different correct solutions to a problem.

> 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. I'm thinking out of the box, like you can think other ways.

These girls' comments concurred with the findings of other studies (Fisher, Margolis, & Miller, 1997; Goode et al., 2006; Seymour, Hart, Haralambous, Natha, & Weng, 2005) that showed that girls had far less computer experience than boys and they usually stuck to what they had been taught in class, not exploring further, like most boys. Pair programming taught them more than just the one method or solution the teacher could manage in class time.

Help and support

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 better solutions. The fact that they could help their partners was singled out as one of the most enjoyable things about pair programming.

> 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.

The girls had fewer problems with errors once pair programming had been introduced. Furthermore, getting stuck and the struggle to fix errors were factors that negatively influenced the enjoyment of programming, but with the help of a partner in pair programming, they got stuck less and fixed problems faster and more effectively. 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 introduced the following was said: "Yes, I do (enjoy pair programming) because I don't get stuck a lot."

A suppressor to the enjoyment of programming was having to consult the teacher when getting stuck. "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." During pair programming the girls had to consult their teacher far less than before. The fact that they had to consult their teacher less was described as one of the best things about pair programming: "I liked the fact that I didn't have to ask the teacher anything. I really liked that." This resulted in the teacher getting a more favorable impression of the girls. Girls valued their teacher's opinion, but felt that the teacher had sometimes been unintentionally discriminatory towards them. "Not having to say: Sir! all the time. So, obviously he gets a better perspective on you."

Motivation, focus and encouragement

When asked what the reasons were for the shortage of girls in IT, the following was said:

> They might feel that they're not good enough to be able to be in IT. That might lack their confidence, 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.

They described how the partners encouraged each other to solve problems that they might not have attempted while working alone. Pair programming was even described as a confidence booster.

> You feel better about yourself, more confident about how you program and stuff.

> 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.

> 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.

This confirmed the findings of several previous studies (Bishop-Clark et al., 2006; Margolis & Fisher, 2002; Shashaani, 1993) that had shown that girls lack confidence in their abilities and pair programming proved to be one strategy to improve their confidence.

The girls also 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. "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.'"

The girls felt a commitment between the two partners, which resulted in them wanting to become involved, to work hard at completing the task at hand and to attempt to do well.

> 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 finish the coding, do it well.

> It puts a lot of pressure and I feel that is good because it makes you want to complete the coding, it makes you want to do the job right.

> 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."

One girl repeatedly said that she enjoyed the fact that, instead of giving up quickly when they were working alone in the past, pair programming changed the classroom scene into a competition of the pairs: "It's the competition of the teams. As individuals we struggled and then we like: 'let's just leave this' but as teams, it becomes like a real competition".

The implications of the increased enjoyment brought about by pair programming were:

Achievement

A critically related issue to the enjoyment of a subject is the achievement in it. Some participants reported an improvement in their marks since pair programming had been introduced and no decline or dissatisfaction with marks was reported. These findings concur with other studies reporting that pair programming improves learner achievement (Bishop-Clark et al., 2006; McDowell et al., 2003; Werner et al., 2004b). The girls opined that pair programming should be used in IT classes because marks would improve as a result.

Greater interest

One girl opined that if pair programming had been introduced earlier, there would have been more girls taking the subject IT. "If they had introduced it earlier I think we would have had much more girls"

When the girls were asked what could be done to attract more girls to IT, the girls agreed that pair programming should have been used when they had been introduced to programming in Grade 10.

> I think about from the beginning of Grade 10 that pair programming should be implemented. 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 implemented from the beginning of your IT schooling years.

It will allow for you to develop much faster, learn faster and understand the work better.

Persistence

The girls opined that pair programming would help to prevent dropping-out from the subject IT.

Yes, 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.

A lot of people have left because they just couldn't grasp it ... So, I think pair programming with IT, like generally through, all IT pairing, I think would really, really help

"Critical Mass"

The girls did not enjoy the fact that there were so few of them in the IT class: "Please try and get more girls in the class because it's just, it's hard to work with boys especially when they (the boys) don't know what they're doing. It is hard". Pair programming will bring about a "critical mass" of girls in other words the more girls there are in an IT class, the more girls will join:

I know it (pair programming) 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 (few) people there 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. Yes, the more people that come, not just getting guys, getting girls as well.

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.

Difficulties of pair programming

The respondents had very little negative perceptions about pair programming but they expressed some frustration at conflict in experience, skills, personality and dedication.

If you have a lazy partner, that really frustrates me. Like, all the work is on you and the pressure as well.

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.

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.

Conclusion

Based on the findings of this study, it can be concluded that pair programming contributed positively to these school girls' enjoyment of programming and of the subject IT itself. This conclusion echoes a remark made by one of the participants: "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!**"

From this research several reasons emerged supporting the belief that pair programming has the potential to increase girls' enjoyment of programming and the subject IT. Pair programming affords the opportunity for increased socialization, improved comprehension and immediate help and support. The girls experience increased motivation, focus and encouragement. Consequently pair programming has the potential to increase interest, participation, persistence in IT and potentially greater learning. Pair programming thus challenges the belief that computing is a solitary activity and fulfills the need of girls for a more social class environment.

It is recommended that IT teachers implement pair programming right at the beginning of a programming course. When implemented correctly, not only the learners, especially girls, will benefit from it, but also the IT teacher. Although this study did not focus on the teacher in the IT class, it is worth mentioning that the teacher involved in this particular study reported that he had noticed such great benefits from using pair programming in his Grade 11 class that he had implemented it in his Grade 10 class as well.

Information Technology teachers are often unaware of what causes girls to enjoy their subject and what drives them away. Information Technology teachers should familiarize themselves with the preferences of the girls in their classes. Pair programming is as yet an untapped resource worth considering in any programming class, especially for attracting more girls to IT – and for retaining them. This sentiment was expressed by one of the participants:

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.

The research reported in this article and further research that might flow from this study, will hopefully make a difference to the number of girls taking IT as a subject and in their persistence in the subject. Information Technology (CS) teachers will hopefully be inspired to implement pair programming in their classes, not only to benefit the girls in their class, but to all the learners. This will contribute to the fulfillment of 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: "That's an IT woman, she knows what she's doing".

Acknowledgments

Note

1. *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.

References

ACM, WGBH. (2009). Interim report. New Image for Computing. Retrieved September 3, 2010, from http://www.acm.org/membership/NIC.pdf.

Balcita, A.M., Carver, D.L., & Soffa, M.L. (2002). Shortchanging the future of information technology: the untapped resource. *ACM Special Interest Group on Computer Science Education Bulletin*, *34*, 32–35.

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*(1), 3.

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*, 213–228.

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

Byrne, M. (2001). Sampling for qualitative research. *Association of periOperative Registered Nurses Journal*, *73*, 494–498, Feb.

Chou, C., & Tsai, M. (2007). Gender differences in Taiwan high school students' computer game playing. *Computers in Human Behavior*, *23*(1), 812–824.

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.

Cohoon, J.M. (2001). Toward improving female retention in the computer science major. *Communications of the ACM*, *44*, 108–114.

Fisher, A., Margolis, J., & Miller, F. (1997). Undergraduate women in computer science: Experience, motivation and culture. *SIGCSE Bulletin*, *29*(1), 106–110.

Frieze, C. (2007). *The critical role of culture and environment as determinants of women's participation in computer science* (Doctoral Thesis). Retrieved from ACM. (UMI Order Number: AAI3262816).

Garcia, D.D., Chapman, G., Hazzan, O., Johnson, M., & Sudol, L A. (2010, March). Rediscovering the passion, beauty, joy, and awe: making computing fun again, part 3. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education SIGCSE '10* (pp. 394–395). New York: ACM.

Goode, J., Estrella, R., & Margolis, J. (2006). Lost in translation: Gender and high school computer science. In J. McGrath Cohoon & W. Aspray (Eds.), *Women and information technology: Research on underrepresentation* (pp. 89–114). Cambridge, MA: The MIT Press.

Gürer, D., & Camp, T. (2002). An ACM-W literature review on women in computing. *ACM Special Interest Group on Computer Science Education Bulletin*, *34*, 121–127.

Ho, C., Slaten, K., Williams, L., & Berenson, S. (2004). Examining the impact of pair programming on female students. NCSU CSC Technical Report . Retrieved February 6, 2008, from ftp://ftp.ncsu.edu/pub/unity/lockers/ftp/csc_anon/tech/2004/TR-2004-20.pdf.

Kelleher, C., Pausch, R., & Kiesler, S. (2007). Storytelling Alice motivates middle school girls to learn computer programming. In *Proceedings of the SIGCHI conference on Human factors in Computing Systems (CHI'07)* (pp. 1455–1464). New York, NY: ACM.

Margolis, J., & Fisher, A. (2002). *Unlocking the clubhouse: Women in computing*. Cambridge, MA: The MIT Press.

McDowell, C., Werner, L., Bullock, H.E., & Fernald, J. (2003). The impact of pair programming on student performance, perception and persistence. In *Proceedings of the 25th international Conference on Software Engineering* (pp. 602–607). Washington: IEEE Computer Society.

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*, 81–84.

Merriam, S.B. (2009). *Qualitative research. A guide to design and implementation*. San Francisco, CA: Jossey-Bass.

Oliver, C. (2010, August 19). A level results 2010. *The Guardian*. Retrieved from http://www.guardian.co.uk/education/interactive/2010/aug/19/a-level-results-2010.

Sanders, J. (2005). *Gender and technology in education: A research review*. Retrieved October 14, 2011, from http://www.josanders.com/pdf/gendertech0705.pdf.

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*, 169–181.

Shashaani, L. (1997). Gender differences in computer attitudes and use among college students. *Journal of Educational Computing Research*, *16*(1), 37–51.

Shotick, J., & Stephens, P.R. (2006). Gender inequities of self-efficacy on task-specific computer applications in business. *Journal of Education for Business*, *81*, 269–273.

Simon, B., & Hanks, B. (2008). First-year students' impressions of pair programming in CS1. *ACM Journal on Educational Resources in Computing*, *7*, 5:1–28.

Tillberg, H.K., & Cohoon, J. McGrath. (2005). Attracting women to the CS major. *Frontiers*, *26*(1), 126–140.

Werner, L., Hanks, B., & McDowell, C. (2004a). Pair-programming helps female computer science students. *ACM Journal of Educational Resources in Computing*, *4*(1), 4.

Werner, L., McDowell, C., & Hanks, B. (2004b). Retaining women in computer science: The impact of pair programming. Project update ITWF/ITR workforce PI's meeting, Retrieved June 5, 2008, from http://www.cse.ucsc.edu/∼charlie/pubs/nsf_summary_ nov_2004.pdf.

Werner, L., Hanks, B., McDowell, C., Bullock, H., & Fernald, J. (2005). Want to increase retention of your female students? *Computing Research News*, *17*, 2.

Williams, L., & Kessler, R. (2002). *Pair programming illuminated*. Reading, MA: Addison-Wesley.

Williams, L., & Upchurch, R.L. (2001). In support of student pair-programming. *SIGCSE Bulletin*, *33*(1), 327–331.

Appendix 1

The table below indicates the questions used as guidelines for either the first interview or the second interview, or for both.

Nr.	Question	Interview# 1	Interview# 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?		●
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?		●