Are Females Disinclined to Tinker in Computer Science?

Samantha Krieger
Department of Computer Science
University of British Columbia
sfkrieger@gmail.com

Meghan Allen
Department of Computer Science
University of British Columbia
meghana@cs.ubc.ca

Catherine Rawn
Department of Psychology
University of British Columbia
cdrawn@psych.ubc.ca

ABSTRACT
Our study explores the use of tinkering strategies across genders in undergraduate students of Computer Science. We present a definition and a framework for tinkering as it applies to this context, and use these to investigate how students across genders tinker with hardware and software. Our goal is to examine gender-based differences in tinkering behaviour and to discuss the meaning and implications of these differences on teaching and learning in Computer Science. We gathered data via interviews and a questionnaire and used both qualitative and quantitative methods for analysis. Our work is preliminary and suggests further areas of research in this domain.

Categories and Subject Descriptors
K.3.2 [Computer and Information Science Education]: Computer Science Education; K.4.2 [Computer and Society]: Social issues

General Terms
Human Factors

Keywords
computer science education; tinkering; gender; programming

1. INTRODUCTION
Educators and technologists alike have noted the value of tinkering and exploratory play in learning and innovation [9, 13, 14]. However, cross-disciplinary data suggest that 'exploratory learning styles' or 'tinkering behaviours' are expressed more predominantly in males than females across a variety of populations [2, 12, 16]. In Computer Science (CS) literature this gendered tendency has been surmised of CS students [8, 10, 21], but never studied directly as it has in other fields.

As computers, like automobiles, lend themselves to being tinkered with, certain aspects of Computer Science may maintain a culture of tinkering as a dominant method of learning and instruction, both in the field at large and in programming specifically. This may exclude females unaccustomed or inexperienced in this style of learning, from both within and outside the discipline. However, it is plausible that the prevailing culture of CS may cause learners of both genders who enjoy tinkering to self-select and study Computer Science, reducing gender differences in tinkering within undergraduate CS populations.

Recent literature points to a lack of research on gender differences in programming and suggests it merits attention [20]. It is therefore important to investigate gender differences in tinkering behaviour so as to inform educational practices, and to explore whether it is a factor contributing to the gender gap in CS.

The focus of this study is to examine tinkering behaviours between genders on programming assignments and tasks in undergraduate students of Computer Science. Using interviews with CS professors and a questionnaire administered to undergraduate CS students, we attempt to gain a basic understanding of what tinkering may mean in this field. In particular, we seek to explore whether there are indeed gender-based differences in students' definitions of tinkering and in their use of tinkering behaviours in Computer Science.

Our contributions are: (1) a review of the current literature on tinkering behaviour across a variety of disciplines, (2) a preliminary definition and description of tinkering in Computer Science, (3) a test of the hypothesis that female Computer Science students are relatively less likely to exhibit tinkering behaviours than their male peers, as is suggested in [8, 10, 21], and (4) suggestions and implications for academia.

2. BACKGROUND
It is well established that Computer Science is a male dominated field with declining female enrolment and high rates of attrition among women. Despite rising female participation in Computer Science during the 1970s, females' enrolment in the major took a sharp decline in the late 1980s and has generally decreased since then [6]. A trend first documented in Camp's "The Incredible Shrinking Pipeline" [5], this pattern of enrolment has garnered a great deal of attention and has been aggressively studied since.

In his viewpoint article published in 2000, De Palma attributes the drastic rise and fall of female participation in the late 1970s and 1980s to the introduction of the microcomputer [8]. He argues that gender-based tinkering tendencies proliferated with the arrival of the microcomputer, were largely responsible for the shrinking pipeline of the 1980s and 1990s, and continue to contribute to the ongoing gender gap in Computer Science. De Palma appeals for a return to the logic and formal characteristics of Computer Science in an effort to remove the hardware (and software) 'noise' that he alleges may be distracting for both genders, but particularly women. De Palma's argument contains a compelling explanation for the original shrinking pipeline phenomenon of the 1980s and 1990s but carries a significant assumption of the tinkering behaviour of females, one that we will explore.

There is growing interest in the role and importance of microcontrollers and tinkering in technology. The 21st century saw the emergence and commercialization of single-board computers and microcontrollers such as the Raspberry Pi and

SIGCSE '15, March 4–7, 2015, Kansas City, MO, USA

http://dx.doi.org/10.1145/2676723.2677296

Arduino, tools initially designed as learning aids in the Computer Science classroom [19]. The presence of these tools also facilitated a culture of 'making' and semi-formal tinkering. In 2005 the concretization of this trend became clear in the establishment of "Make Magazine" and the "The Maker Movement" [9], a movement of self-proclaimed hobbyists and tinkerers focused on hardware and technology-based Do-It-Yourself (DIY) projects.

Appeals can now be heard to bridge at-home tinkering with in-school practices [14]. In fact, at the time of writing, Coursera has released its first course on tinkering called "Tinkering Fundamentals: A Constructionist Approach to STEM Learning", which attempts to " ... help educators and enthusiasts develop a practice of tinkering and making" [18]. Ultimately, tinkering and the use of gadgetry like the single-board computer appear to be on the rise and have popularized the notion of tinkering in both formal and informal realms.

In light of this, it seems particularly worthwhile to investigate tinkering behaviour of females and ask: (1) What is tinkering in Computer Science? (2) Are there gender-based differences in tinkering behaviours among students of CS? It is important to note that we are not attempting to make a judgment on whether tinkering tactics are beneficial when learning Computer Science. Our focus is on investigating if their application varies across genders. Although much of the related work describes tinkering as a useful problem solving technique, we contend its utility is context and application specific.

3. RELATED WORK

If there are gender-based differences in tinkering tendencies among Computer Science undergraduates, research in K-12 Science Education suggests that these practices are present from a young age. In a study on tool usage in the elementary school science classroom, Jones et al. report a difference in the way young girls follow instructions as they exhibit less play with instruments involved in science activities than boys [13]. Boys were more competitive, displayed less adherence to rules and directions, and used and manipulated tools in exploratory and inventive ways [13]. These findings are corroborated in a meta-study on major themes in gender in Science Education, which highlights related research on gender differences in exploratory behaviours and manipulation of science equipment [3].

In a study examining tinkering behaviours in the elementary Science classroom, Parsons uses both qualitative and quantitative approaches to investigate the intuitive methods students use and apply in learning physical science concepts [16, 17]. Parsons develops a model of tinkering behaviours providing a number of characteristics pertaining to different types of tinkering that can be observed. Ultimately, she argues that tinkering in most respects can be seen as a method more consistent with common male 'ways of knowing' over females.

Many of these results have been echoed in the field of Human Computer Interaction (HCI). A study on middle-school-aged students' reactions to and perceptions of information and communication technology shows girls exhibiting more goal-oriented behaviour with respect to their use and interest in technological devices, whereas boys tend to be interested in the construction and function of the devices [12].

Beckwith et al. investigated how gender differences interact with end-users' use of software features and end-user programming environments, such as spreadsheet software [1, 2, 24]. Their early findings demonstrated differences in the frequency and degree to which males and females explore new features [2], while their later work examined the strategies adopted by males and females when testing and debugging in more detail [2, 4]. Throughout their research Beckwith et al. discovered gender differences in exploratory behaviours, causing them to investigate the meaning of tinkering and its gendered application across environments.

In an effort to design interfaces that encourage females' exploratory behaviour, Beckwith et al. discovered meaningful differences in how males and females employed tinkering methods in both testing and debugging spreadsheets [2]. They defined tinkering as turning a feature rapidly on and off. In analyzing these patterns they measured tinkering frequency, number of tinkering episodes, and tinkering rate within episodes.

Across multiple studies the group found males tinkered more than females [2, 4]. However, males' tinkering was often found to be counter-productive to effective debugging, particularly in some specific sub-experiments [2]. By contrast, although females were found to tinker less, their approach appeared to be more effective in facilitating a better understanding of the features and led to more productive and successful testing and debugging. Beckwith et al. concluded that pausing and reflecting lay at the crux of effective tinkering for both genders and that this was important in successfully mastering new features of interfaces [2].

What much of this research shows is that there is indeed a female 'lack of proclivity' to tinker within a variety of settings. However, the literature on tinkering is young and diverse and how tinkering is conceptualized and studied across domains is context specific. For Beckwith et al., the term tinkering is used to describe users' actions and practices with interfaces, whereas the tinkering behaviours studied by Parsons in the elementary school classroom were undoubtedly different. In order to examine tinkering behaviours in Computer Science, we first consider the definition of the term in our field.

4. DEFINITION OF TINKERING

There are strong unifying conceptual similarities in how the term tinkering is used across disciplines. To develop a working definition for tinkering in Computer Science, we consider its overall conceptual meaning and a description of how tinkering is practiced in the Computer Science classroom. Our primary goal is to enumerate a number of tinkering-type behaviours that characterize tinkerers and non-tinkerers in order to test tinkering expression among students. We interviewed three Computer Science professors from the University of British Columbia and reviewed academic literature to establish these basic behaviours, in particular looking at tinkering as a method of *problem solving* in Computer Science.

Tinkering appears to be inextricably linked to exploration and exploratory behaviour. It is generally considered an informal practice, often with a purpose of improvement, and is commonly associated with experimentation, or 'trial and error' methods. As a problem solving technique and learning strategy it is often in contrast to formal, established, or prescribed methods.

One important distinction is that the term tinkering can apply to problem solving with both hardware and software. Arguably, the stereotypical view of tinkering in Computer Science is of tinkering with hardware. This vision of tinkering, employed by De Palma, often includes behaviours such as taking apart and building computers or computing equipment, and the use of microcontrollers. In investigating De Palma's assumptions we will start by assessing the integrity of these claims and investigating differences of 'hardware tinkering' across genders. However, we argue that to define tinkering solely with respect to hardware is of limited use in our discipline, and that tinkering is pervasive in software engineering as well.

Tinkering in Computer Science generally involves:

(1) Exploratory behaviours (e.g. examining parts of a code base that students are not required to modify or understand, addition of unrequired extra features on projects).

(2) Deviation from instructions (e.g. deviating from suggested methods of completion on labs and assignments).

(3) Lack of reliance on formal methods of learning and instruction (e.g. using Q&A websites and forums rather than suggested reference material or instructor's office hours).

(4) Use of trial and error techniques (e.g. heavy use of debuggers, using and testing code samples found online).

Taken collectively, we argue that the concert of these practices offers an initial description of tinkering in CS. The list is not meant to be complete. Rather, it should be considered an initial attempt at a categorization of tinkering behaviours.

4.1 Interest and Tinkering

The relationship of interest to tinkering is an important consideration. One might develop a practice of tinkering born from interest in the subject, or one could employ it as a pragmatic approach. Although the motivation causing students to adopt tinkering strategies is not our primary concern, accounting for gender-based differences in computer-related interests is an important part of this exploratory work.

5. THE STUDY

We used an online questionnaire asking students to self-report on a variety of tinkering behaviours. It also included two open-ended questions requesting that students provide a definition of tinkering in CS, and then self-classify as tinkerers or non-tinkerers.

5.1 Questionnaire

The questionnaire included 34 closed-ended questions across four categories.

(1) *General information* - academic and personal information such as students' age, major, gender, and courses completed. (7 items)

(2*) Interest and Experience* - types and extent of computer-related interests pursued, including extra-curricular time spent on a variety of computer-related activities such as programming and reading technology blogs or books. (9 items)

(3) *Tinkering in Hardware* - (de)construction of computing equipment and use of microcontrollers. (3 items)

(4) *Tinkering in Programming* - programming and problem solving practices in labs and assignments. At the University of British Columbia, labs are completed in an on-campus location, and are generally completed in the presence of teaching assistants. Assignments do not have the same venue association, are often less structured, and have a longer time frame for completion. This section includes questions such as frequency and use of the debugger, reliance on different types of reference material, and use of instructor or university provided resources. (15 items)

5.2 Participants

Students attending summer Computer Science courses at the University of British Columbia were recruited to participate in this study using in-class announcements. The study was also advertised in the weekly departmental undergraduate emails. Of the 107 students who completed the questionnaire, data from 93 respondents with a declared major in Computer Science or Computer Engineering were included for analysis. Thirty-three participants were female (35.5%), and 60 participants were male (64.5%). Incomplete responses and responses with no declared gender were not included in the analysis.

Participants ranged in year level with 29% in the first two years of the degree, 40.8% in third, and the remaining 30.2% in fourth or further. Participant age ranged from 17 to 32, with a mean age of 23 (*SD* 3.58). On average, participants had completed five of the eight required Computer Science courses.

6. RESULTS

The following is a summary of the results collected from the survey. The open-ended responses were analyzed using qualitative and quantitative methods.

6.1 Computer Interest and Experience

Females were more likely than males to report no programming experience before university (60% of females, 25% of males, $X^2(1) = 10.341$, $p = .001$). Males more frequently reported having already learned a programing language (26% of males, 10% of females, $X^2(1) = 3.876$, $p = .049$), and having played around with or read about computers before university (44% of males, 12% of females, $X^2(1) = 9.640$, $p = .002$). Males were no more likely to have participated in high school computer science classes or to have reported learning HTML/CSS before university.

On questions concerning computer usage and experience, males reported higher frequency and time spent on computer-related tasks or interests across all relevant questions. Males reported greater average frequency in programming on their own time ($t(85) = -4.67$, $p < 0.001$, $d = 0.98$), reading tech blogs or magazines or reading about technology for pleasure ($t(83) = -4.45$, $p < 0.001$, $d = 2.84$), and playing computer games ($t(66) = -2.38$, $p = 0.021$, $d = 0.52$).

These results are consistent with other studies that have reported a large discrepancy between programming and computer-related interest levels and experiences of male and female Computer Science students [e.g. 10,15]. This suggests our population is similar to those at other universities nationwide and globally.

6.2 Tinkering in Hardware

There were significant differences between male and female practices for both questions related to tinkering with hardware. 79% of male respondents have built or taken apart a computer or piece of computing hardware out of interest, in comparison with 30% of females ($t(60) = -5.15$, $p = <.001$). Additionally, 25% of males and 6% of females own or have made use of single-board computers or microcontrollers (such as the Arduino or Raspberry Pi) for non-academic purposes ($t(88) = -2.59$, $p = .011$).

6.3 Programming Practices

Students answered 15 questions about their problem solving practices in projects, labs, and assignments. The response options were presented in the form of a Likert-type scale with 4 options, ranging from 0 (*Never*) to 3 (*Always*). When asked about both labs and assignments, two items revealed significant differences between males and females. Males were more likely to change the code that is given (outside of what is required) ($t(77) = -3.23$, $p = 0.003$, $d = 0.70$). Females were more likely to ask for help from a TA or instructor during lab ($t(82) = 2.63$, $p = 0.010$, $d = 0.60$). We found no significant differences in exploring extra features to add, attempting the bonus question, using debuggers, using manuals or tutorials on related topics, looking for similar examples and trying to figure out how they work, looking for code snippets and trying to run them, or deviating from instructions.

When asked specifically about assignments, we found that females were again more likely to ask for help from a TA or instructor during their office hours ($t(83) = 3.23$, $p = 0.002$, $d = 0.71$). We found no statistically significant differences between genders for examining parts of the given code that you are not required to understand or change, following the suggested method in the assignment description, posting to the course discussion forum, or posting to external forums.

6.4 Student Definitions of Tinkering

Males' and females' responses to the questions "What does it mean to tinker with computers?" and "How would you define tinkering?" differed in content and style. Females generally used terms and concepts relating to the motivation or intangible purpose for tinkering, while males' responses often involved more details and descriptors of what tinkering with computers specifically entails.

50 out of 60 males (83%), and 26 out of 33 females (78%) completed responses to these questions. We devised a list of 17 concepts that appeared frequently in the responses. We masked gender and classified each response by assigning it to one or more categories, as appropriate. We then consulted with an external rater, and conducted a test for Inter-rater Reliability. A quarter of the responses (25%) were discussed collectively, the remaining 75% were scored by each rater individually, with an error between raters of < 5%. We then established a consensus for the categories that applied to each response.

In their descriptions, both males and females considered tinkering to involve 'playing around', 'experimenting', 'testing', and 'modifying' to see some sort of output, for a purpose of 'improvement'. Both groups describe a process of 'exploring', 'taking apart and building' with a purpose for 'understanding'. These words (or close synonyms) were the most commonly used, and were employed by both groups with each one appearing in at least 10% of valid responses from each gender.

Words and concepts more prevalent in females' responses were 'improvement' (in 23% of females' responses, 12% of males' responses), 'understanding', (30% F, 16% M), 'playing around' (35% F, 18% M), and 'exploration' (31% F, 22% M). This reflects a trend showing females tending toward descriptions centered on the conceptual purpose and motivation behind tinkering. When asked to define tinkering, two typical female examples were "to play with the computer and experiment with it", and "Trying things out for fun and to learn how something works".

Males tended to be more descriptive in their definitions of tinkering in *Computer Science*. Terms or concepts appearing more frequently in males' responses include 'modification' (28% male, 15% female), 'taking apart/building' (26% M, 15% F), 'customization' (12% M, 4% F), and 'deviation from intended purpose' (14% M, 4% F). Males often focused on the nature and substance of tinkering behaviours rather than giving a rationale for the behaviour. Males' responses were often very technical; multiple responses mentioned overclocking, and making setting/BIOS changes. None of these technical details were present in females' responses.

Furthermore, a large number of males explicitly made a point of including both software *and* hardware in their responses. Often males included separate descriptions of tinkering in hardware and software and provided examples of both (42% M, 19% F).

One notable example from the males' responses is the following:

> "To me, "tinkering" with a computer can be broadly defined as making it into something it was not at first. This could be interpreted in a physical sense ("tinkering" with the hardware to upgrade a computer, or assemble one "from scratch" using individual components, case-modding, etc.) or a software one (installing a new OS, programming new capabilities into it, modding and customizing an OS, setting up programs to work together for some purpose, etc.)."

As this response exemplifies, males often included examples of how tinkering is a process that allows the individual to 'customize' their experience and use the computer to 'cater to their needs'. Unlike females' responses, which more often tended to describe tinkering as 'exploratory', males' responses often suggested that tinkering is a goal-oriented behaviour. This behaviour might be initiated out of interest or curiosity, but facilitates the construction of something desirable (e.g. customized experience or improved performance).

It is important to underline that a number of terms or concepts appeared with similar frequencies between males and females, such as 'experiment' (11% female, 12% male), 'testing to see an output' (15% F, 12% M), and 'performance' (8% F, 8% M).

These findings are preliminary and require more investigation for verification. However, the responses do indicate that there may be a gendered understanding of the term 'tinkering' in CS.

There were several limitations in the data analysis. The number of responses from females was fewer by virtue of the number of questionnaire respondents of each gender. This likely had an effect on the breadth of the responses collected from females. Furthermore, we believe that participants' level of interest affected the nature and extent of their descriptions and interpretations of tinkering. As was mentioned previously, interest is a confounding variable that may affect not only tinkering behaviours but also students' descriptions of what tinkering means in CS. This should be considered in future studies on tinkering.

6.5 Student Self-Classifications

With empty responses removed, 35 out of 50 males (70%) said 'yes' or provided positive confirmation (e.g. 'sure', 'of course') to the question "Do you consider yourself a tinkerer?". Thirteen said 'no' or 'not really' (26%) and 2 (4%) had wavering responses (e.g. 'somewhat' or 'sometimes'). Among the 33 females, with 7 empty responses removed, 4 self-classified as tinkerers (15.4%), 12 did not (46%), and 10 (38.4%) provided wavering responses.

7. DISCUSSION

Our data show that males and females report some different tinkering behaviours, self-classify as tinkerers at different rates, and may have different perceptions of the tinkering process. The results demonstrate that in certain ways females as a group tend to tinker less, and do not consider themselves tinkerers. These findings support the theories of De Palma and corroborate behaviours suggested by other authors [8, 10, 21].

Our purpose was not to examine gender differences in interests in computing, or frequency at which males and females utilize computer hardware. Our goal was to explore more generally the presence or absence of gender differences in the application of particular problem solving skills both within and outside the Computer Science classroom. Collectively, we recognized these behaviours as tinkering. In pursuing and investigating 'tinkering type' behaviours we have observed certain problem solving practices that vary along gender lines.

Our assumption is that tinkering is a pursuit that applies to both software and hardware. Furthermore, we contend that if an individual develops a practice of tinkering in one area, this methodology likely pervades their practices in the other. In other words, the methodologies learned from tinkering when solving hardware problems might have implications in how students problem solve, complete assignments, and appeal for help when programming at work, at home, or in the classroom.

Risk taking and confidence are fundamental to the act of tinkering, but are found to be different across genders. We speculate that these factors may contribute to the results we have seen thus far. Moreover, previous research has found gender differences in students' goals in the pursuit of a CS degree [15]. We argue that these differences in perspective on the purpose and utility of a CS major may be reflected in students' perception of the tinkering process as well.

7.1 Risk Taking

Burnett et al. found that tinkering involves aspects of risk taking both with things that are tangible (e.g. expensive equipment) and intangible (e.g. time). In Computer Science, tinkering with a computer could mean the risk of breaking physical or software components, rendering them unusable, or the risk of lost time spent on non-fortuitous computer-related pursuits. Copious amounts of research in the fields of Psychology and HCI have shown large differences in perception of risk and in risk taking behaviours between genders [e.g. 4, 11].

Although a number of males specifically mentioned 'breaking things' in their descriptions of tinkering, females only mentioned this when classifying themselves as non-tinkerers. A female's telling response to the question, "Do you tinker with computers?" was "No, only because I've almost broke [sic] something previously. I am afraid of actually doing something that I cannot reverse." This sentiment was echoed in other females' responses. It is one of many potential reasons for females' inexperience in the construction or deconstruction of computer-related equipment.

Although risk-taking behaviour was not mentioned by our participants in relation to software, we extrapolate that differences in risk taking may be present in programming practices. As four (8%) male students pointed out, tinkering with software also requires 'breaking' code. Although the effects are generally felt differently than with hardware, the tendency to 'break' working source code or aggressively approach programming may not be a tendency that females employ to the same degree as males. An openness to break things may prove to be an efficient strategy on some programming tasks. Thus, aversion to risk could pose a problem for females saddled with a number of different programming or computer-related assignments, all with poor or limited reference material that merely require a little 'playing around'. To males employing or experienced in tinkering practices this exploratory mode may be a more trivial endeavor.

7.2 Confidence in Applied Problem Solving

Research in mathematics shows females performing more poorly when given application-oriented problems, but shows good performance on the theoretical correlates of the same problems [22]. Males do not exhibit this same discrepancy. Vermeer et al. argue that skills needed to solve applied problems go beyond cognitive skills, and that affective variables such as low confidence in females may inhibit problem solving behaviours in these applied mathematical problems [22]. We speculate that this is an issue in Computer Science, and in the utilization of applied problem solving techniques, such as tinkering as well.

Gender-based discrepancies in confidence may have inescapable effects on a number of different behaviours in learning Computer Science. This might leave females struggling with applied problems or applied problem-solving strategies like tinkering. Explicit instruction in applied work through the practice of 'formal' tinkering may help increase females' confidence or acquaint them with alternative applied avenues for learning.

7.3 Goal Orientation

In their seminal work on the gender gap in Computer Science, Fischer, Margolis, and Miller outline the difference in female and male narratives of motivation for and pursuit of a CS degree [10].

> "When the first-year females talk about their personal history with computers, their narratives are not filled with long and detailed accounts... They contextualize their interest in computer science, instead, within a larger purpose: what *they can do in the world.* " [10]

Although this is in reference to reasons for selecting and pursuing a CS major, the description relates to our results of males' and females' definitions of tinkering in CS.

Analogous to the findings of Fischer et al. our results showed a difference between genders in the style and nature of the definitions of tinkering. Female students appeared generally more concerned with the purposes and goals of tinkering (namely in how it facilitates understanding), its general purpose for improvement, and focusing on tinkering as an exploratory pursuit. This was in contrast to the males' responses that showed more emphasis on the details of tinkering and greater preoccupation with what tinkering entails.

As females appear to see tinkering more as an open-ended pursuit, devoid of a tangible purpose, they may be less apt to tinker. On the other hand, since males may tend to recognize its concrete, less general purposes (such as improved performance or customization), they may be more equipped and inclined to tinker at home. Ultimately, if females viewed tinkering as having tangible, meaningful effects, with specific ideas of what it entailed, they may be more inclined to tinker.

8. SUGGESTIONS AND IMPLICATIONS

The implications of these results are potentially far reaching. Here we highlight some key issues and present preliminary suggestions to benefit both tinkering and non-tinkering learners.

8.1 Distribution of Course Resources

As our results show, females are more likely than males to report asking for help from TAs and instructors both in labs and during office hours. Moreover, the use and reliance on more formal methods of instruction in Computer Science has been documented in other related research on gender differences in CS [7]. This research reports females dominating extra tutorial sessions offered by the department, even when computer experience and programming performance are controlled for [7].

In courses where resources are limited, a scarcity of office hours could be a problem for females preferring this mode of learning and assistance. Ensuring the presence of accessible course staff members may be important to both females and non-tinkering males alike.

8.2 Design of Class Materials

The design of assignments and reference material may also affect males and females differently. Our results show a discrepancy between genders in the exploration of supplied assignment code that students are not required to modify. When exploratory behaviour is deemed useful on an assignment or lab, the inclusion of instructions outlining exploratory techniques could benefit those unaccustomed to tinkering. This modification may provide the potential educational benefit of tinkering to both non-tinkering males and females alike.

Ultimately, redesigning course materials for tinkering and non-tinkering types may mean providing stepping-stones to teach tinkering for non-tinkerers. To do so effectively we need to first determine whether tinkering behaviours are useful for students and, if they are, how to encourage these behaviours.

9. FUTURE WORK

Further research is needed to investigate the complexities of these findings and to explore the implications of gender-based differences in tinkering. Although the current maker movement encourages tinkering as a problem solving technique, it is important to consider how, when, and if tinkering is appropriate. We need to consider types of tinkering techniques and strategies, situations in which these techniques and strategies are expressed, and when they may or may not be useful. We can then study the effect of different types of tinkering strategies on academic performance, confidence, and comprehension.

10. CONCLUSION

Our study confirms some of the conjectures on the tinkering behaviours of females with respect to male Computer Science students. Indeed, female Computer Science students do tinker less with hardware, and may show disinclinations to tinker with software as well.

In this exploratory work we have highlighted a number of gender-based tendencies in tinkering behaviours across a variety of computing practices. We investigated gender-based tinkering behaviours in Computer Science not simply as they relate to hardware, but as a pragmatic approach to problem solving in Computer Science overall.

11. ACKNOWLEDGMENTS

We would like to thank the following people for helping us with this project: Patrice Belleville, Paul Carter, Anne Condon, Maria Klawe, Dena Krieger, Alisa Lipson, Kyle Melnick, Gail Murphy, Michele Ng, and Steve Wolfman. We would also like to thank the UBC Computer Science Department for its support, the anonymous reviewers for their helpful suggestions, and the Maria Klawe Endowment fund for funding Samantha's travel to SIGCSE.

12. REFERENCES

[1] Beckwith, L. and Burnett, M. 2004. Gender: An Important Factor in End-User Programming Environments? In *2004 IEEE Symposium on Visual Languages and Human Centric Computing*, 107–114.

[2] Beckwith, L., Kissinger, C., Burnett, M., Wiedenbeck, S., Lawrance, J., Blackwell, A., and Cook, C. 2006. Tinkering and Gender in End-user Programmers' Debugging. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. New York, NY, 231–240.

[3] Brotman, J. S. and Moore, F. M. 2008. Girls and science: A review of four themes in the science education literature. In *J. Res. Sci. Teach.* 45 (Nov. 2008), 971–1002.

[4] Burnett, M., Wiedenbeck, S., Grigoreanu, V., Subrahmaniyan, N., Beckwith, L., and Kissinger, C. 2008. Gender in End-user Software Engineering. In *Proceedings of the 4th International Workshop on End-user Software Engineering*. ACM. New York, NY, 21–24.

[5] Camp, T. 1997. The Incredible Shrinking Pipeline. In *CACM*. 40, 10 (Oct. 1997), 103–110.

[6] Camp, T. 2002. The Incredible Shrinking Pipeline. In *SIGCSE Bull*. 34, 2 (Jun. 2002), 129–134.

[7] Carter, J. and Jenkins, T. 1999. Gender and Programming: What's Going on? In *Proceedings of the 4th Annual SIGCSE/SIGCUE ITiCSE Conference on Innovation and Technology in Computer Science Education*. ACM. New York, NY. 1–4.

[8] De Palma, P. 2001. Viewpoint: Why Women Avoid Computer Science. In *CACM*. 44, 6 (Jun 2001), 27–30.

[9] Dougherty, D. 2012. The Maker Movement. In *Innov. Technol. Gov. Glob.* 7, 3, 11–14.

[10] Fisher, A., Margolis, J., and Miller, F. 1997. Undergraduate Women in Computer Science: Experience, Motivation and Culture. In *Proceedings of the Twenty-eighth SIGCSE Technical Symposium on Computer Science Education*. ACM. New York, NY, 106–110.

[11] Gustafson, P. E. 1998. Gender Differences in Risk Perception: Theoretical and Methodological Perspectives. In *Risk Anal.* 18, 6 (Dec. 1998), 805–811.

[12] Hou, W., Kaur, M., Komlodi, A., Lutters, W. G., Boot, L., cotton, S. R., Morrell, C., Ozok, A. A., and Tufekci, Z. 2006. 'Girls Don't Waste Time': Pre-adolescent Attitudes Toward ICT," in *CHI '06 Extended Abstracts on Human Factors in Computing Systems*. ACM. New York, NY, 875–880.

[13] Jones, M. G., Brader-Araje, L., Carboni, L. W., Carter, G., Rua, M. J., Banilower, E., and Hatch, H. 2000. Tool time: Gender and students' use of tools, control, and authority. *J. Res. Sci. Teach.* 37, 8 (Oct. 2000), 760–783.

[14] Lamers, M. H., Verbeek, F. J., and van der Putten, P. W. H. 2013. Tinkering in Scientific Education. In *Advances in Computer Entertainment,* D. Reidsma, H. Katayose, and A. Nijholt, Eds. Springer International Publishing, 568–571.

[15] National Center for Women & Information Technology. 2012. Girls in IT. *The Facts. National Center for Women & Information Technology*. Retrieved August 7, 2014 from http://www.ncwit.org/resources/girls-it-facts.

[16] Parsons, D. S. 1995. Making sense of students' science: The construction of a model of tinkering. *Res. Sci. Educ.* 25, 2 (Jun 1995), 203–219.

[17] Parsons, S. 1990. *The development of a model of tinkering□: a study of children's science*. Doctoral Thesis. University of British Columbia.

[18] Petrich, M., Wilkinson, K., and Anzivino, L. 2014. Tinkering Fundamentals: A Constructionist Approach to STEM Learning. *Coursera*. Retrieved August 1, 2014 from www.coursera.org/course/tinkering.

[19] *Raspberry Pi*. Retrieved August 26, 2014 from http://www.raspberrypi.org/about/.

[20] Sheard, T., Simon, S., Hamilton, M., and Lönnberg, J. 2009. Analysis of Research into the Teaching and Learning of Programming. In *Proceedings of the Fifth International Workshop on Computing Education Research Workshop*. ACM New York, NY, 93–104.

[21] Tillberg, H. K. and Cohoon, J. M. 2005. Attaching Women to the CS Major. *Front. J. Women Stud.* 26, 1, 126–140.

[22] Vermeer, H. J., Boekaerts, M. and Seegers, G. 2000. Motivational and gender differences: Sixth-grade students' mathematical problem-solving behaviour. In *J. Educ. Psychol.* 92, 2, 308.