In Their Own Words: Gender Differences in Student Perceptions of Pair Programming

Kimberly Michelle Ying
University of Florida
Gainesville, Florida
kimying@ufl.edu

Lydia G. Pezzullo
University of Florida
Gainesville, Florida
lpezzullo@ufl.edu

Mohona Ahmed
University of Florida
Gainesville, Florida
mohonaahmed@ufl.edu

Kassandra Crompton
University of Florida
Gainesville, Florida
kcrompton@ufl.edu

Jeremiah Blanchard
University of Florida
Gainesville, Florida
jblanch@cise.ufl.edu

Kristy Elizabeth Boyer
University of Florida
Gainesville, Florida
keboyer@ufl.edu

ABSTRACT

Women continue to be underrepresented in computer science. Previous research has identified factors that contribute to women's decisions to pursue computing-related majors, but in order to truly address the problem of underrepresentation, we need to develop a deeper understanding of women's experiences within computer science courses. Pair programming is demonstrably beneficial in many ways, and we hypothesize that there are gender differences in student perceptions of this widely used collaboration framework. To explore these differences and move toward a thorough understanding of students' experiences, this paper investigates students' written responses about their experiences with pair programming in a university-level introductory computer science course. Using thematic analysis, we identified overarching themes and distinguished between what men and women reported. Both women and men wrote about their overwhelmingly positive perceptions of pair programming. Women often mentioned that pair programming helps with engagement, feeling less frustrated, building confidence, and making friends. Women also noted that it is easier to learn from peers. These findings shed light on how pair programming may lower barriers to women's participation and retention in computing and inform ongoing efforts to create more inclusive spaces in computing education.

CCS CONCEPTS

• **Social and professional topics → Computing education**; **CS1**; **Adult education**;

KEYWORDS

Gender; Pair programming; Undergraduate education

SIGCSE '19, February 27-March 2, 2019, Minneapolis, MN, USA

ACM ISBN 978-1-4503-5890-3/19/02…$15.00
https://doi.org/10.1145/3287324.3287380

ACM Reference Format:
Kimberly Michelle Ying, Lydia G. Pezzullo, Mohona Ahmed, Kassandra Crompton, Jeremiah Blanchard, and Kristy Elizabeth Boyer. 2019. In Their Own Words: Gender Differences in Student Perceptions of Pair Programming. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19), February 27-March 2, 2019, Minneapolis, MN, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3287324.3287380

1 INTRODUCTION

The need for qualified talent in STEM is high and projected to increase [22]; yet women continue to be underrepresented in these fields [23]. While some areas in STEM have shown good response to efforts toward gender equity, computing lags behind, with women representing just 26% of the computer and information sciences workforce and African American and Latina women representing less than 16% [23]. This disparity is in place by the time prospective professionals graduate: Of all graduates from computer science and computer engineering bachelor's programs in the United States, just 18% are women [24].

Many women are discouraged from pursuing education and careers in computing for social reasons: they may not feel a sense of belonging and identification with the field [9], or they may desire more social support from peers than the class style allows [6]. Promising work has already identified practices that address these roadblocks and improve women's interest and persistence, such as group learning [10, 12] and pair programming [15]. The pair programming paradigm, where two programmers work together on the same code, not only produces higher quality code and promotes a more efficient work process [8, 13, 18], it also offers structured collaboration with peers [16, 17] and improves interest and retention for undergraduates, including women [14, 15].

As a matter of equity in opportunity, research is needed to understand and address the causes of the gender gap in computing. Quantitative research has been used to evaluate pedagogical approaches [10, 12] and investigate hypotheses about students' attitudes, beliefs, and traits [19, 20] as they pertain to this issue. Qualitative work offers a human-focused perspective into the unique experiences of individuals: for example, it has characterized the social contexts of computer science learning [11] and contrasted the attitudes and opportunities of women both in CS and who have left the field [9, 21]. Thematic analysis of written data is a promising qualitative approach because it captures details of individuals' unique reflections and uses these to identify important factors in the phenomenon of interest. The technique has been used to characterize perceptions of pair programming in general [4], but has not yet been used to contrast the firsthand accounts of male and female students in pair programming activities.

This paper advances the field by identifying potential factors influencing gendered perceptions of computer science coursework at the undergraduate level. It does so through a qualitative analysis of 104 open-ended survey responses from students in an introductory course for computer science majors. The central question asked is, **what differences do we observe in men's and women's perceptions of pair programming?** By identifying themes in students' perceived advantages and disadvantages of pair programming and examining which students reported these themes, this study supports hypothesis-forming for future work on advancing equal representation in computing fields.

2 RELATED WORK

The factors impacting women's interest [5, 19] and persistence [9, 21] in computing-related majors are predominantly cultural and societal. In an interdisciplinary critical review, Cheryan and colleagues [7] point to the intersection of a masculine culture and a lack of opportunities for early exposure as the key roadblocks to girls' and women's interest in pursuing computer science. Qualitative research indicates that a supportive peer community plays an important role in persistence and identity formation in computing [9, 21]. Cultural perceptions of computer science frame the field as solitary work, leading learners to believe that they will not find sufficient support from their peers in the classroom [6]. Moreover, underrepresentation perpetuates itself by leaving women who do choose computing careers feeling alienated and vulnerable to stereotype threat and impostor syndrome [21].

Pedagogical strategies have the potential to mitigate these problems. Pair programming, a widely-implemented practice in both classrooms and professional environments, is one such intervention. In pair programming, two coders share one computer: they take turns in the role of driver, who operates the computer, or navigator, who is free to read the code and suggest actions. Students working in the pair programming paradigm demonstrate improved code quality, and they work more efficiently [8, 13, 18]. The structured collaboration that pair programming offers may be the reason it can lead to better course retention than individual programming work alone [16, 17]. Prior research has evaluated excerpts related to pair programming from CS1 students' open-ended, end-of-semester reflection papers and found themes of a cognitive, affective, and social nature [4], but to our knowledge has not taken a qualitative approach to understanding the unique experiences of men and women in CS1 pair programming. This study, which takes as data student responses to open-ended questions directly inquiring about pair programming, expands on previous work by comparing the themes that predominated among female versus male respondents. It could be expected that women might appreciate pair programming in more or different ways than men. Our work explores this possibility by looking at patterns in the advantages and disadvantages of pair programming that men and women reported during an introductory computer science course.

Table 1: Prompts included as part of the students' quiz.

(1)	How does your experience in the class so far compare to what you were expecting?
(2)	Describe your relationship with the partner you had for labs 3-5 (Selection, Hex Decoder, Debugging Version Control).
(3)	In what ways do you think pair programming impacted your learning?

3 DATA AND METHODS

3.1 Course Context and Participants

This study was conducted within an introductory computer science class at a large public university in the Southeastern United States in Spring 2018. The course was taught using the Java programming language and was a required course for majors in computer science and computer engineering. Students had lecture twice a week for a total of 150 minutes of instruction. Labs, each administered by a teaching assistant (TA), were held once each week for two hours in classes of 25 or fewer students. There were 411 students enrolled in the course, 29.7% of whom were women. 187 students consented to the collection of their course data for research purposes. Consenting students did not receive any incentives for their participation, and there were no penalties for students who did not participate in the study. The demographic survey included the question, "I identify my gender as", with the options of "Female", "Male", "Other", and "Prefer not to answer". 129 of the consenting students reported their gender in the demographic survey, and 106 of those students submitted responses to the prompts regarding pair programming.

After removing two blank prompt responses, there were 104 prompt responses analyzed in total. The authors of those responses were 38 women (36.5%) and 66 men (63.5%). None of the participants selected "Other" or "Prefer not to answer" for the gender identity question. The race/ethnicities of the participants were White/Caucasian (40.4%), Asian/Pacific Islander (18.3%), Multiracial (16.3%), Hispanic/Latino (14.4%), Black/African-American (7.7%) and unspecified (2.9%). The students' majors were Computer Science (34.6%), Computer Engineering (19.2%), other Engineering (25%), and other (21.2%). Levels of prior experience differed markedly by gender, with 59.1% of male participants reporting prior experience programming and only 18.4% of female participants reporting programming experience before the course.

3.2 Procedure

At the beginning of the third lab session, students were introduced to pair programming and assigned partners with whom they would work for the next three lab sessions. Students were paired based on consent to the study and based on gender (to ensure there were some female-female pairs, male-male pairs, and mixed pairs). Due to student absences, some students may have had more than one partner throughout the three lab sessions. Additionally, if lab sections had an odd number of students, a trio was formed so that no student worked alone. Assignments for labs 3 and 4 involved writing a program and were conducive to pair programming. Lab 5 was a tutorial and did not involve submitting any code. For this lab, students worked on their own computers but referred to their partners if they needed help. After completing their fifth lab, students were given three short-answer prompts (Table 1) to complete at home as part of their weekly quiz grade. A minimum of 100 words was required for each prompt, and students were informed they would be graded only for completion. This paper reports on responses to prompts 2 and 3, which inquire about the students' feelings towards pair programming and their partners during lab sessions.

3.3 Data

There were 106 submissions for the prompts from consenting students who reported their gender in a demographic survey. From those submissions, two were removed: one whose responses to prompts 2 and 3 were blank, and the other whose response to prompt 3 was blank. The remaining 104 responses had an average word count of 114 for prompt 2 and 118 for prompt 3, with an aggregate average of 232 words. The minimum response length for prompt 2 was 11 words, while the maximum length was 232 words. Prompt 3 had a minimum response of 14 words and a maximum response of 305 words. In aggregate, the shortest response was 25 words long and the longest response was 485 words long.

3.4 Methods

Following the qualitative research methodology of thematic analysis, three researchers collaboratively coded the positives and negatives of pair programming mentioned in each student's aggregate response. Thematic analysis is a method used for organizing qualitative data into themes which relate back to the research question [2]. We used a bottom-up approach, allowing our data to directly inform the themes based on codes which were created to summarize the students' main points. The process was as follows: (1) Researchers independently read the student's response and recorded every positive and negative aspect of pair programming mentioned; (2) Researchers discussed the positives and negatives they recorded for that student until reaching a consensus; (3) Any positives and negatives not previously mentioned were added to a spreadsheet as a new code. To perform the thematic analysis, the codes were grouped by similarity and then themes were determined by summarizing the codes within each grouping. Any themes which overlapped extensively were combined, and grouping was complete once the three researchers agreed that the remaining themes were distinct from each other.

4 RESULTS

Thematic analysis resulted in 52 codes, of which 31 were positive and 21 were negative. These codes were organized into 16 themes: seven positive themes and nine negative themes, as shown in Tables 2 and 3, respectively. Due to the interrelated nature of the topics and phenomena discussed in the responses, there is some overlap between the themes reported. Similarly, some codes may fit within more than one theme but are reported under the theme which the researchers felt was most relevant.

4.1 Positive Perceptions of Pair Programming

From the 104 student responses, 98 students expressed at least one positive sentiment about pair programming, for a total of 334 positive sentiments. Seven themes emerged from these reported benefits, as shown in Table 2: *improve learning experience*, *career skills*, *positive atmosphere*, *networking*, *efficient/productive*, *more engaged*, and *personal gain*.

Table 2: Reported positives and their corresponding theme with counts and percentages of men and women who mentioned each code within their response.

Positives		Count (Percent)	
Theme	**Original Code**	**Men**	**Women**
IMPROVE LEARNING EXPERIENCE	*Improves understanding and learned new things*	24 (36.4%)	20 (52.6%)
	Broadens problem-solving techniques / Different approaches	18 (27.3%)	12 (31.6%)
	Someone to ask questions and discuss ideas	37 (56.1%)	19 (50.0%)
	Receive feedback	2 (3.0%)	0 (0.0%)
	Reinforce knowledge by explaining to partner	8 (12.1%)	2 (5.3%)
	Better recollection of material for the future	2 (3.0%)	0 (0.0%)
	Easier to learn from peers	0 (0.0%)	2 (5.3%)
	Introduced to new resources	1 (1.5%)	0 (0.0%)
CAREER SKILLS	*Improves coding*	8 (12.1%)	3 (7.9%)
	Developing workplace skills	11 (16.7%)	5 (13.2%)
	Learn to work together / Working with others	15 (22.7%)	8 (21.1%)
	Open to new suggestions	0 (0.0%)	1 (2.6%)
POSITIVE ATMOSPHERE	*More enjoyable*	5 (7.6%)	4 (10.5%)
	More comfortable	0 (0.0%)	1 (2.6%)
	Helps not feel discouraged / Less frustrating / Less stressful	4 (6.1%)	9 (23.7%)
	Friendlier atmosphere	1 (1.5%)	0 (0.0%)
	Comforting / Reassuring / Supportive	4 (6.1%)	5 (13.2%)
NETWORKING	*Help from classmates outside of class*	3 (4.5%)	1 (2.6%)
	Meet new people / Make friends	1 (1.5%)	4 (10.5%)
	Improve social/communication skills	9 (13.6%)	3 (7.9%)
EFFICIENT / PRODUCTIVE	*Efficient*	24 (36.4%)	7 (18.4%)
	Identifying errors	14 (21.2%)	5 (13.2%)
	Easier to seek help from TA	2 (3.0%)	1 (2.6%)
	Easier	7 (10.6%)	3 (7.9%)
	Require less help from TA	1 (1.5%)	1 (2.6%)
MORE ENGAGED	*Motivated by partner*	3 (4.5%)	3 (7.9%)
	Greater responsibility	1 (1.5%)	2 (5.3%)
	More engaged/focused with work	1 (1.5%)	1 (2.6%)
PERSONAL GAIN	*Feels good to help someone*	2 (3.0%)	0 (0.0%)
	Builds confidence	0 (0.0%)	3 (7.9%)
	More rewarding	1 (1.5%)	0 (0.0%)

4.1.1 Improve Learning Experience. The most common theme observed was improving the learning experience, with 86 student responses mentioning at least once that pair programming improved their learning experience. Note that this theme may have been primed by the wording of prompt 3, resulting in the high number of student responses under this theme. Specifically, the most common codes from this theme (and overall) were *someone to ask questions and discuss ideas*, with over 50% of both men and women mentioning this benefit, and *improves understanding and learned new things*, with over 40% of students mentioning this benefit and over 50% of women in particular. On more occasions than women, men reported the benefit of reinforcing their knowledge by explaining to their partner. *Receive feedback*, *better recollection of material for the future*, and *introduced to new resources* were codes derived exclusively from male students' responses. On the other hand, women exclusively wrote that it was easier to learn from peers, with 5.3% reporting this perception. One woman wrote:

> "...it's always been intimidating or embarrassing for me to ask for help from people that have much more experience than myself. It's nice having someone at my level to brainstorm with, and they're quickly accessible in comparison with a TA that is responsible for the entire lab..."[1]

Another woman wrote:

> "Having someone closer to my age than a TA makes it less scary to ask questions and also provides me with a constant form of encouragement."

4.1.2 Career Skills. Men and women reported improvements to career skills at similar rates, with over 20% citing the benefits of learning to work together, and 15.4% specifically mentioning developing their workplace skills. One woman wrote:

> "I think paired programming is a great idea because it helps people learn how to work together on a program. All of these skills are needed in the work place and will definitely impact our future careers."

4.1.3 Positive Atmosphere. Women attributed pair programming leading to a positive atmosphere, specifically by reducing negative emotions like discouragement, frustration, and stress and by creating an environment that was comforting, reassuring, and supportive. Men did not report this positive aspect of pair programming as much as the women in this study. These women described the relief they felt from pair programming:

> " Without a partner I am not sure if I would have been able to complete the labs at all. Perhaps, it wouldn't have been impossible, but it would have definitely been stressful and maybe so much that I don't learn at all because I'd be focused on trying to complete the lab only instead of understanding what is going on."

> "I would feel incredibly overwhelmed if I had to complete every lab alone. Having a friend by me to lean on when I have a question is very comforting and makes me feel as though I can program better. [...] I really hope we continue to do pair programming as it has been a huge stress reliever as opposed to doing them separately."

4.1.4 Networking. Women reported meeting new people and making friends as a benefit of pair programming more often than men, which points to its potential for mitigating the feelings of isolation many women experience.

> "It is also very nice to have conversation with someone during the lab and makes it more enjoyable. I also like meeting people in the class because without the pair programming, I would probably not have branched out to converse with others."

4.1.5 Efficient / Productive. Men commented on improvements to the quality of their code and coding process, citing greater efficiency and better identification of errors more often than did women. One man wrote:

> "At first when the students in my lab were all paired with each other, I thought it might slow our learning. After a few lab projects I started to realize that it was actually the exact opposite."

Another man wrote:

> "I would much rather have a partner than not have a partner because I don't have to ask the TA for help as often and can get the work done faster."

4.1.6 More Engaged. While engagement was not referenced frequently by any students, women mentioned the theme of engagement more often than men. One woman described being more motivated during lab:

> "His own motivation to successfully complete the lab helps me become more motivated to get the job done."

Another woman described how her motivation to do well in the course extends past the actual time spent pair programming:

> "I also feel the need to study more which is a good thing because I don't want to be looked at as the 'weak link' of the pair so it definitely motivates me to try and understand more and prepare for the labs more."

4.1.7 Personal Gain. Interestingly, while few students commented on personal gain, three women (7.9%) said specifically that pair programming boosted their confidence while zero men reported this.

> "I think pair programming has provided me with everything I need to gain a little bit of confidence and not give up or get frustrated so easily when I can't get my code to run properly."

4.2 Negative Perceptions of Pair Programming

From the 104 student responses, 31 students expressed at least one negative sentiment about pair programming, for a total of 56 negative sentiments. While student perceptions were positive overall, it is important to investigate the negative perceptions to determine how we might prevent these feelings in the future. Nine themes emerged from these reported disadvantages as shown in Table 3: *unequal workload*, *inefficient*, *learn more alone*, *incompatible partner*, *less hands on*, *less convenient*, *frustrating*, *one person's style takes over*, and *relying on someone else*. Among these negative themes, a few patterns were noticeable.

4.2.1 Predominantly female sentiments. Both men and women discussed an incompatible partner as a detriment to their pair programming experience, with women reporting it more frequently. This was most often cited as simply "incompatibility", but two women specifically said they felt they were burdening their partner, while

[1] All quotes are taken directly from the students' responses and preserve any typos or grammatical errors.

Table 3: Reported negatives and their corresponding theme with counts and percentages of men and women who mentioned each code within their response.

Negatives		Count (Percent)	
Theme	**Original Code**	**Men**	**Women**
UNEQUAL WORKLOAD	*Unequal workload (partner taking over)*	1 (1.5%)	1 (2.6%)
	Unequal workload (partner freeloading)	3 (4.5%)	0 (0.0%)
	Slowed learning	1 (1.5%)	0 (0.0%)
	Inefficient/Slower	5 (7.6%)	3 (7.9%)
	Work better/faster alone	4 (6.1%)	1 (2.6%)
LEARN MORE ALONE	*Hinders independent learning*	0 (0.0%)	1 (2.6%)
	Harder to complete later on your own	0 (0.0%)	1 (2.6%)
	Learn better alone	4 (6.1%)	2 (5.3%)
	Only learns part of material	2 (3.0%)	1 (2.6%)
	Difficult to focus	0 (0.0%)	1 (2.6%)
	Learning is more difficult	0 (0.0%)	1 (2.6%)
INCOMPATIBLE PARTNER	*Incompatible partner*	1 (1.5%)	2 (5.3%)
	Discouraged by partner	0 (0.0%)	1 (2.6%)
	I am burdening partner	0 (0.0%)	2 (5.3%)
	Partner is burdening me	1 (1.5%)	0 (0.0%)
	Partners are not obligated to teach	0 (0.0%)	1 (2.6%)
LESS HANDS-ON	*Less hands-on*	3 (4.5%)	2 (5.3%)
LESS CONVENIENT	*Less convenient*	3 (4.5%)	0 (0.0%)
FRUSTRATING	*Frustrating*	2 (3%)	2 (5.3%)
ONE PERSON'S STYLE TAKES OVER	*One person's style takes over*	1 (1.5%)	1 (2.6%)
RELYING ON SOMEONE ELSE	*Relying on someone else / Responsibility*	2 (3.0%)	0 (0.0%)

no men reported this sentiment. One woman expressed her frustrations with having a partner that did not care to explain things to her:

> "But, in my experience, is also frustrating to work with someone who is at a different level of understanding of the subject matter than you are. I understand how they may not feel obligated to try to explain things but it also does not help those that are behind to work with someone further along and who is unwilling to explain anything. And of course our partners are just students so it isn't really their responsibility to teach."

Men and women both reported they felt they learned more alone, but men gave almost no additional reasons for this, while a few women elaborated on this point, citing difficulty focusing and challenges in transferring skills to independent work.

4.2.2 Predominantly male sentiments. A few men perceived pair programming to be less convenient or that their partner was freeloading, but no women reported these issues. One student expressed the freeloading as only hindering his experience when the labs were more difficult:

> "I think for the easier labs, paired programming has helped me learn easier because I can do the coding for part of the lab and then give the computer to my partner and watch and learn from him. However, during the harder labs, it feels as if paired programming is pointless in my case because my partner is worse at coding then I am and thus whenever it is his turn to code, it is still just me telling him what to write."

One of these students had only negative things to say about pair programming:

> "I honestly think pair programming is stupid. I am always more prepared than my partner, and I end up teaching them the concepts. It is annoying and kind of a waste of my time. If this class was curved, I would act stupid in front of my partner and the do the lab at home. If I did the lab at home it probably wouldn't even take an hour, but instead I am forced to work with someone and it takes 2 hours. I do not think that pair programming has led to a positive impact in my education. It has only led to the resentment of my lab partner."

5 DISCUSSION

Students' perceptions of pair programming seem overwhelmingly positive. Many men and women said that pair programming improved their learning experiences overall and that it led to a more efficient and productive coding process while supporting the development of career skills. Students discussed negatives rarely. Some gender patterns do appear in the themes reported that point to relationships between the benefits and disadvantages of pair programming in relation to the challenges women face in CS1 classrooms. It is critical to note that women are not a homogeneous group with one set of experiences (and neither are men). Each student brings a unique set of past experiences, attitudes, skills, and personality traits to the classroom. This analysis does not purport to generalize about all women in CS or all women in this sample, but rather to describe and discuss the diverse perspectives in our data in a systematic manner.

Partners as Social Supports. Many men, but few women, cited reinforcing their knowledge through explaining to their partner as a benefit of pair programming (Table 2). This aligns with social cognitive learning research, which has shown that explanation and discussion helps to advance a student's understanding. When students feel positive about explaining to less experienced partners, this is likely to come across in their interactions. Some women's reflections link this dynamic to a positive atmosphere, mentioning that working with a more experienced partner who "helps me understand" helped to build confidence and that a knowledgeable partner is "less scary" to ask for help compared to a TA. Many women also said that sharing the experience of difficulty and confusion with a partner helped them feel more comfortable, less like "the only one who does not understand something." Prior research has found that women tend to have lower self-confidence than men when it comes to computer science ability, even when controlling for quantitative ability [1]. Providing a positive atmosphere where women feel less discouraged, frustrated, and stressed is a particularly important benefit for this reason.

Career Skills and Perceptions. While it is promising that some students saw pair programming as a means to develop career skills, many students did not make this connection or choose to mention it. Misconceptions about computer science abound from popular culture, including that programming is a solo activity and that computer scientists lack interpersonal skills [6]. Pair programming is an authentic way to battle this misconception. Prior research has shown that women value careers that are people-oriented and view computer science as lacking in that area [3]. In order to retain women in the field, we need to ensure they are aware of the collaboration that occurs in industry and highlight the opportunities within computer science that are more people-oriented.

Burden and Resentment. Among the few negatives that students reported, women and men largely focused on similar threads: they felt the process was inefficient, slow, and less hands-on (Table 4). However, there was one gender difference. Two women, but no men, reported the feeling of burdening their partner, while three men, but no women, said their partner was leaving them to carry the weight of the work. The dynamics of the freeloading phenomenon are worthy of further study: it may be that some students are in fact just taking advantage of their partner; others could be less-experienced students who might stay quiet due to feelings of low confidence or intimidation, or students who feel socially alienated from their partners. The feelings of burdening a partner may be entirely self-reflective or could be responses to a partner's behavior; this phenomenon could also be investigated as a window into students' affective experiences.

Implications for Instructors. It is critical that students, especially those that are underrepresented, feel welcome in the community. Given that women reported social and affective benefits to pair programming more often than men, instructional methods should encourage social interactions which can aid in reducing negative emotions. Several women valued having a peer to ask questions to, saying it was not as intimidating as asking questions to a superior. Lowering the barrier to receiving help has the potential to help all students. Instructors should emphasize that asking questions is part of the learning process to make students feel more comfortable when reaching out for help. Providing multiple ways to receive help, such as via an anonymous discussion board or through an assigned study group, could lessen the intimidation some women feel. Additionally, we need to ensure that students are able to see the value in working with others, regardless of their partner's aptitude or prior experience. Instructors should preface pair programming with a discussion on how both parties can benefit from pair programming, citing how students can reinforce their knowledge and improve their communication skills.

Table 4: Six most common negative codes mentioned by women versus six most common negative codes mentioned by men.

Rank	Women's code	Rank	Men's code
1	Inefficient/Slower	1	Inefficient/Slower
2	Learn better alone	2	Learn better alone
2	Incompatible partner	2	Work better/faster alone
2	I am burdening partner	3	Unequal workload (partner freeloading)
2	Less hands-on	3	Less hands-on
2	Frustrating	3	Less convenient

Limitations. As is the case in all qualitative research, the individual lenses through which researchers view the data influence the themes that emerge. However, the methodology of collaborative coding and discussion to consensus is designed to produce a reliable labeling of the data that can be used for further analysis. Additionally, this class was made up of 20 in-class lab sections run by 13 teaching assistants. The variability in instructors and the number of lab sections is a potential source for variability in the enforcement of the pair programming paradigm.

6 CONCLUSION AND FUTURE WORK

Our goal for this study was to understand how men and women perceive pair programming in an introductory computer science course. We analyzed 104 student responses about pair programming and identified overarching themes as well as gender differences within those themes. Both men and women had predominantly positive sentiments about their experience with their partners and with pair programming. They cited benefits of improved learning experience, gaining career skills, and networking at nearly equal rates. Benefits to social and affective well-being were reported more by women, while benefits to the overall process of completing the lab assignments were reported more by men. Only women felt they were burdening their partner, and only men felt their partner was burdening them or was freeloading.

Having assigned partners gives women the opportunity to ask questions in a safe space. This is important not just for learning, but for mitigating feelings of isolation, which are common for women in CS courses. Future research should investigate other ways to encourage students to use each other as resources. These experiences may positively impact the culture in the classroom and increase students' sense of belonging.

The disadvantages of pair programming reported by students stemmed largely from incompatible partners and preference by some to work alone. Future studies should investigate voluntary pair programming and methods for matching students. In their responses, many students describe times when they might prefer to work alone instead of with a partner. By understanding these situational preferences, we can scaffold pair programming experiences to fit within these preferences.

Future work using additional data sources, such as video recordings, students' discussion board or chat conversations, and project and assignment artifacts are also needed to better understand and support women's interest, persistence, and success in computer science. Prior research has analyzed dialogue between students engaging in pair programming, and future studies should expand on this work to look specifically at how student pairs collaborate with respect to their gender composition.

7 ACKNOWLEDGEMENTS

The authors are grateful to the members of the LearnDialogue Group for their help and encouragement. This material is based upon work supported by the National Science Foundation under grant CNS-1622438. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Sylvia Beyer, Kristina Rynes, Julie Perrault, Kelly Hay, and Susan Haller. 2003. Gender differences in computer science students. *SIGCSE Bulletin* 35, 1 (2003), 49–53. https://doi.org/10.1145/792548.611930

[2] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77–101. https://doi.org/10.1191/1478088706qp063oa

[3] Lori Carter. 2006. Why students with an apparent aptitude for computer science don't choose to major in computer science. *ACM SIGCSE Bulletin* 38, 1 (2006), 27. https://doi.org/10.1145/1124706.1121352

[4] Mehmet Celepkolu and Kristy Elizabeth Boyer. 2018. Thematic Analysis of Students' Reflections on Pair Programming in CS1. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education - SIGCSE '18* (2018), 771–776. https://doi.org/10.1145/3159450.3159516

[5] Sapna Cheryan, Victoria C. Plaut, Paul G. Davies, and Claude M. Steele. 2009. Ambient Belonging: How Stereotypical Cues Impact Gender Participation in Computer Science. *Journal of Personality and Social Psychology* 97, 6 (2009), 1045–1060. https://doi.org/10.1037/a0016239

[6] Sapna Cheryan, Victoria C. Plaut, Caitlin Handron, and Lauren Hudson. 2013. The Stereotypical Computer Scientist: Gendered Media Representations as a Barrier to Inclusion for Women. *Sex Roles* 69, 1-2 (2013), 58–71. https://doi.org/10.1007/s11199-013-0296-x

[7] Sapna Cheryan, Sianna A. Ziegler, Amanda Montoya, and Lily Jiang. 2017. Why are some STEM fields more gender balanced than others? *Psychological Bulletin* 143, 1 (2017), 1–35. https://doi.org/10.1037/bul0000052

[8] Alistair Cockburn and Laurie Williams. 2001. The Costs and Benefits of Pair Programming. In *Extreme Programming Examined*, Giancarlo Succi and Michele Marchesi (Eds.). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 223–243. http://dl.acm.org/citation.cfm?id=377517.377531

[9] Wendy DuBow, Joanna Weidler-Lewis, and Alexis Kaminsky. 2016. Multiple Factors Converge to Influence Women's Persistence in Computing: A Qualitative Analysis of Persisters and Nonpersisters. In *2016 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, Adrienne Decker, Kurt Eiselt, Jamie Payton, Tiffany Barnes, and George K. Thiruvathukal (Eds.). Atlanta, GA, USA, 1–7. https://doi.org/10.1109/RESPECT.2016.7836161

[10] Susan Horwitz, Barbara Ryder, Monica Sweat, Susan H. Rodger, Maureen Biggers, David Binkley, C. Kolin Frantz, Dawn Gundermann, Susanne Hambrusch, Steven Huss-Lederman, and Ethan Munson. 2009. Using peer-led team learning to increase participation and success of under-represented groups in introductory computer science. *ACM SIGCSE Bulletin* 41, 1 (2009), 163. https://doi.org/10.1145/1539024.1508925

[11] Jennifer Jenson, Suzanne De Castell, and Mary Bryson. 2003. "Girl talk": Gender, equity, and identity discourses in a school-based computer culture. *Women's Studies International Forum* 26, 6 (2003), 561–573. https://doi.org/10.1016/j.wsif.2003.09.010

[12] Julie Krause, Irene Polycarpou, and Keith Hellman. 2012. Exploring formal learning groups and their impact on recruitment of women in undergraduate CS. *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education (SIGCSE '12)* (2012), 179. https://doi.org/10.1145/2157136.2157192

[13] Charlie McDowell, Linda Werner, Heather Bullock, and Julian Fernald. 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)*. ACM, New York, NY, USA, 38–42. https://doi.org/10.1145/563340.563353

[14] Charlie McDowell, Linda Werner, Heather E Bullock, and Julian Fernald. 2006. Pair programming improves student retention, confidence and program quality. *Commun. ACM* 49, 8 (2006), 90–95.

[15] Nachiappan Nagappan, Laurie Williams, Miriam Ferzli, Eric Wiebe, Carol Miller, Suzanne Balik, and Kai Yang. 2003. Improving the CS1 Experience with Pair Programming. In *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*. 359–362.

[16] Clem O'Donnell, Jim Buckley, Abdulhussain Mahdi, John Nelson, and Michael English. 2015. Evaluating Pair-Programming for Non-Computer Science Major Students. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. 569–574.

[17] Leo Porter and Beth Simon. 2013. Retaining Nearly One-Third more Majors with a Trio of Instructional Best Practices in CS1. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*. Denver, CO, 165–170.

[18] Alex D. Radermacher and Gursimran S. Walia. 2011. Investigating the Effective Implementation of Pair Programming: An Empirical Investigation. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education (SIGCSE '11)*. ACM, New York, NY, USA, 655–660. https://doi.org/10.1145/1953163.1953346

[19] Rachael D. Robnett and Campbell Leaper. 2012. Friendship Groups, Personal Motivation, and Gender in Relation to High School Students' STEM Career Interest. *Journal of Research on Adolescence* 23, 4 (2012), 652–664. https://doi.org/10.1111/jora.12013

[20] Mary Beth Rosson, John M. Carroll, and Hansa Sinha. 2011. Orientation of Undergraduates Toward Careers in the Computer and Information Sciences. *ACM Transactions on Computing Education* 11, 3 (2011), 1–14. https://doi.org/10.1145/2037276.2037278

[21] Heather Thiry and Sarah Hug. 2014. "We Should All Help Each Other": Latina Undergraduates' Practices And Identities In The Figured World Of Computing. *International Conference of the Learning Sciences* (2014), 214–221.

[22] United States Department of Labor: Bureau of Labor Statistics. 2018. Employment Projections: Employment by detailed occupation, 2016 and projected 2026.

[23] United States Department of Labor: Bureau of Labor Statistics. 2018. Labor Force Statistics from the Current Population Survey: Employed persons by detailed occupation, sex, race, and Hispanic or Latino ethnicity.

[24] Stuart Zweben and Betsy Bizot. 2017. 2016 Taulbee Survey: Generation CS Continues to Produce Record Undergrad Enrollment; Graduate Degree Production Rises at both Master's and Doctoral Levels. *Computing Research News* 29, 5 (2017), 3–51.