# Long Term Effects of Partner Programming in an Introductory Computer Science Sequence

Andrew Giugliano and Andrew DeOrio
agiuglia@umich.edu, awdeorio@umich.edu
Department of Electrical Engineering and Computer Science
University of Michigan

## Abstract

Computer scientists often work in teams on complex software projects, and their education often includes group work or pair programming. In the literature, group work and pair programming have been shown to improve student learning outcomes. For example, students programming in pairs produce better programs and have similar exam scores. In this paper, we explore the impact of student partnerships in introductory programming projects on future student performance in subsequent courses.

We examine the data from 2,234 students enrolled in introductory computer science courses at a large, highly ranked, public research institution, including participation in group work and performance in two different courses. The first is a second-semester "CS2" programming and introductory data structures course where some students completed projects in partnerships while others worked alone. Students then advanced to a "CS3" course where all students completed projects alone. Both courses are part of the core computer science curriculum for all computer science majors and minors, and are considered gateway courses to specialized upper level courses.

In our results, we observed an association between student partnerships and increased project performance during the same semester, confirming the observations of prior work. When examining student performance in the following course (CS3), our largest statistically significant result was for students in the lowest GPA quartile: those who had partnered in CS2 were associated with improved project scores in CS3, where they all worked alone.

## 1 Introduction and Related Work

Pair programming is a software development technique where two programmers work together at one workstation on the same problem. In both education and industry, the goal of this practice is to increase the correctness of programs the pairs produce, as well as programmers sharing knowledge.

Past work has provided substantial evidence that Pair Programming leads to higher student performance in introductory computer science courses. In 2002, McDowell *et al.* [1] showed that students completing projects in pairs led to higher project performance. This applied to both low and high achieving groups of students. They also observed similar exam performance among students who completed projects in pairs and those who completed projects independently.

Additionally, Nagappan *et al.* (2003)[2] and McDowell *et al.* (2003, 2006)[3][4] described pair programming leading to higher student retention rates in first year computing courses.

In addition to improving retention, other research into pair programming has examined its impact on different demographic groups. In 2008, Braught *et al.* [5] observed that pair programming led to higher programming skills for students with lower SAT scores. More recently in 2013, Wood *et al.* [6] found that pair programming can lead to higher performance in introductory programming classes, especially for students who begin with low confidence levels. In 2004, Werner *et al.* [7] observed that the practice of pair programming in introductory programming courses increases retention of women students in computer science.

In addition to examining pair programming in the context of introductory computer science courses, the community has also explored pair programming in second semester computing courses ("CS2"). In 2005, and later in 2006, Mendes *et al.* [8][9] focused in on the effects of pair programming in CS2. They found that partnerships were associated with better individual work, as well as better exam scores. Additionally, they found that pair programming was favored among students as compared to working independently. In this study, partnerships were used in voluntary laboratory sessions where the work did not contribute to a student's overall grade.

Not only have researchers explored pair programming in academia, but they have also extensively examined pair programming and its effects in industry. Williams *et al.* (2000)[10] and Cockburn *et al.* (2001)[11] extended this research to industry software developers. The study showed that pair programming led developers to finish higher-quality programs in less time. Additionally, those who participated in partnerships were more satisfied with the coding experience, a finding that is consistent with previous studies in the college course setting.

### 1.1 Pair Programming Concerns

Notwithstanding the observed advantages of pair programming, there are also concerns with the practice of providing students the opportunity to partner on course projects. Primarily, academics worry that students will not pair program at all; and instead elect to divide the work with a "divide and conquer" strategy. The risk is that some students may miss the focus of the project by only completing different subsections. If they elect to do the subsections that are the easiest for them, it may inhibit students from learning subject matter with which they most need practice. Additionally, if the work is not partitioned evenly, one partner may complete most of the project while the other gains little or no experience.

Another concern about partnerships in education is that students may become dependent on partnerships. This could lead to problems for students in future courses, when they may be required to work independently. This concern is magnified when considering the core skills being

developed in early courses.

Our goal in this paper is to evaluate these concerns by analyzing the performance of students who partner in an early course and then work alone in a future course. In the next section, we discuss the composition of our data as well as the methods we use to produce our results. In Section 3 we show the results of our findings. Finally, in Section 4, we discuss the implications of our results, followed by conclusions.

## 2 Methods

To evaluate the effects of student partnerships in both CS2 and CS3 courses, we take a statistical approach, analyzing performance and demographic data. Our data comes from the grade books of both courses over several semesters. The grade book data contains aggregate project and exam scores from CS2 and CS3. Additionally, these data contain a record of whether each student participated in a partnership or chose to work alone on the CS2 programming projects. In CS3, all students worked alone. In addition to grade data, we obtained gender and cumulative grade point average data from university records.

Using these data, we evaluate the effect of partnerships for both the group as a whole, as well as the effect on different subgroups of students. Specifically, our research questions are:

- Are student partnerships during a past semester associated with changes in student performance while working alone in a future semester?
- Do observations about student partnerships vary with different demographic groups?

In the remainder of this section, we describe the content of the CS2 and CS3 courses.

### 2.1 Description of CS2

In our study, CS2 follows a CS1 introductory programming course. The primary audience of CS2 is underclassmen who are planning to pursue a major in Computer Science and Engineering. Other popular majors of CS2 students include Business Administration, Economics and Statistics, as well as others working towards a Computer Science Minor. The only enforced prerequisite for the class is an introductory programming course taught in C++.

CS2 focuses on core computer science concepts and covers four major topics. First, functional abstraction, including specification, recursion, iteration, and functional generalization. Second, data abstraction, including types, type hierarchies, abstract data types, abstraction and polymorphism. Third, dynamic resource management, including creation, deletion, and interaction with containers. Finally, elements of C++, including arrays, structs, classes, objects, function and operator overloading, strings, pointers, templates, linked lists, stacks and queues, iterators, exceptions, and functors.

The class is partitioned into several lecture sections with 3 hours a week of total lecture time, as well as an additional 1 hour lab section. During lab sections, students complete short assignments to reinforce material learned in lecture. Final grades in CS2 follow the rubric in Table 1.

| Assignment | % of Final Grade |
|---|---|
| Coding Projects | 40% |
| Lab Exercises | 5% |
| Midterm Exam | 25% |
| Final Exam | 30% |

**Table 1: Grading rubric for CS2**.

Assignments include 5 programming projects that focus on specific topics from the course. Projects typically consist of students writing code to build a program ranging in size from several hundred lines of code to over 1,000. A few examples include a game driver for a Euchre card game, and a post-fix notation calculator. Students use the C or C++ programming languages to complete each project. Students participated in optional partnerships during projects 2 - 5, while project 1 was completed alone.

Following are the topics of each project:

- Project 1, Basic statistics: C++ readiness
- Project 2, Lists and Trees: Recursion and Functional Generalization
- Project 3, Euchre Card Game: C-Style programming (structs, arrays and pointers)
- Project 4, Blackjack game and AI players: Object oriented programming C++ style (classes, inheritance, polymorphism)
- Project 5, Post-fix Notation Calculator: Dynamic memory management, linked lists, stacks

#### 2.1.1 Partnerships

Students worked in optional partnerships on projects 2, 3, 4 and 5. Students could not change partners in the middle of a project, but had the freedom to change partners in between. Partnerships submitted one project for the pair, with both partners receiving the same grade. Partnership guidelines encouraged pair programming, and strongly discouraged "divide-and-conquer" partnership strategies.

Within the span of the semesters of CS2 from which we use data, the curriculum, course policies and grading rubric remained consistent.

### 2.2 Description of CS3

CS3 is an advanced data structures and algorithms course whose main audience is second or third year student. Its prerequisites are CS2 and a discrete math course. CS3 covers three major topics. First, algorithm analysis, including O-notation and program efficiency. Second, data structures, including lists, stacks, queues, hash tables, trees and graphs. Finally, algorithms, including searching, sorting, recursive algorithms and graph algorithms.

Similar to CS2, the class is partitioned into several lecture sections with 3 hours per week of instruction, along with a 1 hour per week discussion section. During discussion sections, student

instructional assistants reinforce concepts learned in lecture and rehearse practice problems. Final grades in CS3 follow the rubric in Table 2.

| Assignment | % of Final Grade |
|---|---|
| Coding Projects | 40% |
| Homework | 10% |
| Midterm Exam | 25% |
| Final Exam | 25% |

**Table 2: Grading rubric for CS3.**

Four coding projects focus on speed and memory efficiency, in addition to correctness. Several examples include a stock market simulator, and a map routing program. Following are descriptions of each project, all written in C/C++:

- Project 1: Breadth first search (BFS) and depth first search (DFS)
- Project 2: Priority queues and recursive algorithms
- Project 3: Hash tables and balanced trees
- Project 4: Graph algorithms

#### 2.2.1 Partnerships

All students complete all projects independently in CS3. The course syllabus allows students to discuss general course concepts as they apply to the projects, but not specific solutions or code.

The curriculum, course policies and grading rubric for CS3 remained consistent during the semesters we collected our data set.

### 2.3 Design of the Study

Our data comes from a span of four semesters of CS2, and two semesters of CS3. An academic year comprises two semesters. We mapped records of individuals from CS2 to CS3, observing project and exam performance, as well as their decision to partner in CS2.

For our study, we used several methods to summarize student performance in CS2 and CS3. For projects in CS2, we focused on each student's average grade across all five projects. We used the same approach for projects in CS3. Similarly, exam scores for both CS2 and CS3 are aggregated according to the course rubrics.

### 2.4 Statistical Analysis

We evaluated results by comparing sample means from different groups, and verifying statistical significance using the student's *t*-test. The composition of this data is detailed in Section 2.7. When analyzing gender groups, we split our data into two partitions, men and women. When

analyzing GPA groups, we split the data into 4 groups, being the 1st, 2nd, 3rd and 4th quartiles based upon GPA.

Next, we summarize our data from CS2 (Section 2.5) and CS3 (Section 2.6).

### 2.5 Summary of CS2 Data

We first summarize student performance in CS2, where students participated in optional partnerships. In total, our data set contains 2,234 records for students in CS2, spanning the four semesters of academic years 2013-2014, and 2014-2015.

Before analyzing our data, we first cleaned it by removing several categories of students from the raw data. We removed students who were caught for cheating on projects or exams. Records from students who had withdrawn from the class, which would produce incomplete data were removed. Finally, we removed students who audited the course, which again would produce incomplete data. After cleaning the data set, 1,919 records remained.

Among students who withdrew from CS2 during the semester, 19 worked in partnerships, while 81 worked alone. We note that despite more students partnering, fewer withdrew. We did not have information in our data set on CS3 withdraw rates.

Table 3 describes the number of students who chose to work in partnerships during each of the four semesters. During semesters 1 and 2, students did not have the opportunity to participate in a partnership. During semesters 3 and 4, partnerships were optional.

| Semester of CS2 | Partner Category | Number of students |
|---|---|---|
| 1 | Alone | 357 |
| 2 | Alone | 530 |
| 3 | Alone | 185 |
| | Partner | 310 |
| 4 | Alone | 215 |
| | Partner | 322 |

**Table 3: Number of students who worked in partnerships in CS2**, across the four semesters of our study.

### 2.6 Summary of CS3 Data

Next, we summarize our data from CS3, where all students worked alone. In total, we have 665 records spanning the the second semester of academic year 2013-2014 and the first semester of academic year 2014-2015.

We created a one-to-one mapping of a student's performance in CS2 to their performance in CS3. Therefore, we reduced our data set to only include records with entries for both CS2 and CS3. This subset of our data contains 665 records. Table 4 describes the number of students in each semester of CS2 for which we have CS3 performance data.

As mentioned above, during semester 1 and 2 students did not have the opportunity to be participate in CS2 partnerships.

| Students in CS3 | | |
|---|---|---|
| **Past Semester of CS2** | **Partner Category in CS2** | **Number of students in CS3** |
| 1 | Alone | 32 |
| 2 | Alone | 126 |
| 3 | Alone | 92 |
| | Partner | 152 |
| 4 | Alone | 103 |
| | Partner | 160 |

**Table 4: CS3 data, divided by students who decided to partner in CS2**, spanning students' past CS2 semesters 1-4. Semesters 1 and 2 had a no partnership policy for projects, while 3 and 4 had the option to partner on projects.

### 2.7 Summary of Gender and GPA Data

When evaluating the effects of partnerships in CS2 on performance in CS2 and CS3, we examined demographic subsets of our data set to observe the effect partnerships may have on different groups. First, we look at the effects of partnerships within different gender groups. Within our CS2 data set (Section 2.5), we summarize the composition of each gender group in Table 5. We noticed that women were somewhat more likely to partner on projects (64%) compared men (60%).

| | **Number of students** | | **Partner Category** | |
|---|---|---|---|---|
| **Gender** | **CS2** | **CS3** | **Partner** | **Alone** |
| Men | 1510 | 399 | 473 (60%) | 305 (40%) |
| Women | 408 | 107 | 158 (64%) | 88 (36%) |

**Table 5: Composition of gender data**, as well as a count of how many of each gender group chose to partner or not partner on projects in CS2. Note: we only consider students who had the choice to partner (semesters 3 and 4 of CS2).

We also stratified our data by GPA, partitioning our data set into quartiles. Table 6 shows the GPA boundaries and number of students in each group described in each CS2 semester (Section 2.5). We noticed that the first, third and fourth quartile had similar ratios of students who partnered to those who worked alone, while students from the second quartile were somewhat more likely to partner.

When we examine GPA in the CS3 subset of the data (Section 2.6), we see the composition shown in Table 7. In this table, we only included students in CS2 semesters 3 and 4, where students were given the choice to partner. Here, we noticed that students from the first quartile were far more likely to partner, students from the second and third quartiles were similar in likeliness to partner, and students from the fourth quartile were the least likely to partner.

| Quartile | GPA range | Number of students | Partner | Alone |
|---|---|---|---|---|
| 1st | [ 0, 3.013 ] | 481 | 146 (58%) | 104 (42%) |
| 2nd | ( 3.013, 3.378 ] | 480 | 179 (68%) | 86 (32%) |
| 3rd | ( 3.378, 3.716 ] | 479 | 154 (61%) | 98 (39%) |
| 4th | ( 3.716, 4.166 ] | 479 | 153 (59%) | 105 (41%) |

**Table 6: Composition of CS2 GPA groups** used in our results to observe patterns in performance among specific groups based on GPA. Several students from the business school had GPAs above 4.0, which is the result of "A+" being counted as 4.4. Note: partnership numbers include only students who had the choice to partner (semesters 3 and 4 of CS2).

| Quartile | GPA range | Number of students | Partner | Alone |
|---|---|---|---|---|
| 1st | [ 2.267, 3.145 ] | 127 | 88 (69%) | 39 (31%) |
| 2nd | (3.145, 3.423 ] | 127 | 75 (59%) | 52 (41%) |
| 3rd | ( 3.423, 3.763 ] | 126 | 78 (61%) | 48 (39%) |
| 4th | ( 3.763, 4.166 ] | 127 | 71 (56%) | 56 (44%) |

**Table 7: Composition of CS3 GPA groups** used in our results.

## 3 Results

We examine the effect of student partnerships in CS2 on performance in both CS2 and a later CS3 course. We also investigate its effect on different subgroups based on gender and grade point average. In the following sections, we will examine the effect of CS2 partnerships on CS2 project and exam performance. Then, we perform a similar analysis with CS3 project and exam performance. Finally, we divide our data set into groups by gender and GPA, examining the size of the effect on these subgroups.

### 3.1 Effect of CS2 partnerships on CS2 performance

In our first analysis, we examine the effect of project partnerships in CS2 on student project and exam performance during the same semester that students participated in the partnership. Figure 1 shows that scores of students who chose to partner had a higher median and a tighter distribution compared to students who worked alone on projects. Additionally, we can see in Table 8 that students who chose to partner scored an average of 3.3% higher on projects, and with $p < 0.0001$, we can conclude that this result is statistically significant.

When examining the effect of partnerships in CS2 on performance on CS2 exams, we observed the opposite effect, compared to project scores. We observed that exam scores were lower when students choose to partner on projects in CS2. As we see in Table 8, students who chose to partner scored an average of 2.8% lower on exams.

| Evaluation | Partnered Mean | Partnered N | Alone Mean | Alone N | Difference in mean | $p$ Value |
|---|---|---|---|---|---|---|
| Projects | 83.3% | 632 | 80.0% | 393 | 3.3% | 0.0001 |
| Exams | 71.8% | 632 | 74.6% | 393 | -2.8% | 0.001 |

**Table 8: CS2 project and exam performance** for students who worked alone, and those who partnered.

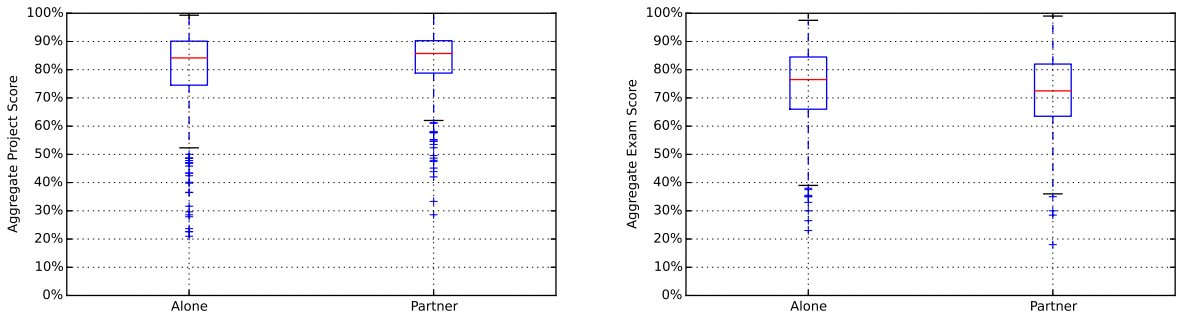


**Figure 1: Box plot of CS2 project and CS2 exam scores by partnership status.** On the left, is a box plot showing the project scores and on the right, a box plot showing exam scores.

### 3.2 Effect of CS2 partnerships on CS3 performance

In our next analysis, we examine the effect of CS2 project partnerships on CS3 student performance. In Table 9, we see the difference in means and $p$ values. We did not observe any statistically significant results, because $p > 0.05$.

We also look at the effect of CS2 partnerships on student performance on CS3 exams. The difference in mean and $p$ value are in Table 9. Again, these results are not statistically significant.

| Evaluation | Partnered Mean | Partnered N | Alone Mean | Alone N | Difference in mean | $p$ Value |
|---|---|---|---|---|---|---|
| Projects | 77.0% | 312 | 76.7% | 195 | 0.3% | 0.867 |
| Exams | 62.7% | 312 | 64.6% | 195 | -1.9% | 0.153 |

**Table 9: CS3 project and exam performance** for students who worked alone, and those who partnered in their preceding CS2 course.

### 3.3 Effect of CS2 partnerships on CS2 performance among gender groups

In Figure 2, we analyze the effect of partnerships on projects in CS2 by gender group. In Table 10, we see that we can conclude with statistical significance that partnering is associated with better project scores in CS2 for both gender groups. Furthermore, women stand to benefit more

from partnerships as compared to men, scoring 4.9% higher on average compared to 2.7%. Furthermore, we observed that men scored 3.2% lower on exams in CS2 when participating in a partnership.

| | | Partnered | | Alone | | | |
|---|---|---|---|---|---|---|---|
| **Evaluation** | **Group** | **Mean** | **N** | **Mean** | **N** | **Difference in mean** | **$p$ Value** |
| Projects | men | 83.0% | 473 | 80.3% | 305 | 2.7% | 0.005 |
| | women | 84.1% | 158 | 79.1% | 88 | 5.0% | 0.007 |
| Exams | men | 72.0% | 473 | 75.2% | 305 | -3.2% | 0.001 |
| | women | 70.9% | 158 | 72.5% | 88 | -1.6% | 0.388 |

**Table 10: CS2 project and exam performance by gender** for students who worked alone, and those who partnered.

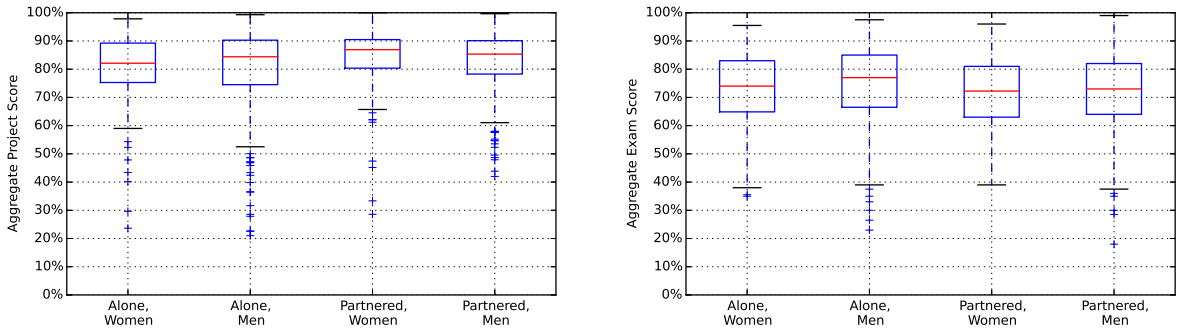


**Figure 2: Box plot of CS2 project and exam scores by gender and choice in partnership.**

### 3.4 Effect of CS2 partnerships on CS3 performance among gender groups

Next, we analyze the effect of partnerships on performance in CS3 among gender groups. In both Figure 3 and Table 11, we see that men who partnered in CS2 scored better on projects in CS3 by 4.6% on average. We did not observe statistically significant results for women.

When analyzing the relationship between partnerships in CS2 and exams in CS3 (Table 11), we did not find any statistically significant associations describing whether partnering in CS2 has an effect on exam scores in CS3 for either gender group.

| | | Partnered | | Alone | | | |
|---|---|---|---|---|---|---|---|
| **Evaluation** | **Group** | **Mean** | **N** | **Mean** | **N** | **Difference in mean** | **$p$ Value** |
| Projects | men | 77.2% | 244 | 72.6% | 155 | 4.6% | 0.023 |
| | women | 76.7% | 67 | 69.3% | 40 | 7.3% | 0.111 |
| Exams | men | 62.9% | 244 | 64.6% | 155 | -1.7% | 0.110 |
| | women | 61.9% | 67 | 60.9% | 40 | 1.0% | 0.712 |

**Table 11: CS3 project and exam performance by gender**

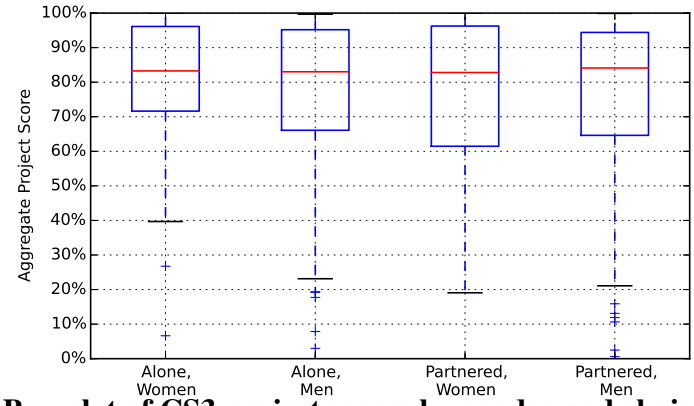


**Figure 3: Box plot of CS3 project scores by gender and choice in partnership.**

### 3.5 Effect of CS2 partnerships on CS2 performance among GPA groups

In Figure 4, we analyze the relationship between CS2 projects, partnerships and GPA. We can see, project scores raise dramatically with each successive group, and groups who partner tend to better than their counterparts who did not partner We see in Table 12 that for the 3 lower quartiles of students, project scores tended to be higher for students who chose to partner. Additionally, the lower the GPA is, the higher the effect of partnerships on increasing performance.

In Figure 4, we additionally analyze the relationship between CS2 exams, partnerships and GPA. Here, exam scores raise dramatically with each successive group, and groups who did not partner tend to better than their counterparts who did partner. In Table 12 we see that the 3 higher quartiles of students, exam scores tended to be lower for students who chose to partner.

| | | Partnered | | Alone | | | |
|---|---|---|---|---|---|---|---|
| **Evaluation** | **Quartile** | **Mean** | **N** | **Mean** | **N** | **Difference in mean** | $p$ **Value** |
| Projects | 1st | 76.6% | 146 | 67.8% | 104 | 8.8% | 0.000021 |
| | 2nd | 81.4% | 179 | 77.7% | 86 | 3.7% | 0.033 |
| | 3rd | 85.7% | 154 | 83.6% | 98 | 2.1% | 0.022 |
| | 4th | 89.5% | 153 | 90.6% | 105 | -1.2% | 0.095 |
| Exams | 1st | 61.6% | 146 | 62.9% | 104 | -1.3% | 0.434 |
| | 2nd | 66.9% | 179 | 70.2% | 86 | -3.3% | 0.031 |
| | 3rd | 74.4% | 154 | 78.2% | 98 | -3.8% | 0.001 |
| | 4th | 84.5% | 153 | 86.4% | 105 | -1.9% | 0.037 |

**Table 12: CS2 project and exam performance by GPA quartile** for students who worked alone, and those who partnered.

### 3.6 Effect of CS2 partnerships on CS3 performance among GPA groups

In Table 13, we analyze the relationship between partnerships, GPA, CS3 projects and CS3 exams. We can see that for the first (lowest) quartile, students who partner in CS2 do better on

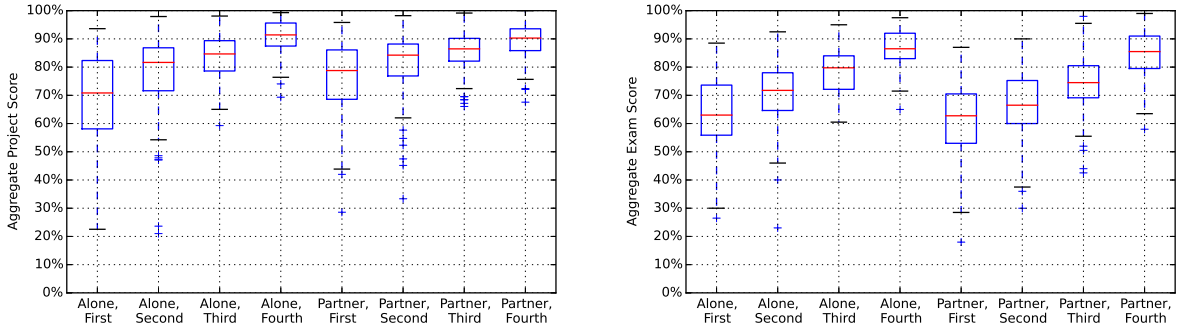


**Figure 4: Box plot of CS2 project and exam scores by GPA group and choice in partnership.** On the left is a box plot showing CS2 project scores and on the right, showing CS2 exam scores.

projects in CS3 by an average of 9.2%. Additionally, we can see that students in the fourth quartile do slightly worse on exams in CS3 when partnering in CS2, by an average of about 3.8%. This is also shown in Figure 5.

| | | Partnered | | Alone | | | |
|---|---|---|---|---|---|---|---|
| **Evaluation** | **Quartile** | **Mean** | **N** | **Mean** | **N** | **Difference in mean** | $p$ **Value** |
| Projects | 1st | 60.4% | 88 | 51.2% | 39 | 9.2% | 0.032 |
| | 2nd | 71.0% | 75 | 66.2% | 52 | 4.8% | 0.149 |
| | 3rd | 81.7% | 78 | 77.7% | 48 | 4.0% | 0.168 |
| | 4th | 90.8% | 71 | 92.1% | 56 | -1.3% | 0.469 |
| Exams | 1st | 55.2% | 88 | 55.6% | 39 | 0.04% | 0.846 |
| | 2nd | 57.4% | 75 | 58.2% | 52 | -0.8% | 0.669 |
| | 3rd | 64.4% | 78 | 66.6% | 48 | -2.0% | 0.223 |
| | 4th | 72.0% | 71 | 75.8% | 56 | -3.8% | 0.008 |

**Table 13: CS3 project and exam performance by GPA quartile** for students who worked alone, and those who partnered in their prior CS2 course.

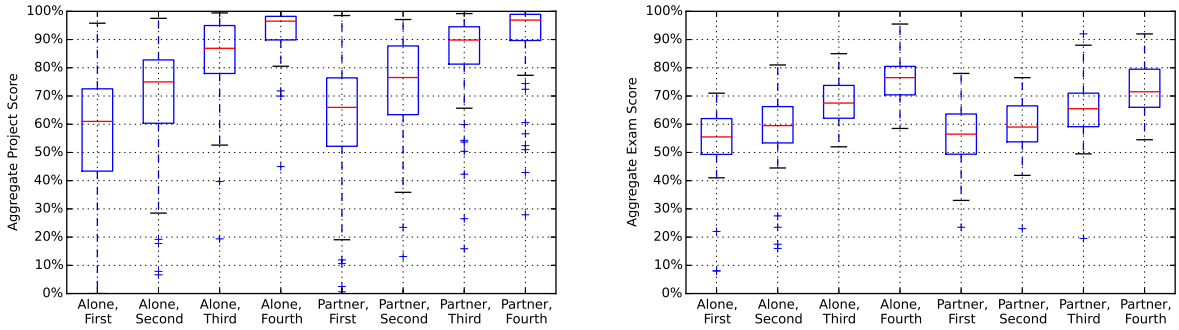


**Figure 5: Box plot of CS3 project and exam scores by GPA group and choice in partnership.** On the left is a box plot showing CS3 project scores and on the right, showing CS3 exam scores.

## 4 Discussion

We observed many statistically significant findings on the effect of partnerships during CS2 on student performance during the same semester. Our results confirmed an overall positive effect of partnerships observed by other researchers. When analyzing the impact of partnerships on student performance in a subsequent CS3 course, our strongest positive observations were for students in the lowest GPA quartile.

In this section, we discuss our key findings, including the effect of partnerships during the same semester (CS2), and during the following semester (CS3). We also examine several subgroups of the data set by gender and GPA. Finally, we discuss several limitations.

### 4.1 Effects of CS2 partnerships on CS2

First, we discuss the effect of project partnerships in CS2 on student performance during the same semester. In the previous section (Table 8 in particular), we observed that the effect of partnerships produced different results on projects and exams. Overall, those students who partnered tended to score better on projects, with statistical significance. Our observations on student project performance are consistent with most literature on pair programming; as mentioned in Section 1, where other studies have found that pair programming leads to fewer bugs in coding work, and higher overall scores. Furthermore, we saw that students who choose to work alone were associated with higher withdraw rates than those who worked with a partner.

On the other hand, we also observed that students working alone were often associated with higher exams scores. Several factors could influence this observation. For example, the instructors did not control team selection. There could be a relationship between students who chose to work alone and their exam-taking ability.

Additionally, the instructors did not have control over the student partnership dynamics, and the extent to which they employed pair programming, or whether students divided project work. This could have caused some students to become less familiar with some exam topics. We discuss these and other limitations in Section 4.4.

### 4.2 Effects of CS2 partnerships on CS3

When analyzing associations between partnerships in CS2 and performance in CS3 in the general population, we did not observe any statistically significant relationships. While analyzing subgroups of these students, we did observe statistically significant associations, which we discuss in the next Section.

We note that our lack of statistically significant associations included a lack of statistically significant negative impacts. This could point to the conclusion that partnerships in CS2 were not associated with long-term harm to student performance.

Another possible reason for the lack of statistically significant associations between project partnerships in CS2 and CS3 performance among the overall population is that students are forced to work alone on projects in CS3, and therefore the effect of whether or not they partnered in CS2 may be diminished.

### 4.3 Effects of partnerships on different demographic groups

Throughout our analysis, we observed several statistically significant associations between partnerships and the performance of different subgroups of students, divided by gender and GPA.

When analyzing the effect of partnerships on CS2 performance for men and women, we see that women scored an average of 5% higher on projects in CS2; nearly double the benefit observed for men. Conversely, the negative effect of partnerships on CS2 exam performance for women is half that of men (-1.6% as compared to -3.2%, respectively). This is consistent related work mentioned in Section 1; partnerships can be particularly beneficial to women in introductory computer science courses.

When examining the effect of partnerships on CS2 performance for different GPA groups, we see that the associated benefit of partnerships decreases with higher GPA. The first quartile (lowest) GPA group had an average of 8.8% higher on project scores when partnering, while the third quartile only had an increase of 2.1% while partnering. Additionally, when looking at exam scores in CS2, we observed that the second and third quartile groups had the most negative association with partnerships (-3.3% and -3.8%, respectively). Overall, we observed that partnerships had the greatest associated benefit for students with lower GPAs, but have a diminished, if not adverse association with students in the middle two quartiles.

When analyzing the effect of partnerships on CS3 performance for men and women, we found only one statistically significant observation. Men who partnered in CS2 had an average project score in CS3 4.6% higher than those who had worked alone in CS2.

When analyzing the effect of partnerships on CS3 performance for different GPA groups, we can make two statistically significant observations. First, we see that students in the lowest GPA quartile do better on projects in CS3 after partnering in CS2. This suggests that partnerships may help students in the first quartile increase project performance in future classes. Secondly, we see that students in the fourth quartile tend to do slightly worse on exams in CS3, when they choose to partner in CS2.

### 4.4 Limitations

While our experiments were designed to control as many variables as possible, there are several threats to validity.

First, students had the choice to partner on projects in their CS2 course; and furthermore they had their choice of partners. While this is a practical approach used in many computer science courses, there were factors we could control in this study. Specifically, we did not control the

composition of the set of students who chose to partner may be different than the set of those who chose not to partner.

Another limitation to our study is that we had did have control over group dynamics. While the instructors strongly encouraged students practice pair programming, we do have data on the extent to which students carried out the pair programming strategy. For example, it is possible that students who chose to partner ended up using a "divide-and-conquer" approach to the project work, with the result of missing sections of the project. However, we point out, that this study remains relevant to pair programming in college courses, since this course organization is common, and often times it may not be feasible to monitor whether students use the pair programming strategy.

Finally, while the courses in our study remained largely consistent in lecture content, project requirements and exam content during the semesters we gathered our data, there may be small differences from semester to semester. For example, while exams covered the same material and strove for commensurate level of difficulty, they were not identical.

## 5 Conclusions

In computer science, group work and pair programming are common approaches to solving problems, both in industry and in academia. Prior work has shown that pair programming is associated with improved project work, compared to working alone. In our study, we examined data from 2,234 students enrolled in CS2 and CS3 courses at a large, highly ranked, public research institution. CS2 allows students to complete projects in optional partnerships, while in CS3, students work alone. Our study examined the effects of partnering both during the semester the partnership took place (CS2) and in the subsequent course (CS3).

In our results, we observed associations between partnership participation and increased project performance in CS2. Specifically, we saw that women scored an average of 5% higher on projects in CS2; nearly double the benefit observed for men. We also saw that for student in the lowest GPA quartile, partnerships were associated with 8.8% higher project scores on average. We also observed that students in partnerships withdrew from the course at lower rates. On the other hand, we also observed that students working alone were associated with higher CS2 exam scores.

In the subsequent CS3 course, we observed that students in the lowest GPA quartile who partnered in CS2 were associated with 9.2% higher project scores in CS3, on average. Those in the highest GPA quartile who worked alone scored an average of 3.8% higher on exams.

Overall, we saw that partnerships were mostly associated with increased project performance in both CS2 and CS3; especially among those in the lowest GPA quartile. Additionally, we observed that working alone was mostly associated with higher exam scores in both CS2 and CS3; especially among those in the highest GPA quartile.

## References


[1] Charlie McDowell, Linda Werner, Heather Bullock, and Julian Fernald. The effects of pair-programming on performance in an introductory programming course. In *Proc. SIGCSE*, pages 38–42, 2002.

[2] Nachiappan Nagappan, Laurie Williams, Miriam Ferzli, Eric Wiebe, Kai Yang, Carol Miller, and Suzanne Balik. Improving the CS1 experience with pair programming. In *Proc. SIGCSE*, pages 359–362, 2003.

[3] Charlie McDowell, Linda Werner, Heather E. Bullock, and Julian Fernald. The impact of pair programming on student performance, perception and persistence. In *Proc. ICSE*, pages 602–607, 2003.

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

[5] Grant Braught, L. Martin Eby, and Tim Wahls. The effects of pair-programming on individual programming skill. In *Proc. SIGCSE*, pages 200–204, 2008.

[6] Krissi Wood, Dale Parsons, Joy Gasson, and Patricia Haden. It's never too early: Pair programming in CS1. In *Proc. Australasian Computing Education Conference*, pages 13–21, 2013.

[7] Linda L. Werner, Brian Hanks, and Charlie McDowell. Pair-programming helps female computer science students. *Journal on Educational Resources in Computing (JERIC)*, 4(1), 2004.

[8] Emilia Mendes, Lubna Basil Al-Fakhri, and Andrew Luxton-Reilly. Investigating pair-programming in a 2nd-year software development and design computer science course. In *Proc. SIGCSE Conference on Innovation and Technology in Computer Science Education*, pages 296–300, 2005.

[9] Emilia Mendes, Lubna Al-Fakhri, and Andrew Luxton-Reilly. A replicated experiment of pair-programming in a 2nd-year software development and design computer science course. In *Proc. SIGCSE Conference on Innovation and Technology in Computer Science Education*, pages 108–112, 2006.

[10] Laurie Williams, Robert R. Kessler, Ward Cunningham, and Ron Jeffries. Strengthening the case for pair programming. *IEEE Software*, 17(4):19–25, 2000.

[11] Alistair Cockburn and Laurie Williams. Extreme programming examined. chapter The Costs and Benefits of Pair Programming, pages 223–243. Addison-Wesley, Boston, 2001. ISBN 0-201-71040-4.