

ON SELF-SELECTED PAIRING IN CS1: WHO PAIRS WITH WHOM?*

*Vincent A. Cicirello
Computer Science and Information Systems
The Richard Stockton College, Pomona, NJ 08240
(609)-626-3526
cicirelv@stockton.edu*

ABSTRACT

In recent years, we have seen an increased use of pair programming in CS-1 courses. Widespread evidence shows pair programming as beneficial at all levels of programming skill. During three academic years, we studied what students in CS-1 choose when given the option to either pair program or to program independently. We explore which students choose to pair and how often, and examine the majors and academic years of the self-selected pairs. Among our objectives is to learn about student pair preferences to help guide the assignment of partners when mandatory pair programming is used. We show that Math majors are more likely than Computer Science/Information Systems (CS/IS) majors to prefer pair programming in CS-1. Math majors prefer pairing with Math majors. CS/IS majors more often choose to pair with non-majors, perhaps indicating self-chosen mentoring roles.

INTRODUCTION

In recent years, instructors of CS-1 have increased use of pair programming, where pairs of students program together from a single computer. One member of the pair controls the computer, writing code; while the other reviews code for syntax and logic errors and plans algorithmic strategies. The roles are reversed periodically. Pair programming originated with extreme programming [1], but there is widespread evidence that pair programming is beneficial at all levels of programming skill including in educational settings. It has been shown to increase student retention (e.g., [4]), to

* Copyright © 2009 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

improve individual programming ability (e.g., [3]), and has been shown to particularly improve retention of women as computer science majors [8].

We present a study of pair programming in CS-1 at a public liberal arts college, using data from nine CS-1 sections across 3 years with total enrollment of 219. Our CS-1 course, titled “Programming and Problem Solving I”, is a core requirement for Computer Science and Information Systems (CS/IS) majors as well as for Math majors. In our study, we explore which students choose to pair program when it is an optional, rather than required, aspect of programming assignments. Our aim is to determine if a student’s major or academic year influenced their embracing of the pair programming process.

Our objective is to gain insight into how best to assign partners when pair programming is a rule rather than an option. Research on pair compatibility provides mixed guidance on how to match students, ranging from recommending similar perceived skill level pairs [9], especially for freshmen [2] to indications that the vast majority of pairings are compatible [9] which seems to imply that random pairings are sufficient. Others have shown that allowing students to self-select partners is extremely effective, noting that instructors only need to find partners for a small number of introverted students [6]. However, finding compatible partners is most important for this group. We aim to examine whom students choose to pair with when they self-select their partners, and to use the results to motivate some partner forming guidance.

Among our results, we show that Math majors are more likely than CS/IS majors or non-majors to embrace pair programming, and tend to pair with other Math majors; while CS/IS majors who choose to pair usually pair with a non-major.

OUR CS-1 COURSE

We introduce object-oriented concepts early in our CS-1/CS-2 sequence. We use Cay Horstmann’s “Java Concepts” textbook [5]. The pre-requisite for our CS-1 is Discrete Math, Calculus, or concurrent enrollment in one of these. In CS-1, we provide a general overview of programming, and proceed by covering how to manipulate objects. We spend a significant amount of time on implementing classes. We then cover primitive data types, decision statements, loops and iteration, arrays, debugging. We integrate unit testing throughout. We do not cover any of the more advanced object-oriented concepts in CS-1 such as inheritance, interfaces, etc (we continue with those in CS-2).

The content and sequence of topics remained constant across all nine sections of CS-1 in this study, and the same instructor taught all during a three-year period. During that time, other faculty taught four other sections of the course, but those sections were not used in this study. The instructor significantly uses “live” programming [7].

STUDENT DEMOGRAPHICS

CS-1 is a core requirement for the CS/IS major and for the Mathematics major. CS and IS are not separate majors. Instead, we have a single integrated major, and students choose a concentration in CS or IS. The two dominant groups are CS/IS (32%) and

Math (39.3%) majors, together comprising 71.3% of the students. Approximately 11% have not declared a major. The next two largest groups are Business (5.9%) and Physics (5.0%). Several other majors are represented by no more than 4 students each.

The largest group by academic year are freshmen (37.9%) since a majority of CS/IS majors take CS-1 in the first year (nearly 56% of CS/IS majors in the study). There is a significant group (22%) of Math majors who take CS-1 in the first year. This is also the most common time for undeclared students to take CS-1 (half of the undeclared students). The next largest group are sophomores (28.3%) including CS/IS majors who either decided their major after beginning their college career or who were not ready for CS-1 in the first year (e.g., perhaps needed to increase their mathematics skills). A significant number of Math majors also take CS-1 in the sophomore year, as well as undeclared students. Juniors (25.1%) make up another significant group, largely because junior year is the most common time for Math majors at our institution to take CS-1 (41% of the Math majors). The remaining 8.7% are seniors (almost entirely Math majors).

Of the 219 students enrolled in the course, 201 remained in the course to the end. The other 18 students (8.2%) withdrew prior to course completion.

PAIR PROGRAMMING AS AN OPTION

At the start of the course, the students are introduced to pair programming. They are given rules on how to pair program, and are told that if they choose to pair program for assignments that all code must be written from one computer while both students are present. The two students must alternate roles periodically, where the roles include what I describe to them as “pilot” and “co-pilot”. The current pilot operates the computer, writing code, test cases, etc; while the co-pilot reviews what the pilot is doing, looking for syntax and logic errors, offering algorithmic suggestions, etc. We recommend they balance the time as pilot and co-pilot as evenly as possible. Every few weeks the students are reminded of the proper way to pair program.

Minimal attempts are made to ensure that the pair programming protocol is followed. The course is structured to have a 75-minute lab session each week, in which some time is set aside for students to begin working on the next programming assignment. This time is generally not sufficient for students to complete the assignment, with significant time expected outside the classroom. However, it does offer the instructor an opportunity to ensure that those students, who are choosing to pair program, begin to do so properly. Although what they do once they leave the classroom is beyond the instructor’s control, the classroom offers an opportunity to remind students to switch roles and so forth. In an attempt to discourage the addition of a student’s name to an assignment worked on by one individual, the students are told that the instructor notes who started to work independently and who is pair programming with whom.

Who Chooses to Pair and How Often?

Table 1 shows the frequency with which students chose to pair. Each section of the course had eight programming assignments throughout the semester. The Table shows

how many students chose to pair program for all 8, for 7 out of 8, for 6 out of 8, and so forth. The last column shows the number of students who chose to work independently. The Table also shows the percent of students and cumulative percentage. The “cumulative %” rows indicate the percent of students who chose to pair program for at least that number of assignments. The Table summarizes the data for the entire set of 201 students who completed the course, as well as by major and academic year.

Approximately 39% chose to pair program for at least one assignment. However, less than 20% chose to pair program for 3 or more assignments, with approximately 17% pair programming for at least half of the 8 assignments. Academic year does not seem to influence student interest in pair programming in any significant way.

Table 1: The frequency students choose to pair program

All 201 students									
	Number of assignments paired out of eight								
	8	7	6	5	4	3	2	1	0
#	11	5	5	4	9	4	20	21	122
%	5.5%	2.5%	2.5%	2.0%	4.5%	2.0%	10.0%	10.5%	60.7%
Cumulative %	5.5%	8.0%	10.5%	12.4%	16.9%	18.9%	28.9%	39.3%	100.0%
All 63 CS/IS majors									
	Number of assignments paired out of eight								
	8	7	6	5	4	3	2	1	0
#	1	1	1	0	1	2	5	5	47
%	1.6%	1.6%	1.6%	0.0%	1.6%	3.2%	7.9%	7.9%	74.6%
Cumulative %	1.6%	3.2%	4.8%	4.8%	6.4%	9.5%	17.5%	25.4%	100.0%
All 83 Math majors									
	Number of assignments paired out of eight								
	8	7	6	5	4	3	2	1	0
#	8	4	4	1	7	1	11	12	35
%	9.6%	4.8%	4.8%	1.2%	8.4%	1.2%	13.3%	14.5%	42.2%
Cumulative %	9.6%	14.5%	19.3%	20.5%	28.9%	30.1%	43.4%	57.8%	100.0%
All 55 students majoring in something other than CS/IS or Math									
	Number of assignments paired out of eight								
	8	7	6	5	4	3	2	1	0
#	2	0	0	3	1	1	4	4	40
%	3.6%	0.0%	0.0%	5.5%	1.8%	1.8%	7.3%	7.3%	72.7%
Cumulative %	3.6%	3.6%	3.6%	9.1%	10.9%	12.7%	20.0%	27.3%	100.0%
All 69 upperclassmen (Juniors and Seniors combined)									
	Number of assignments paired out of eight								
	8	7	6	5	4	3	2	1	0
#	3	3	2	0	7	0	9	6	39
%	4.4%	4.4%	2.9%	0.0%	10.1%	0.0%	13.0%	8.7%	56.5%
Cumulative %	4.4%	8.7%	11.6%	11.6%	21.7%	21.7%	34.8%	43.5%	100.0%
All 132 underclassmen (Freshmen and Sophomores combined)									
	Number of assignments paired out of eight								
	8	7	6	5	4	3	2	1	0
#	8	2	3	4	2	4	11	15	83
%	6.1%	1.5%	2.3%	3.0%	1.5%	3.0%	8.3%	11.4%	62.9%
Cumulative %	6.1%	7.6%	9.9%	12.9%	14.4%	17.4%	25.8%	37.1%	100.0%

The number of students who chose to pair program varies by major. Math majors appear to be far more interested in pair programming than CS/IS majors or non-majors. Nearly 60% of Math majors chose to pair program for at least one assignment, compared to just over 25% of CS/IS majors, and approximately 27% of non-majors. Math majors are also far more likely to continue to choose pair programming for subsequent assignments, with nearly 30% pair programming for at least half (4) of the assignments and almost 20% choosing to pair for at least three-quarters (6) of the assignments. Contrast this to CS/IS majors where only 6% chose to pair program for four or more assignments and just 5% chose to pair for 6 out of 8 assignments. Only 11% of students majoring in something other than CS/IS and Math chose to pair program for 4 or more assignments and less than 4% of them chose to pair program for 6 or more assignments.

The author speculates that the reason CS/IS majors seem less likely to willingly choose to pair program is that perhaps they feel they need to work on assignments themselves to thoroughly learn the material. They may need extra encouragement to convince them of the benefits of pair programming. The obvious solution is to require pair programming of all students, but this data is still indicative of the need to reinforce the reasons behind the pair programming process.

Math majors, however, are especially interested in pair programming. As a core requirement of the Math major, all Math majors must take our CS-1 course. Since I have been teaching CS-1, I often find a few Math majors per semester who are apprehensive at the start of the semester, with several of these students turning out as the “star” students in the course. These apprehensive students probably simply have no idea of what to expect, and assume the course will be unfamiliar. These students may be comforted by the ability to work on assignments with a teammate. It is also likely a novel element of coursework for Math majors, since there are probably few (if any) Math courses that involve collaborative problem solving on homework assignments.

Table 2: Number of assignments (N=143) submitted by "pairs" by: (a) academic year, and (b) major.

	FR	SO	JR	SR
FR	29 (20.3%)	25 (17.5%)	16 (11.2%)	2 (1.4%)
SO		16 (11.2%)	14 (9.8%)	3 (2.1%)
JR			19 (13.3%)	15 (10.5%)
SR				4 (2.8%)

(a)

	CS/IS	Math	Other
CS/IS	6 (4.2%)	6 (4.2%)	28 (19.6%)
Math		82 (57.3%)	18 (12.6%)
Other			3 (2.1%)

(b)

Other non-majors may choose to work independently for reasons similar to those of CS/IS majors. Non-majors who take CS-1 as an elective probably have some personal interest in doing so and may incorrectly feel that they will get more from the class if they work on all assignments alone. Also, they are less likely to know other students in the class or might have a schedule that makes it hard to work outside of class with

other CS-1 students, since there are so few CS-1 students with majors other than CS/IS or Math.

Who Pairs With Whom?

In this Section, we explore the majors and academic years of the students' chosen partners. First, consider the frequency that pairing students switch partners from one assignment to the next. Eighty students chose to pair program for at least one assignment. Of those 80, 59 of them pair programmed at least twice. All but 7 of them continued with the same partner. Of the "repeat" pair programmers, approximately 88% of them were happy enough with their initial partner to continue to pair program with them.

In Table 2(a), we summarize the academic years of the pairing students. A total of 143 programming assignments were submitted by "pairs" in the study. This Table shows how many of those 143 assignments were submitted by pairs consisting of different combinations of academic year. The first thing to note is that students are more likely to pair with a student in the same academic year than with someone who is ahead of them. In fact, nearly half of the paired assignments (47.6%) were submitted by same academic year pairs (see the diagonal cells of the Table). This is not surprising since students are more likely to know other students from their academic year due to encountering them in other courses or in the dorms. It is also consistent with previous findings indicating that students prefer to pair with those perceived to be of the same skill level (e.g., [9, 2]).

A significant number of pairings also involve students one academic year apart (e.g., freshmen/sophomore, sophomore/junior, or junior/senior pairs). Such pairings account for 37.8% of the paired assignments submitted. It is not uncommon for students to be in courses with students a year ahead or behind them, so these students may already know each other, or again may perceive each other of the same skill level. Only 13.3% of the assignments submitted by pairs were from pairs where the students were two academic years apart (i.e., sophomore/senior or freshmen/junior), and only 1.4% were freshmen/senior pairs. These 14.7% of the pairings are possibly mentoring relationships.

In Table 2(b) we summarize the pairings by the majors of the pairing students. The most prominent finding is that well over half of all assignments submitted by pairs (57.3%) were submitted by pairs of Math majors. This is likely related to what we saw earlier where Math majors are far more likely to embrace pair programming.

It is worth noting that very few pairs consist of one CS/IS major and one Math major (only 4.2% of the pairings). This is somewhat surprising considering that these two majors are the two largest groups of students in the course. It is likely related to the low number of CS/IS majors who chose to pair program in the study.

Another interesting finding is that CS/IS majors who chose to pair program were far more likely to pair with a non-major. Nearly 20% of all assignments submitted by students who pair programmed were submitted by pairs consisting of one CS/IS major and one non-major student. Although we saw earlier that non-majors are far less likely to choose to pair program than Math majors, it appears that the non-majors who chose

to pair program chose to pair with CS/IS majors, evidence of self-selecting mentoring relationships (i.e., the major mentoring the non-major).

CONCLUSIONS

We examined which students choose to pair program in CS-1 when it is an option rather than a rule. We discovered that Math majors are far more likely to choose pair programming and that they most often choose to pair with another Math major. The majority of CS/IS majors who chose to pair program chose to do so with a non-major partner. This perhaps indicates that CS/IS majors are self-selecting mentoring roles. Our results seem to indicate that students prefer to pair with students in the same or similar academic year. This study provides us with guidance for pair formation in situations where we use pair programming as a rule. CS/IS majors are perhaps best paired with students from other majors but in the same academic year. Math majors in CS-1 are perhaps most compatible as a pair with another Math major.

REFERENCES

- [1] K. Beck and C. Andres. *Extreme Programming Explained*. Addison-Wesley, 2005.
- [2] J. Bevan, C. McDowell, and L. Werner. Guidelines for the use of pair programming in a freshmen programming class. In *Proc. 15th Conf. on Software Engineering Edu. and Training*, pages 100–107, 2002.
- [3] G. Braught, L. M. Eby, and T. Wahls. The effects of pair programming on individual programming skill. *SIGCSE Bulletin*, 40(1):200–204, 2008.
- [4] C. McDowell, L. Werner, H. Bullock, J. Fernald. Pair programming improves student retention, confidence, and program quality. *CACM*, 49(8):90–95, 2006.
- [5] C. Horstmann. *Java Concepts*. Wiley, 5th edition, 2008.
- [6] N. Jacobson and S. K. Schaefer. Pair programming in cs1: Overcoming objections to its adoption. *SIGCSE Bulletin*, 40(2):93–96, 2008.
- [7] J. Paxton. Live programming as a lecture technique. *J. of CCSC*, 18(2):51–56, 2002.
- [8] L. Werner, B. Hanks, and C. McDowell. Pair programming helps female computer science students. *J. on Educational Resources in Computing*, 4(1), 2004.
- [9] L. Williams, L. Layman, J. Osborne, and N. Katira. Examining the compatibility of student pair programmers. In *AGILE’06*, pages 411–420, 2006.