

The Impact of Placement in Introductory Computer Science Courses on Student Persistence in a Computing Major*

Linda Ott
Michigan Technological University
Houghton, MI, USA
linda@mtu.edu

Briana Bettin
Michigan Technological University
Houghton, MI, USA
bcbettin@mtu.edu

Leo Ureel II
Michigan Technological University
Houghton, MI, USA
ureel@mtu.edu

ABSTRACT

Multiple studies have shown that when novice and experienced programmers are enrolled in the same introductory programming course, the novice programmers are negatively impacted. We have two entry points into our course sequence for majors. One course is intended for students with little or no programming experience, while the alternate course is intended for students who have had previous programming experience. In 2015 we discovered that many students with programming experience were enrolling in the course for novice programmers. A change in our placement strategy in 2016 resulted in a greater portion of the students with programming experience actually enrolled in the course intended for students with programming experience. Last year we reported on the impact this change had on the courses and the students enrolled in these introductory courses. Although student performance improved only slightly, many fewer students with little or no previous programming experience reported that their first programming course was unreasonably difficult in 2016. In this paper we examine how this change in placement strategy and resulting changes in the courses is impacting student persistence in the major. Initial indications are that a greater percentage of students with little or no previous programming experience are persisting in their computing major when these students begin in an introductory course that does not also include students who have substantial programming experience.

CCS CONCEPTS

• **Social and professional topics** → **Computer science education**; CS1;

KEYWORDS

Computer science education, CS1, placement, retention, persistence

ACM Reference Format:

Linda Ott, Briana Bettin, and Leo Ureel II. 2018. The Impact of Placement in Introductory Computer Science Courses on Student Persistence in a Computing Major. In *ITiCSE '18: ITiCSE '18: Innovation and Technology in*

*This work is supported by NCWIT's Extension Services for Undergraduate Programs under grant HRD-1203148/1203198/1203174/1203179 and the National Science Foundation under grant DUE-1504860.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ITiCSE '18, July 2–4, 2018, Larnaca, Cyprus

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5707-4/18/07...\$15.00

<https://doi.org/10.1145/3197091.3197139>

Computer Science Education, July 2–4, 2018, Larnaca, Cyprus. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3197091.3197139>

1 INTRODUCTION

This research began as part of our efforts to increase the diversity of our undergraduate computing majors. Our University is a technological university with the vast majority of students majoring in STEM-related fields. As such, the University has not achieved the gender balance that many universities have. In Fall 2016 the campus student population was 27% female, an all-time high. In this context, it has been extremely difficult to attract a gender-balanced student population to our computing majors. This study began shortly after the particularly troubling decade between 2000 and 2010 when on average only 5% of the graduating classes from the Department of Computer Science were female.

In 2014 the Departments of Computer Science and Electrical and Computer Engineering began participating in the National Center for Women & Information Technology's Extension Services for Undergraduate Programs [1]. This program aims to develop strategies to recruit and retain higher numbers of women into computer science and engineering programs. We were required to survey our first-year students during the Fall and Spring semesters to better understand what attracts students to our majors and identify issues that our first-year students encounter. Our first-semester survey and analysis for Fall 2015 was previously published [3].

When analyzing the first-semester survey conducted Fall 2015, we found a substantial mismatch between the percentage of students reporting programming experience and the student enrollments in the first-semester programming courses. CS has offered two entry points for the introductory programming sequence for over 20 years. CS1121 is a CS1 course intended for students with little or no programming experience while CS1131 covers both CS1 and CS2 material at an accelerated rate. This latter course is intended for students with experience, but not sufficient knowledge to place out of the first course. Both CS1121 and CS1131 utilize pair or group programming in the lab sections. Thus, students regularly interact with their peers on programming problems.

Research indicates that when both novice and experienced programmers are enrolled in the same course, the novice programmers are negatively impacted [4, 6, 8, 9, 12, 16, 18]. This can be partially explained by research showing that students often develop their sense of self-efficacy and belonging in a field by comparing their performance with their perception of their classmates' performance [8, 9, 12, 16, 18]. A mixture of novice and experienced programmers in a course can also create difficulties for the instructor. When many students have seen the material previously, the instructor can get the impression that the class has caught on quickly to the material

	Fall 2015		Fall 2016	
	# of Majors	% of Majors	# of Majors	% of Majors
CS1121	83	69%	77	51%
CS1131	23	18%	45	30%

Table 1: Survey distribution in introductory courses

even though the novice students are struggling to keep up. Often, student complaints are bimodal – a portion feels the course is too fast while others feel the course is too slow.

As our introductory sequence was designed with multiple entry points based on programming background, we believed we were in alignment with best practice. However, when we surveyed students in Fall 2015, many students who were enrolled in CS1121 indicated they had previously programmed. Our second-semester follow-on survey indicated that many of the students who had taken CS1121 had substantial programming experience. Student attitudes expressed in the survey, as well as an analysis of their performance, indicated the current enrollment practices were having a negative impact on the students’ experience and success in these courses.

Prior to Fall 2016 our majors were automatically scheduled for CS1121 as part of their initial schedule. Students who were interested in taking the accelerated course took an assessment during orientation. Those achieving a satisfactory score on that assessment were given permission to enroll in CS1131. Many students with programming experience were opting not to take the assessment and simply remaining in CS1121. For Fall 2016 we modified the procedures used to place students. Our goal was to enroll more of the students with experience in CS1131 and consequently decrease the number of students with experience enrolled in CS1121.

The change involved the timing and presentation of the assessment and initial course placement. Rather than having the assessment after students arrived on campus for the fall semester, the assessment was offered online over the summer. Students were emailed information about the programming assessment about the same time they received information about a math assessment. The math assessment is required of all first-year students in order to be registered for a math class. Computing majors were told the CS assessment was also mandatory. After responding to the first part of the CS assessment - which asked questions about their programming background, students were allowed to opt out of the programming portion of the assessment if they had no previous programming experience. Approximately 80% of our majors took the assessment during the summer of 2016 with roughly 25% of the students opting out of the programming portion. In comparison, 22% of our incoming majors took the assessment in 2015. Based on the 2016 assessment, students were then enrolled in either CS1121 or CS1131. As a result, approximately twice as many of our majors were enrolled in CS1131 Fall 2016. See Table 1.

In a previous paper we discussed the impact this change had on the two courses and on student performance [2]. In this paper, we examine the impact this change is having on student persistence in the Department of Computer Science.

2 BACKGROUND

2.1 Related Research

The impact of having novice and experienced programmers in the same introductory course is not restricted to that course, but can impact student persistence in the major. Students with low self-efficacy often come to believe they are not suited for programming [14], often resulting in decreased effort in the course [4, 6, 8, 9, 12, 16, 18]. Ultimately many of these students withdraw from the course and switch majors [11].

A contributing factor is group programming. Compatibility of students in pair programming is often not determined by academic data, but by the students’ confidence in their programming abilities [19]. This again correlates self-efficacy to success. Thus, pairing students who have substantial programming experience with novices can easily lead to negative outcomes in the pair’s programming success as well as negatively impacting the novice programmer’s confidence and sense of belonging. Students are likely to see success in the course as a positive indicator of skill, belonging, and “being a programmer”, thus increasing likelihood of persistence. Related to success, a previous study [15] notes that a student’s desire to learn programming has a large impact on their performance in the course. This desire to learn programming may come from activities prior to the introductory course, but if a course fosters this within students, the course itself can work to reinforce retention.

Indeed, one study [16] concludes that students being placed with similarly skilled peers can create a community sense of belonging via shared commiseration on common problems – this sense of belonging encourages students to persist, as they recognize they are not alone. Such a community cannot exist, however, when student skill levels are incredibly varied. As research in CS persistence has shown, students tend to drop a CS major program within the first two years [7]. Thus, it is critical that a student’s perception of ability and sense of belonging be established early in the program.

For more experienced students there are also potential negative effects when enrolling in the course for novices. As they already know much of the material, these students are likely not engaged in the classroom [5] and establish poor study habits in their major, such as beginning programming assignments at the last minute. They can then be ill-prepared to handle more advanced coursework, lacking appropriate study habits for university level material [5, 17].

Belonging and perception of ability have been correlated positively to retention, more so than even the grades received [16]. To foster strong perception of ability in students, their self-efficacy in relation to their peers must be realized. This makes proper placement of students appear to be essential to build confidence and in turn, retain these students. Similarly, proper placement is valuable for experienced students to combat the potential negative effects of being placed in a course inappropriate for their skill level [4, 8, 18].

Finally, although there is substantial evidence that students develop a stronger sense of self-efficacy in a class with other students of similar experience, providing alternate course options does not necessarily result in increased persistence. A recent study [10] showed no difference in student persistence in the introductory course sequence after an alternate introductory course for students with some programming experience was introduced. Although faculty advisors provided guidance on course selection based on

	Fall 2015 # Responses (Percentage)	Fall 2016 # Responses (Percentage)
CS1121 was too easy	1 (4%)	1 (4%)
CS1121 was taught at an appropriate level for a beginner programmer like me.	3 (12.5%)	4 (15%)
CS1121 was very challenging, but I enjoyed it.	2 (8%)	2 (8%)
CS1121 was a bit intimidating because others clearly had previous programming experience, but I did fine anyhow.	3 (12.5%)	12 (46%)
Because other students have had previous programming experience, CS1121 seems unreasonably difficult for new programmers like me.	11 (46%)	5 (19%)
CS1121 was too difficult for a beginner programmer like me.	4 (17%)	2 (8%)

Table 2: Perceptions of the Level of Difficulty of CS1121 by Students with No Previous Programming Experience

students’ self-reported experience, students were allowed to select their introductory course. The distribution of experienced and novice programmers between the courses was not reported.

2.2 Key findings from previous study

As reported previously [2], the percentage of computing majors who enrolled in CS1121 was reduced in Fall 2016 (see also Table 2). Although the percentage of students enrolled in CS1121 who indicated they had programming experience was similar to 2015’s, the amount of experience the students reported decreased substantially between 2015 and 2016. In particular, in 2015 40% of the students responding to our survey had more than a year of formal coursework in programming. In 2016 only 14% of the students had more than a year of coursework, and no student had more than a year of coursework in Java. In 2015 and 2016 a similar percentage of students were successful in completing their initial course, with success being defined as earning a satisfactory grade, e.g. higher than a “D” in the US system. As more students were enrolled in the accelerated course, this meant that more first-year students were farther along in their courses after Fall semester 2016. The most striking change between the two years, however, was in the students’ perception of the difficulty of CS1121. In particular, 63% of the students with no previous programming experience who were enrolled in CS1121 in Fall 2015 indicated the course was too difficult. That portion dropped to 27% for Fall 2016. See Table 2.

This change in the perceived difficulty of the course is quite likely to impact a student’s sense of self-efficacy and belonging. Indeed, the 46% of the CS1121 class in Fall 2016 that indicated that they “did fine even though others in the course had previous programming experience” were likely to come out of the course with increased confidence in their programming abilities. Succeeding at something that one perceives to be difficult can give a strong sense

	Spring 2016	Spring 2017
# Responses	52	69
Response rate	42%	46%

Table 3: Survey response rates

of accomplishment. Ideally, this increased comfort in the initial course leads to long-term persistence in the major. Here, we examine the persistence of the students in the Department of Computer Science to determine if indeed this change in how students are placed is having a positive impact on student persistence.

3 IMPACT ON PERSISTENCE

As mentioned earlier, CS students typically drop out of the major in their first two years [7]. This broad trend is consistent with student behavior in our Department. Thus, although neither the 2015 nor the 2016 cohort has yet graduated, as the 2016 cohort is now in its 4th semester the current data should be a strong indication of anticipated persistence through graduation. To examine persistence, we reviewed the academic records of the students who enrolled in the Department of Computer Science as first-year students in Fall 2015 and Fall 2016. Students in the Department can major in either computer science (CS) or software engineering (SE). For this study we viewed students in the two majors as a single cohort. Students in both majors take similar courses their first two years.

Students were considered to be persisting if they were enrolled as either a SE or CS major in Spring 2018 and were taking courses appropriate for their major. Thus, a student was not considered to be persisting if they were listed as a CS or SE major, but were taking courses indicating they were considering switching majors. In addition to looking at the overall persistence, we looked more carefully at the students who had responded to our first-year surveys. This allowed us to get a sense of how prior programming experience is impacting persistence. Although our survey response rate was good (see Table 3), breaking the respondents into groups by course placement and previous experience created small subsets. Hence, we are not expecting to find statistical significance, but instead attempting to identify issues and/or trends for any specific subset.

Table 4 shows the initial enrollments by course and the number and percentage of students who are persisting in the major as of Spring 2018. Since we are primarily focused on the introductory courses, students who entered with CS Advanced Placement credit or transfer credit and therefore started in more advanced courses are not included in Table 4. Thus, the percentages of majors for a given year do not add up to 100%.

As can be seen in Table 4, the overall persistence has increased. Of the majors who enrolled in CS1121 or CS1131 in Fall 2015, 51% are persisting while for Fall 2016, 61% are persisting. The more dramatic increase in persistence is among students enrolled in CS1131 with 73% of the students persisting compared to 55% from Fall 2015.

We also examined the data by gender, however, because of the small numbers and privacy concerns, we are not reporting that data. Prior programming experience appeared to play a major role in the persistence of female students. We saw much lower persistence among female students who enrolled in CS1121 with no prior

		CS/SE Majors Enrolled	% of Majors Enrolled in Course	persist	% Persist
CS 1121	2015	83	69%	42	51%
	2016	77	51%	42	55%
CS 1131	2015	23	18%	12	52%
	2016	45	30%	33	73%
Overall	2015	106		54	51%
	2016	122		75	61%

Table 4: Student persistence based on all CS/SEs enrolled in CS1121 and CS1131 Fall 2015 and 2016

	2015 cohort		2016 cohort	
	Persist (Percentage)	Non- persist	Persist (Percentage)	Non- persist
W/no programming	4 (31%)	9	9 (45%)	11
W/ Total	16 (47%)	18	32 (62%)	20

(Note since this is a subset of the given class, the percentages persisting do not match the total for the class.)

Table 5: Comparison of persistence for survey respondents starting in CS1121.

programming experience both years. On the other hand, females with prior programming experience persisted at a similar or higher rate to the males, regardless of whether the previous experience was from informal activities or formal coursework.

To better understand how programming experience may impact persistence, we examined the persistence and performance of the students who responded to the surveys in Spring 2015 and 2016. These surveys included questions on the students' previous programming experience.

As seen in Table 5, whether students had previous programming experience or not, a greater percentage in the 2016 cohort compared to the 2015 cohort are persisting. We made some additional observations from the data.

- Student academic performance in the initial course was a strong factor in persistence prior to the change. In the 2015 cohort the students who persisted and had had no prior programming experience performed exceptionally well as measured by grades in CS1121. In 2016 the grades in CS1121 for students with no prior programming experience who persisted ranged between "A" and "C".
- Frequently students who have difficulty in the initial programming course will switch to other computing-related, but less programming-intensive, majors. Both years students did switch to other computing-related majors and it was always to less programming-intensive majors, e.g., systems administration or information systems. However, half as many students in the 2016 cohort compared to the 2015 cohort switched to these majors.

	2015 Cohort	2016 Cohort
CS 1121	1297	1296
CS 1131	1362	1354

Table 6: Mean SAT scores for students responding to surveys.

In order to verify that the cohorts of students had similar academic abilities, we examined SAT scores for factors related to persistence and found no significant differences among the average scores for the various groupings of students. As can be seen in Table 6, the two cohorts were well-matched, at least as measured by SAT scores.

4 DISCUSSION

Although the persistence of students who start in CS1121 only increased a few percentage points, it is important to note that students who are now persisting have less programming experience coming into the course than students in previous years. It is likely that there will always be a higher rate of attrition from CS1121 compared to CS1131, as some of the students enrolling in CS1121 have little or no idea of what programming entails. Thus, it is reasonable to expect that regardless of how accessible and appealing the course is, some students will decide after their initial exposure to programming that a computing major is not for them.

On the other hand, students enrolling in CS1131 do have programming experience and we should thus expect fairly high persistence from that group of students. It is, therefore, encouraging to see that a substantially higher percentage of these students are persisting in the second cohort.

There are factors, in addition to the placement strategy, that are likely having an impact on student persistence. Most significantly, as the student populations for each of the introductory courses have changed, it has been necessary to make changes to both courses to better match the course material and delivery to the current student population. In the past both courses had slowly evolved to accommodate the students that were actually enrolling in the courses, as opposed to focusing on material appropriate for the originally intended student population. Thus, CS1121 was moving at a pace appropriate for students with some programming experience, while CS1131 had evolved into an accelerated course for students with strong programming skills.

CS1121 now starts with a five-week overview of high-level programming concepts. Students are introduced to the concepts of creating variables, logic, iteration, and control structures using Scratch as the programming language. This allows students to explore these concepts without having to struggle with the syntax of a language. These concepts are reviewed as Java is introduced to the students over the remainder of the semester. In addition, a new textbook is being used which focuses on major concepts, along with providing students more hints and clues on programming bugs to avoid. The previous textbook was very comprehensive, which made it difficult for some students to distill out the important concepts.

A side effect of our change in placement strategy is that CS1131 now has broader diversity in terms of student programming experience compared with past years. We attract students from a broad range of high schools. Some students have informal programming

experience, such as from participating in FIRST robotics, an extracurricular activity for students designed to promote interest in technology and robotics. Others have had multiple years of classroom instruction in programming, but perhaps not in Java. The faculty member teaching CS1131 during Fall 2016 was keenly aware of the larger class size and the greater diversity in programming background of the group.

Since we now encourage students with less programming experience to take CS1131, the pace of that course has also changed. More time is being devoted to reviewing fundamental programming concepts and to explaining how those concepts are implemented in Java. Prior to Fall 2016, two to three weeks of the course was devoted to reviewing the material normally covered in CS1121. That material is now spread over five weeks. This makes it more feasible for students with programming experience in a language other than Java to succeed in CS1131. In order to allow for this slower pace, material from the data structures course that had crept into CS1131 has been removed. Students in previous semesters often complained of substantial duplication of material between CS1131 and data structures.

We have not examined the impact of these changes on non-majors enrolled in these courses. CS1121 is required for students in several other degree programs and students minoring in computer science. CS1121 can also be taken as an elective by students in other majors. The Department does not offer a CS0-type course adding to the necessity to ensure that CS1121 truly works for novice programmers.

5 CONCLUSIONS AND FUTURE WORK

In our earlier study, we found that a placement strategy that resulted in an introductory programming class for novices consisting of students with a more homogeneous programming background had a strong impact on the students' perception of the difficulty of their introductory course. This follow-on study indicates that this change in perception is having a positive impact on the students' persistence in their major.

We are continuing to refine the course materials and our placement strategy to improve student performance and persistence. We continue to work with the Registrar's Office and other departments on campus to ensure that students are enrolled in the appropriate course. For example, if a student with Computer Science A Advanced Placement credit takes and passes the programming assessment, it is important that they be enrolled in the second semester programming course (CS1122), and not in CS1131. (The Computer Science A Advanced Placement course taken by high school students covers material equivalent to CS1 using Java). Experience has shown that enrolling students with this advanced placement credit into CS1131 causes similar problems for both the student and the class as does enrolling students with experience into CS1121.

As we continue to modify the introductory courses, a remaining area of concern is the persistence of female students who enter the curriculum with no programming experience. We have seen greater persistence among our female students as our female population has grown. The persistence rate, however, is still very low for females who come with no programming experience. Other researchers

have noted that female students have a negative programming self-efficacy and underrate their abilities compared to their end of year grade [13]. Identifying ways to combat these issues may be what is needed to increase the persistence of female students with no prior programming experience.

REFERENCES

- [1] [n. d.]. NCWIT Extension Services for Undergraduate Programs. <https://www.ncwit.org/project/extension-services-undergraduate-programs>
- [2] Glen Archer, Briana Bettin, Leonard Bohmann, Allison Carter, Christopher Cischke, Linda M. Ott, and Leo Ureel. 2017. The Impact of Placement Strategies on the success of students in introductory computer science. In *Proceedings of Frontiers in Education Conference (FIE) 2017*. 1–9. <https://doi.org/10.1109/FIE.2017.8190526>
- [3] Glen Archer, Leonard Bohmann, Allison Carter, Christopher Cischke, Linda M. Ott, and Leo Ureel. 2016. Understanding Similarities and Differences in Students Across First-Year Computing Majors. In *Proceedings of Frontiers in Education Conference (FIE) 2016*. 1–8. <https://doi.org/10.1109/FIE.2016.7757695>
- [4] Jan Chong and Tom Hurlbutt. 2007. The Social Dynamics of Pair Programming. In *Proceedings of the 29th international conference on Software Engineering (ICSE 07)*. IEEE Computer Society, 354–363. <https://doi.org/10.1109/ICSE.2007.87>
- [5] Sanda L. Christenson, Amy L. Reschly, and Cathy Wylie. 2012. *Handbook of Research on Student Engagement*. Springer-Verlag, New York, NY, USA.
- [6] James P. Cohoon. 2007. An introductory course format for promoting diversity and retention. In *Proceedings of the 38th SIGCSE technical symposium on Computer science education (SIGCSE 07)*. ACM, 395–399. <https://doi.org/10.1145/1227310.1227450>
- [7] Penelope M Huang and Suzanne G. Brainard. 2001. Identifying Determinants of Academic Selfconfidence among Science, Math, Engineering, and Technology Students. *Journal of Women and Minorities in Science and Engineering* 7, 4 (2001), 315–337. <https://doi.org/10.1615/JWomenMinorSciEng.v7.i4.40>
- [8] Neha Katira, Laurie Williams, and Jason Osborne. 2005. Towards increasing the compatibility of student pair programmers. In *Proceedings of the 27th international conference on Software engineering (ICSE 05)*. ACM, 625–626. <https://doi.org/10.1145/1062455.1062572>
- [9] Päivi Kinnunen and Beth Simon. 2011. CS majors' self-efficacy perceptions in CS1: results in light of social cognitive theory. In *Proceedings of the seventh international workshop on Computing education research (ICER 11)*. ACM, 19–26. <https://doi.org/10.1145/2016911.2016917>
- [10] Michael S. Kirkpatrick and Chris Mayfield. 2017. Evaluating an Alternative CS1 for Students with Prior Programming Experience. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE 17)*. ACM, 333–338. <https://doi.org/10.1145/3017680.3017759>
- [11] Robert W. Lent, Steven D. Brown, and Kevin C. Larkin. 1984. Relation of self-efficacy expectations to academic achievement and persistence. *Journal of Counseling Psychology* 31, 3 (July 1984), 356–362. <https://doi.org/10.1037/0022-0167.31.3.356>
- [12] Colleen M. Lewis, Ken Yasuhara, and Ruth E. Anderson. 2011. Deciding to major in computer science: a grounded theory of students' self-assessment of ability. In *Proceedings of the seventh international workshop on Computing education research (ICER 11)*. ACM, 3–10. <https://doi.org/10.1145/2016911.2016915>
- [13] Keith Quille, Natalie Culligan, and Susan Bergin. 2017. Insights on Gender Differences in CS1: A Multi-institutional, Multi-variate Study. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE 17)*. ACM, 263–268. <https://doi.org/10.1145/3059009.3059048>
- [14] Christine Rogerson and Elsie Scott. 2010. The Fear Factor: How It Affects Students Learning to Program in a Tertiary Environment. *Journal of Information Technology Education* 9 (2010), 147–171. <https://doi.org/10.28945/1183>
- [15] Nathan Rountree, Janet Rountree, Anthony Robins, and Robert Hannah. 2004. Interacting factors that predict success and failure in a CS1 course. In *Working group reports from ITiCSE on Innovation and technology in computer science education (ITiCSE-WGR 04)*. ACM, 101–104. <https://doi.org/10.1145/1044550.1041669>
- [16] Nanette Veilleux, Rebecca Bates, Cheryl Allendoerfer, Diane Jones, Joyous Crawford, and Tamara Floyd Smith. 2013. The relationship between learning and ability in computer science. In *Proceeding of the 44th ACM technical symposium on Computer science education (SIGCSE 13)*. ACM, 65–70. <https://doi.org/10.1145/2445196.2445220>
- [17] Ian Ward. 1972. The study habits of college students. *The Vocational Aspect of Education* 24, 58 (1972), 101–104. <https://doi.org/10.1080/03057877280000171>
- [18] Laurie Williams and Robert Kessler. 2002. *Pair Programming Illuminated*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- [19] Laurie Williams, Lucas Layman, Jason Osborne, and Neha Katira. 2006. Examining the Compatibility of Student Pair Programmers. In *Proceedings of the conference on AGILE 2006 (AGILE 06)*. IEEE Computer Society, 411–420. <https://doi.org/10.1109/AGILE.2006.25>