A Support Program for Introductory CS Courses that Improves Student Performance and Retains Students from Underrepresented Groups

Tia Newhall, Lisa Meeden, Andrew Danner
Ameet Soni, Frances Ruiz, Richard Wicentowski
Computer Science Department
Swarthmore College
Swarthmore, PA 19081

ABSTRACT

In line with institutions across the United States, the Computer Science Department at Swarthmore College has faced the challenge of maintaining a demographic composition of students that matches the student body as a whole. To combat this trend, our department has made a concerted effort to revamp our introductory course sequence to both attract and retain more women and minority students.

The focus of this paper is the changes instituted in our *Introduction to Computer Science* course (i.e., CS1) intended for both majors and non-majors. In addition to changing the content of the course, we introduced a new student mentoring program that is managed by a full-time coordinator and consists of undergraduate students who have recently completed the course.

This paper describes these efforts in detail, including the extension of these changes to our CS2 course and the associated costs required to maintain these efforts. We measure the impact of these changes by tracking student enrollment and performance over 13 academic years. We show that, unlike national trends, enrollment from underrepresented groups has increased dramatically over this time period. Additionally, we show that the student mentoring program has increased both performance and retention of students, particularly from underrepresented groups, at statistically significant levels.

Categories and Subject Descriptors

K.3.2 [**Computer and Information Science Education**]: Computer Science Education

General Terms

Human Factors

SIGCSE'14, March 5–8, 2014, Atlanta, GA, USA.

ACM 978-1-4503-2605-6/14/03 ...$15.00.
http://dx.doi.org/10.1145/2538862.2538923.

Keywords

Mentoring, Diversity, CS1, CS2

1. INTRODUCTION

In *Unlocking the Clubhouse*, Margolis and Fisher provide a number of recommendations for making computing curricula more widely accessible [5]:

1. Pay close attention to the quality of the student experience.
2. Accommodate a wide range of computing experience among incoming students.
3. Create a curriculum that reflects the many facets and impacts of computing.
4. Establish structures for under-represented students to come together for support.

Given these broad recommendations there are a number of possible implementations. In revamping our curriculum we focused on addressing the first two recommendations: attending to the quality of student experience and accommodating a wide range of backgrounds. We wanted to create a welcoming and helpful environment for students who might be intimidated by CS culture. Our approach was twofold: first, to improve the structure and content of the course and second, to create a new staff position with the title *Student Mentor Coordinator* to develop and manage a student mentoring program. For this program, we recruit a diverse set of mentors from the pool of students who have successfully completed CS1. The Coordinator trains the mentors to assist current students during in-class lab exercises and evening and weekend study sessions. The Coordinator also monitors the progress of all students taking CS1 and proactively seeks out students at the first sign of trouble to get them assistance immediately.

We first piloted these changes in CS1, and as they proved to be successful we carried them over into CS2. This paper describes the changes we made in detail, and then demonstrates their success at attracting and retaining a more diverse set of students through a longitudinal evaluation of the student population taking our introductory courses. We show that the number of women and underrepresented minorities (URMs) taking CS1 has significantly increased post changes. The demographics of the students taking CS1 now more closely track the demographics of the College as a

whole. One result of our changes is an overall increase in student enrollment in CS: over 55% of the students at the college will take at least one CS course. Moreover, the numbers of women and URMs taking CS2 is increasing as well, but there is still more work to be done here. Furthermore, we have surveyed the student mentors, and the results show that being a part of this support program played an important role in making them feel more confident in their abilities and in their place within the department.

1.1 Related Work

A number of recent studies have looked specifically at how to better attract and retain women students in computer science. Harvey Mudd College instituted three new practices that had a significant impact on creating more gender balance: making CS1 a breadth-first view of the discipline, offering trips to first-year women to attend the Grace Hopper Celebration of Women in Computing conference, and providing research opportunities for women after their freshman year[1]. Other successful initiatives at the college level include requiring students to do pair programming in their introductory courses, offering several themed CS1 options, redesigning CS1 to emphasize applications in areas of interest to women, and creating majors that combine CS with other areas [2, 4, 6].

One promising study done at a collection of eight diverse colleges and universities demonstrated that peer-led team learning (PLTL) increased the participation and success of underrepresented groups in introductory CS classes [3]. This program involved actively recruiting women and minority students to register for computing classes. Then once enrolled, all students were assigned to small groups (4-8 students) that met weekly for several hours to do additional problem-solving sessions with a trained peer leader. Our mentoring program shares some similarities with this one, but is more voluntary in nature. The mentors are available during class and/or lab times to assist students one-on-one or in small groups. The mentors also run weekend and night help sessions to which the entire class is invited, but not required to attend. Despite these differences, the two programs were both successful at recruiting and retaining underrepresented groups.

Student mentoring programs are now becoming more commonplace in Computer Science, however our program incorporates some unique features that we feel have been instrumental to its success. First, the mentors are very integrated into the course; they meet weekly with the faculty member and coordinator to discuss how the class is going, what concepts students are struggling with, and how best to assist students. Feedback from the mentors is taken seriously and modifications are made as a result of their observations. Second we select mentors not based solely on how they performed in the course, but often on other factors such as their friendliness and ability to relate to students who may be struggling. Third we choose mentors in part to retain them in the department; by selecting them, we are demonstrating our confidence in their abilities.

1.2 Institution Background

Swarthmore is a small, highly selective, coed liberal arts college situated in the suburbs of Philadelphia, Pennsylvania. Swarthmore College created a provisional computer science program in 1984, with a single full-time faculty member. The program was converted to a permanent department in 2001 and now has six full-time faculty members. Over the past 30 years the department has grown significantly from enrollments of 30 students per semester in 1984 to 400 students per semester today. We offer both a major and a minor in Computer Science, and currently have 54 majors in the class of 2014, the second largest major at the College behind Economics.

The computer science major at Swarthmore College consists of nine CS courses, and two math courses (discrete and linear algebra are recommended). Majors are required to take three introductory courses: CS1, CS2, and Introduction to Systems. Once these three courses are completed, students are prepared to take any upper-level course. Upper-level courses are divided into three groups: Theory & Algorithms, Systems, and Applications. Majors must take one course from each group. To complete the major, they must complete two elective upper-level courses and a senior project-based seminar.

CS1 serves two important roles in our curriculum. First, it is typically the initial course for potential majors. Second, it serves as a general science course for students who need to fulfill a distribution requirement. Given that it serves these two roles it must be welcoming to students who have never programmed before while also showing students the breadth of the discipline.

Perhaps surprisingly, most of the students who ultimately become CS majors at the College did not try CS while in high school. Instead, they try CS1 as a freshman or sophomore, find that they really enjoy it, and then decide to pursue CS further. Since the pool of potential majors is primarily being drawn from the students enrolled in CS1, it is essential that CS1 attracts and retains a diverse set of students in order to increase the diversity of CS majors.

1.3 Goals and Overview of Changes

We had a number of goals in the re-design of the introductory curriculum. First, and foremost, we wanted to make CS1 accessible to a wider range of students, including students who might not have a strong level of math preparedness. Secondly, we wanted to increase the diversity in CS1 to ultimately match the college-wide demographics. To accomplish this goal, we decided that our program needed to provide more direct support. To do so, we created a new staff position to manage our new support program, which would consist of student mentors. In choosing student mentors, we sought out students from diverse backgrounds to serve as role models. We also hoped that being selected as a mentor in the program would demonstrate our confidence in the students and encourage them to continue on in the major. Lastly, we wanted to ensure that the new students we drew into CS1 would not fall through the cracks, but would experience a warm and helpful environment. By providing more frequent opportunities for feedback (e.g., direct contact, frequent quizzes, lab hours), we could proactively seek out and assist struggling students as early as possible.

2. INTRODUCTORY COURSE CHANGES

We present details of the changes we made to our CS1 and CS2 courses. These include changes to course content and to pedagogical approaches as well as changes to how we evaluate and support students in these classes. Our changes

to CS1 began in Fall 2006. Because of their success, we incorporated similar changes into CS2 starting in Fall 2008.

2.1 CS1 Content Changes

In Fall of 2006, we switched to Python as the primary programming language, after a one year experiment of Java, and several previous years of C. Switching to Python immediately allowed instructors to focus more on problem solving and less on syntax. To detect potential problems early, and provide students feedback quickly, we introduced bi-weekly quizzes throughout the semester, replacing a longer midterm examination from previous semesters. In Fall of 2007, we changed the course title from *Algorithmic Problem Solving* to *Introduction to Computer Science*. While no change in course content accompanied this name change, we wanted to emphasize that any Swarthmore student could take this course, and it was not limited to those with prior CS experience. CS1 meets in a closed lab with 34 computers for interactive, hands-on lectures (2.5 hours per week) and also meets for open lab time each week with the professor to work on weekly assignments (1.5 hours per week).

2.2 Support Program

2.2.1 Role of Student Mentors

A core component of our support framework is the introduction of a new student mentoring program. Our student mentors are assigned to our introductory courses, contributing in two ways; first, student mentors attend class sessions for the course. Second, the student mentors are responsible for running weekly evening help sessions.

In their capacity as student mentors, they serve several roles. During class sessions, the student mentors' main responsibility is to provide assistance to students during in-class exercises. This includes aiding students having difficulty with the exercise/assignment as well as adding context or extra challenges to students who have completed the assignment early. One evening session per week is oriented around the week's lab assignment, allowing students to get one-on-one direct assistance. For CS1, a second session each week focuses on emphasizing course material and preparing for the bi-weekly quizzes.

Student mentors are trained with the primary goal of providing a supportive and friendly environment to introductory students. Student mentors are in place to be helpful, but to also ensure that the students are learning as opposed to simply getting answers. Lastly, mentors provide an important feedback mechanism for faculty, reporting how students are doing overall with the weekly lab assignment and the major sticking points in course material.

2.2.2 Student Mentor Selection

When choosing student mentors, we seek out students who are patient and encouraging when working with their peers. We also look for students who are dependable and prompt. We require that student mentors take a CS class during every semester they mentor, and that they have completed the class they will student mentor for in good standing. Our policy is also generally not to accept seniors as student mentors, due to the goal of the student mentor program to encourage the mentors to considering pursuing a CS major.

We feel it is very important for students to see and interact with role models who are similar to them in the role of mentors. We therefore make sure to balance the student mentors by gender (one male and one female student mentor for each section/shift/lab). We also make efforts to select a diverse range of students from different backgrounds. We tend to prefer students who actually started out in CS1 without much (if any) CS experience prior to Swarthmore. These student mentors often are less intimidating to the students, and they are able to relate to the struggles faced by students coming into the program with no CS background. That is not to say we never hire very experienced students, but those are the minority of our student mentors.

2.2.3 Student Mentor Training

Ongoing training is provided through weekly student mentor meetings, of which thirty minutes are for student mentor training with the Student Mentor Coordinator, and fifteen to thirty minutes are to go over the week's lab assignment with the professor. Weekly topics include general tutoring advice such as "How not to give away answers", and "What to do if a student gets frustrated." We also delve into specific CS topics that are necessary for the student mentors to understand such as "How to draw the function call stack". The student mentor meetings also provide an opportunity for open discussion, in which the student mentors can ask questions and learn from sharing anecdotes about challenging situations they have encountered that week and how they handled them. These anecdotes serve as a starting point for the student mentors to come up with additional tactics that could also be drawn upon in similar situations.

2.2.4 Role of Student Mentor Coordinator

The mentoring program is supported in part by the Student Mentor Coordinator, a 10-month full-time staff member. The coordinator attends introductory course and lab sections, as well as a few evening sessions, in order to observe the student mentors and students. These observations serve two purposes. First, the coordinator draws from these observations in preparing each week's training topics and in giving individualized feedback to the student mentors. Second, during these class observations the coordinator also interacts closely with the students enrolled in the class, providing the coordinator with unique insights into the strengths, weaknesses, and personality traits of all the students in our introductory courses. These insights enable the coordinator to effectively assist the faculty in selecting new student mentors each semester.

The coordinator also holds office hours during which time students enrolled in the introductory classes can come for help. Some students, particularly freshmen, are intimidated about visiting a faculty member during their office hours. However, because the coordinator is not a faculty member, does not teach any courses, and has no role in assigning grades for the class, many students choose to visit the coordinator during office hours. The coordinator performs triage for most of the basic questions. For more complex questions and situations, the coordinator encourages students to visit with the faculty while also informing the faculty member about the student's situation.

The support coordinator also is responsible for scheduling and staffing the evening help sessions, helping to choose student mentors, producing training resources, and running the weekly mentor group meetings. The coordinator is an

important liaison between the faculty and the mentors and between the faculty and the students in the class.

2.3 CS2

Following the observed initial success of the student mentor program in CS1, we have since added mentor support to our intermediate course CS2. Mentors in CS2 started in Fall 2008. Mentors and quizzes both take a reduced role in CS2, as part of our goal is to prepare students for independent learning in upper-level courses without the need for such active student mentor support. Mentors do not attend the lecture part of the course and instead attend lab sections to provide direct assistance on exercises. In addition, student mentors run a single four-hour long help session once per week. In the first few weeks of the semester, CS2 mentors are typically assisting with syntax and basic debugging, as we migrate from Python in CS1 to C++ in CS2. In later weeks, the mentors help students with more advanced debugging, including using a debugger and/or memory leak checkers, unit testing, and improving top-down design. Mentors hold extra sessions during quiz and exam weeks to provide additional support for the theoretical concepts covered in class.

2.4 Costs

The development of a student mentoring program takes on various costs, requiring both institutional and departmental financial support as well as an increase in faculty time.

Financial costs arise from the addition of support staff, i.e., the Student Mentor Coordinator and student mentors. Mentors are paid for attending class and/or lab sessions, evening help sessions, and weekly meetings with faculty and the coordinator. In addition, students receive one hour's wages for time to familiarize themselves with lab assignments. The Student Mentor Coordinator position is a full-time staff member. Lastly, the department provides juice and snacks for evening help sessions.

Faculty incur costs in the form of additional time commitments. These commitments include lab contact hours (90 minutes per lab per week), designing and grading more frequent quizzes, and attending weekly mentor meetings.

3. EVALUATION

We present results from two studies evaluating our project. The first is a longitudinal study evaluating student success, recruitment, and retention in CS1 and CS2. The second is a survey of current and former student mentors. Overall, our studies show many improvements in the representation and retention of students from underrepresented groups since our changes went into effect. They also show that the support program is doing well in meeting its goals both for the students receiving the support and for the students acting as mentors.

Historically women and some US minorities have been very underrepresented in computer science. For example, since 2007 the numbers of students pursuing bachelor degrees in computer science in the US has grown by around 50% [9]. However, the percentage of CS bachelors degrees awarded to woman has not improved, and in fact has stayed near 12% over the last decade, after dropping from a high of about 18% in 2001. Similarly, the percentages of CS bachelors degrees awarded to US underrepresented minority (URM) students has remained very low.

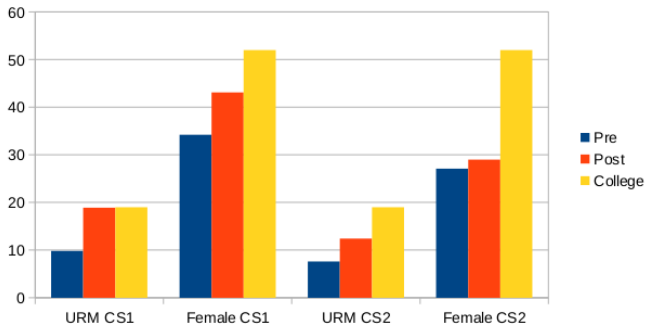

Figure 1: Average Enrollments of URM and Female students in CS1 and CS2 before and after our changes. ***The data are presented as average enrollment percentages over two academic year ranges representing Pre and Post change periods. College-wide average for URM and female students are also shown.***

3.1 Longitudinal Study

We performed statistical analyses of 13 years of data from our CS1 and CS2 courses. The data are from the academic years starting in 1999-2000 and continuing through 2012-2013. We collected final grade data, demographic information (URM status, gender, first generation status), math preparedness measured by Math SAT scores, and help session attendance starting in 2006 for CS1 and starting in 2011 for CS2. In analyzing these data, we asked the following questions about our program:

1. Are there improvements in recruiting and retaining students from underrepresented groups?
2. Is there a correlation between attending help sessions and success in CS1 and CS2?
3. Are there any other predictors of success in CS1 and CS2?

We use enrollment data to illustrate the effects of our program on recruitment and retention of different demographic groups. Figure 1 shows average enrollment data for URM and female students in CS1 and CS2 over two ranges of academic years. The first range (first bar in each group) reflects enrollments prior to our changes (from F99-S07), the second range represents the effects of our changes (from F07-S13) [1]. The last range shows the college-wide demographic information for each group. The graph shows increases in women and URM students in both CS1 and CS2. The increases are statistically significant for both groups in CS1, and for URM students in CS2.

There has also been a significant increase in total enrollments in both courses between the two time ranges: a 96% increase in CS1 enrollments (an average of 75 vs. 147 students per year); and a 64% increase in CS2 enrollments (average of 36 vs. 59 students per year). These data show that as our enrollments have increased so have the percentages of students from underrepresented groups; this differs from national trends where there has been little to no increase in the

[1]We expect to start seeing the effects a semester or two after we first introduced the changes, thus the second range begins in Fall 2007 rather than Fall 2006.

percentage of students from underrepresented groups during this recent period of rapidly increasing CS enrollments.

To evaluate the effects of help session attendance on success as measured by course final grade, we performed an Ordinary Least Squares Regression of the data. The results show that when controlling for math preparedness, there is a a 0.48 improvement in final grade for students who attended CS1 help sessions vs. those who did not. This represents an improvement of one half a letter grade due to help session attendance. We also found that female students attended help sessions more frequently than male students, possibly contributing to their increased representation in the class.

Taking a closer look at how help session attendance affects students within different math backgrounds, we grouped students into three "math preparedness" sets based on Math SAT scores ($<= 700$, $710-750$, > 750). The data show that help session attendance improved final grade within each math grouping. The largest positive effect is on the middle group. However, there is a statistically significant positive correlation between CS1 help session attendance and final grade within each group.

The results for CS2 similarly show that help session attendance is positively correlated with improved final grade. We also found that students' prior CS1 help session attendance was positively correlated with improvements in their CS2 final grade, indicating lasting effects of the support in CS1 on our students' success in CS2. We suspect that the peer support and structured practice provided by these sessions has a positive longterm effect on our students' learning and confidence in future CS courses. Unlike the CS1 results, the CS2 results are are not statistically significant, in large part due to only having two years of attendance data. However, both show stable increasing trends in success in CS2 corresponding to help session attendance.

Finally, our results show that math preparedness is the strongest predictor of success in CS2. This follows results from other studies showing math preparedness as being one of the biggest predictors of success in introductory CS courses [7, 8, 10]. This result indicates that requiring a certain level of math pre-requisite for CS2 may better prepare students for success in the course. Based on this result, we have recently added a recommendation that students take Discrete Math before taking CS2. We are also encouraging student advisees to start in mathematics their first year to better ensure their success in CS.

Overall the analyses yield several statistically significant improvements since implementing our changes. These include an increase in the recruitment and retention of students from underrepresented groups, and a positive correlation between course grade and help session attendance. We have additionally increased the percentages of URM and female students in CS2, although this increase is currently only statistically significant for URM students.

Increasing the numbers of female students taking CS1 and continuing on into CS2 is a crucial first step to increasing the numbers of female students graduating with CS degrees. As can be seen in Figure 2, the percentage of female CS majors at Swarthmore has dramatically increased in the last three years, by far exceeding the national average of 12% [9].

3.2 Student Mentor Survey Results

We conducted a survey of our current and former student mentors, asking them questions about the program and about their participation in the program. The survey included a few quantitative questions asking them to rate their overall experience and several open-ended questions asking them what they thought were strengths and weaknesses of the program and what they felt they got out of participating. We had 27 responses, and the results are discussed below.

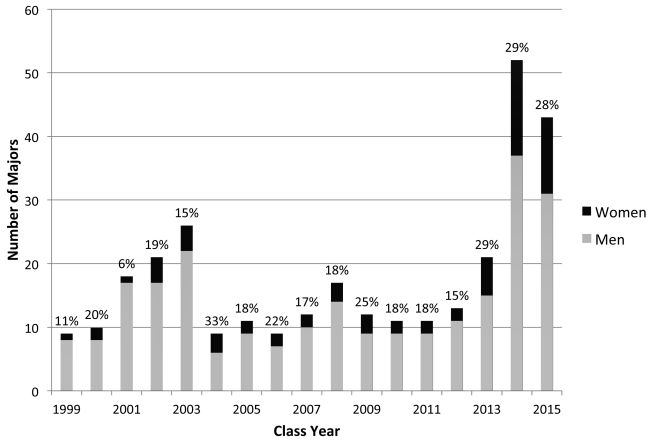

Figure 2: Total number of CS majors at Swarthmore College since 1999 and the percentage of Women vs Men.

Quantitative Results

We asked only a few quantitative questions, most were to evaluate the usefulness of some aspects of their training. A couple, however, asked them to give overall rankings to their experiences. These questions and results are:

1. How would you rate your overall experience as a student mentor?
   Results: On a scale of 1-5, with 5 being best,
   Average: 4.5.
2. Did your experience as a student mentor help you feel more connected, less connected, or no change, to the department?
   Results: **100% said more connected**

These results indicate that the mentoring experience is very positive for the students. Anecdotally, the fact that we have more students express an interest in becoming a mentor than we could ever hire, also speaks to how well our students enjoy being mentors.

Qualitative Results

We asked several open-ended questions about being student mentors, including:

1. Which parts of your experience being a student mentor were most satisfying to you?
2. Why were you initially interested in becoming a student mentor?
3. Did you feel that being a student mentor had any effect on your own abilities as a computer science student?

Again, the results were overwhelmingly positive. In addition to enjoying the process of helping their peers learn,

students commented that serving as a mentor helped them to solidify their knowledge in CS:

- "It gave me a better understanding of the different ways to look at a problem, so now, when I encounter something new, I have more tools available for learning the material and different ways of explaining it to myself."
- "Mentoring has allowed me to see various topics for a second time, giving me a much better understanding of material covered within each course and allowing me to make connections that I missed when I took the class."
- "My understanding of the CS2 material is deeper after spending time teaching it and thinking of multiple ways to explain it, particularly more abstract concepts."
- "It definitely helped my ability to explain difficult concepts and my teaching abilities. As a TA in grad school, my office hours are always the most highly attended and students come see me instead of their assigned TA."

One of the most prominent features of the survey responses was that students who had served as student mentors felt that this program had boosted their confidence and made them feel like more integral members of the department:

- "When I was asked to be a student mentor, it made me feel like an important part of the department which increased my confidence in CS overall."
- "I felt like a staff member and that I had a vested interest in the department. I also felt closer to the profs and more comfortable going to their office hours or asking them to let me do research with them."
- "I felt like a more valuable *asset* to the department, rather than just a student. It definitely upped my involvement and interest inside the department."
- "It felt like affirmation that I was doing well in the department and belonged there."
- "Being a student mentor made me feel I was more deeply a part of the department than I had been before. I ended up getting to know more people, and feeling more connected to the department."
- "I decided to major in CS partially as a result of student mentoring. The confidence shown in me by recommending me as a student mentor made me think that I was smart enough to do computer science, and that kept me going when the course work got difficult."
- "I didn't feel very confident of my place in the CS Department initially. I didn't think of myself as particularly good at science and was concerned that I wouldn't excel in the department. Being a mentor has improved my confidence in myself as a student."

It is clear that the act of being a mentor provided many key benefits to the students. The process of sitting in on classes and labs gave them a better understanding of the introductory curriculum. Running help sessions and interacting with their peers provided them with extra practice at problem solving and explaining abstract ideas. Perhaps most importantly, being chosen to serve in this role validated students and invested them in the department, helping them to persevere and be successful.

4. CONCLUSIONS

Over the seven years that we've been utilizing this mentoring program we've seen dramatic improvements in numbers of students taking our introductory sequence, in overall student success in our courses, and in increased representation and retention of women and underrepresented minorities in CS, beyond national trends. The addition of the student mentoring program has also had a noticeable effect on our departmental culture: students have created their own support groups, including a very active *Women in CS* group; students are more cooperative and helpful in class and lab; students feel more connected to the department; and students are the biggest promoters of our courses, encouraging other students across the College to try CS. Based on the success of this approach, we are now implementing it in our new course *Introduction to Computer Systems*. We continue to evaluate and adjust our program to better meet student needs. We are very pleased with the success of our program, and feel that it could be easily adopted at other institutions.

Acknowledgements

We would like to thank Robin Shores and Jason Martin in the Office of Institutional Research for their invaluable help with the longitudinal study. We also thank Betsy Horner for her help in developing the first student mentor program in our department.

5. REFERENCES

[1] Alvarado, C., and Dodds, Z. Women in CS: An evaluation of three promising practices. In *SIGCSE: Technical Symposium on Computer Science Education* (2001).

[2] Barr, V. Create two, three, many courses: An experiment in contextualized introductory computer science. In *Proceedings of CCSCNE* (2012).

[3] Horwitz, S., Rodger, S. H., Biggers, M., Binkley, D., Frantz, C. K., Gundermann, D., Hambrusch, S., Huss-Lederman, S., Munson, E., Ryder, B., and Sweat, M. Using peer-led team learning to increase participation and success of under-represented groups in introductory computer science. In *SIGCSE: Technical Symposium on Computer Science Education* (2009), pp. 163–167.

[4] Klawe, M., Whitney, T., and Simard, C. Women in computing—Take 2. *Communications of the ACM 52*, 2 (2009).

[5] Margolis, and Fisher. *Unlocking the Clubhouse: Women in Computing*. MIT Press, 2002.

[6] McDowell, C., Werner, L., Bullock, H. E., and Fernald, J. The impact of pair programming on student performance, perception and persistence. In *ICSE Proceedings of the 25th International Conference on Software Engineering* (2003).

[7] Morrison, M., and Newman, T. S. A study of the impact of student background and preparedness on outcomes in CS I. In *SIGCSE: Technical Symposium on Computer Science Education* (2010).

[8] Stein, M. V. Mathematical preparation as a basis for success in CS-II. *Computing Sciences in Colleges 17* (2002).

[9] The Computer Research Association (CRA). The 2012 Taulbee Survey. http://cra.org/resources/taulbee/, 2012.

[10] Wilson, B. C., and Shrock, S. Contributing to success in an introductory computer science course: A study of twelve factors. In *SIGCSE: Technical Symposium on Computer Science Education* (2001).