Improving the Success of Non-Traditional Students in an Introductory Computing Course: An Experience Report

Anonymous for Review

***Abstract*—This paper presents a redesign of an introduction to computing course at a public, minority serving institution in the United States with a majority of non-traditional students. The course redesign was motivated by the desire to improve the success of the students in this course and in the major. Active learning during class and required attendance were the major components of the course redesign. The course policies included flexibility for the occasional absences that are expected with non-traditional students. A comparison of student performance in the experimental and control sections indicated that the requirement of active participation during class is not detrimental to students' performance in the course.**

I. Introduction

This paper presents our experience with redesigning an introduction to computing course (CS0), motivated by the desire to improve the success of non-traditional students in this course and in the major. We intended the work described in this paper to be the ground work for a multi-year longitudinal study of student success in the major. However, after the year that is described in this paper most of the authors of this study changed jobs and the author who remains at the university where this study was conducted holds a contingent position that does not provide much time for working on research. Our experience with redesigning the CS0 course and conducting this study for one year provides the following contributions:

- A framework for using active learning and incentivizing class attendance while providing flexibility for occasional absences.
- An example that illustrates how factors external to the classroom impact student success and the outcomes of this experiment.

We embarked on this study with the hypothesis that an active learning approach in the classroom along with incentives that encourage class attendance and assignment completion would improve student success in this course and have positive impacts on success in courses throughout the major. Prior studies conducted in our department [1] and by the university indicated that the next few courses in the sequence of the computer science and computer engineering majors had relatively low pass rates. While the pass rates in these courses are similar to those seen elsewhere [2], communications from the university administration indicated that it was important for faculty to raise the pass rates in this CS0 course and in courses later on in the major.

This CS0 course was offered at a medium to large sized regional public university that is designated as a minority serving institution in the United States. The majority of students at the university originate from the region where the university is located and are categorized as non-traditional college students. Some of the factors that categorize the students as non-traditional are having significant work and family commitments, being the first in their family to attend college, living at their family home while attending the university, or attending the university five or more years after they graduated high school. Despite the use of the term non-traditional, the majority of undergraduate students in the United States have at least one non-traditional characteristic [3]. The students who take this CS0 course have a wide range of previous experience with Computer Science, as well as a wide range of general college-level skills.

This paper continues in Section II with a discussion of related work. Section III describes the CS0 course and the course redesign experiment. An evaluation of the redesign is presented in Section IV. Finally, this paper concludes with Section V.

II. Related Work

A number of approaches to CS0 courses are discussed in the Computer Science Education literature. The variety of CS0 courses that are described in the literature indicate that there is not a consensus on the definition of a CS0 course, and that different approaches may be appropriate for different institutions. The results from a survey of Computer Science programs in the USA showed that 60.1% of respondents offer a CS0 course [4]. Purposes found in the literature for offering a CS0 course include providing an introduction to programming prior to CS1 [5], and as a way to give non-major students a view of the field of computing [6]. Other universities have noted that they had challenges with retaining students in the major and developed a CS0 course to provide students with additional preparation prior to taking the CS1 course [7].

Incorporating active learning techniques into the classroom has been shown to improve student performance [8], classroom environment, and student retention [9]. Active learning and similar student-centered methods have gained prominence in higher education over the last several decades, and are now accepted as effective, quality education [10]. Studies have also shown that "active learning confers disproportionate benefits for STEM students from disadvantaged backgrounds and for female students in male-dominated fields" [8]. Because the

course described in this paper was offered at a minority-serving institution, these prior studies indicate that the students at this university would benefit from student-centered active learning approaches in the classroom.

III. Overview of the CS0 Course

The course described in this paper is technically two different courses: Introduction to Computer Science taken by Computer Science majors, and Introduction to Computer Engineering taken by Computer Engineering majors. However, these two courses have a significant amount of overlap. Both courses evolved from the same previously offered course, and the same group of faculty generally teach both courses. Therefore, this paper presents a combined description of both courses under the name CS0.

We acknowledge that this CS0 course deviates from the definition of a CS0 course used in much of the related work, as discussed in Section II, because the CS0 course described in this paper is required for the major. We decided to use the term CS0 for this course because it is placed in the curriculum before the CS1 course. The department had three main reasons for requiring all students in the Computer Science and Computer Engineering majors to take the CS0 course. First, we were addressing the retention issues we experienced in these majors and overall retention issues at the university. Research that focuses on professional identity development in computer science has suggested that providing opportunities for students to learn about different areas of computing at an early point in the major could help with retention [11]. The second reason that the department required this CS0 course was so that our series of major courses was similar to those from other departments within our college. The third reason for requiring all students in the major to take this course was because the student advising process at the university did not provide a reliable way for the department to identify students who would benefit from the extra preparation of a CS0 course before the CS1 course. There were enough students who would benefit from taking CS0 before CS1 that it was reasonable to require all students to take the CS0 course.

The instructor for the CS0 course was typically a tenure-line faculty member or a lecturer who had been employed by the department for many years. Additionally, undergraduate or graduate (masters-level) student assistants were assigned to the course. These student assistants attended the class meetings, held office hours, and maintained the LEGO Mindstorms robots.

The course met for two 75-minute sessions per week during a 15-week semester. Because the CS0 course was a lab course where students earned one credit for a three hour per week course, the faculty minimized the amount of work required outside of class time.

A. Control Course

The topics covered in the control version of the course are shown in Table I. The course had a bottom–up approach to covering the breadth of Computer Science starting with bits and ending with programming and applications. Approximately half of class time was used for a lecture on the content and half of the time was used for students to work on the lab assignments. There were approximately ten lab assignments throughout the semester, and students often worked on one lab assignment across multiple class sessions.

TABLE I
Control CS0 Course Topics

Binary and Hexadecimal Representation
Using bits to represent text, images, and sound
Boolean algebra and Circuits
Computer architecture, machine language, and program representation
Algorithms and problem solving: Programming using LEGO Mindstorms Overview of searching and sorting algorithms
Miscellaneous topics: Operating systems, the Internet and the World Wide Web

B. Motivation for Redesigning the Course

As the first class in both the Computer Science and Computer Engineering majors, this is a course where department expectations and culture regarding attendance and assignment completion can be established. Poor attendance and late arrival were issues in all courses in these majors. A prior study of this CS0 course as well as a CS1 course at the same university found that the majority of students who fail the courses are not submitting assigned work [1]. Because much of the graded work for this CS0 class is completed during class time, it seemed that encouraging attendance and participation during class meetings could have a positive impact on the pass rate in this course and may have a positive influence on attendance and assignment completion in later classes in the majors.

We had some hesitation about requiring students to actively participate in class in order to earn the majority of each lab assignment grade because this requirement was a significant change for this course and from the culture within the department. Most of the students are non-traditional students who commute to campus and have significant work and family commitments. Therefore, for the experimental course we developed policies that allow a few absences while maintaining high expectations for class participation.

C. Experimental Course

Three major principles guided the design of the experimental course:

1) To provide better context for the course material;
2) To incentivize attendance and completion of work while accommodating students' external commitments;
3) To increase active learning during class.

1) Adding Context to Course Content: The topics covered in the new CS0 course are shown in Table II. The content of the experimental course was largely the same as the content of the control course because this course fulfills certain learning outcomes as part of the departments' ABET accreditations.

TABLE II
EXPERIMENTAL CS0 COURSE TOPICS

Big Idea	Activity
Algorithms	Programmer and Computer LEGO game
	Logic Puzzles
	LEGO Mindstorms robot programming
	Searching and Sorting Algorithms
Computer Arch. and Data Representation	Fetch, Decode, Execute cycle
	Binary and Hexadecimal Representation
	Two's Complement Representation
	Boolean Algebra
	Gates and Circuits
Social Impacts	Codes of Ethics
	Sustainability
	Diversity and Implicit Bias

For the redesigned course, the content was organized into three "Big Ideas" that present the topics in a top-down order: Algorithms, Computer Architecture and Data Representation, and Social Impacts of Computing. In order to explicitly demonstrate the relationships between the various topics within each unit, a class session at the end of each Big Idea unit was dedicated to discussing the Big Idea and identifying how the lab assignments in that unit relate to the Big Idea. The two motivations for this topic ordering are: (1) to help students better appreciate how these topics relate to Computer Science and Computer Engineering, and (2) to encourage students to become engaged in the course through their enjoyment of working with the popular LEGO Mindstorms robots. The unit on Social Impacts of Computing was a significant addition to the new CS0 course and reflects the computing community's growing recognition of the importance of including ethics and societal issues as part of the computer science curriculum [12], [13].

2) Encouraging Active Learning and Participation: The design of the experimental course focused on three aspects of student participation: active learning, interaction between students, and attendance. The lab assignment structure provided scaffolding for good student habits: prepare for class, actively participate during class, then reflect on the learning. Course policies allowed for occasional absences while maintaining high expectations for participation.

The **pre-lab** was due before class and contributed to 5% to 10% of the lab assignment grade. In order to prepare for the in-class activity, the pre-lab included readings or videos that provided background material and required students to review the in-class activity instructions. Credit was earned for each pre-lab by completing multiple choice questions using the learning management system (LMS).

The **in-class activity** contributed 70% of the lab assignment grade and was completed during one or two class meetings. In order to encourage interaction between students they worked in pairs on each in-class activity. The instructor arbitrarily selected new pairings for each lab with the goal of giving every student the opportunity to work with every other student over the span of the semester. The inclusion of group work in this CS0 course reflects that many studies have demonstrated the benefits of pair programming in computer science courses [14], including some indications that group work can be especially helpful for students who identify with groups that are underrepresented in computer science [15], [16]. Because a late arrival would disrupt the group work process, a late penalty was assessed for any student who arrived after the small groups were formed. The late penalty was a deduction of 10 points on the lab assignment grade.

Students completed a **post-lab** after each in-class activity that contributed 20% to 25% of the total lab grade. The goals of the open-response questions on the post-lab were to reinforce the concepts learned during the in-class activity and to guide students in reflection on how the material relates to the field of computer science.

The following **course policies** provided flexibility for the occasional absences that are expected with non-traditional students:

- The student's two lowest lab assignment grades were dropped when calculating their course grade.
- Three class periods were designated as lab make up days when students could complete the in-class portion of a lab they missed. Other students were encouraged to use this time to review for the upcoming exam.

IV. EVALUATION

The experimental course was taught in two consecutive semesters (fall and spring) within a single academic year. Four sections of the experimental course were offered, with instructors A and B each teaching two sections. Instructor C taught a total of five sections of the control course during the same two semesters. Note that instructor C had limited knowledge about the details of the experimental course until after the end of the academic year, so that the control course was not influenced by this experiment.

Student participation and performance were evaluated by extracting aggregated grade data from the experimental and control sections, as well as historical data from four sections that instructor A taught during the previous four semesters. Assignment completion rates were used to measure participation, and exam scores and course pass rates provided data about student performance.

Assignment completion data, as summarized in Table III, were examined from two perspectives:

1) The fraction of assignments with a non-zero score. That is, (total non-zero assignments)/(total number of assignments).
2) The fraction of assignments with a passing score. This measure was calculated as (total number of assignments with score ≥ 70)/(total number of assignments). A score of 70 is the threshold for passing because the major requires students to earn a course grade of C or better.

The students' grades provide information about their overall course performance. The average exam scores and percent of students who passed the course are summarized in Table IV.

1) The average of the students' scores for all exams was calculated.

TABLE III
ASSIGNMENT COMPLETION

Measure	Experimental	Control	Previous Semesters
Non-zero	99.1%	88.5%	81.5%
Passing	89.8%	86.5%	75.9%

TABLE IV
COURSE GRADES

Measure	Experimental	Control	Previous Semesters
Exam Average	81.0%	82.5%	69.0%
Course Pass	91.8%	95.2%	78.0%

2) Pass rates. A course grade of C or better is a passing grade in the major. The pass rate was calculated as (total number of students with C or better)/(total enrollment for the course). Students who withdrew from the course are excluded.

All of these measures indicate better student performance during the year when this study was conducted (both experimental and control sections) when compared with the data from instructor A's classes in the four previous semesters. Focusing within the year of this study, the experimental sections have a higher percentage of assignments that are submitted (having a non-zero grade) as well as a higher percentage of assignments that have a passing grade, when compared with the control sections during the study period. The reverse pattern is seen in the course grades: the control sections have slightly higher average exam grades and course grades than the redesigned sections.

V. CONCLUSIONS AND REFLECTIONS

The evaluation of course grades shows that the experimental and control courses had similar exam averages and course pass rates. This indicates that requiring non-traditional students to actively participate in class while allowing for occasional absences is not detrimental to student's performance in the course.

An intriguing finding is that participation and performance was better during the year of the study (in both the experimental and control sections) than during the previous four semesters. This finding emphasizes the importance of the control sections in this study. If there were not experimental and control sections during the same academic year, the results would have indicated that the experimental sections had much better student performance than the sections in previous semesters.

A unique factor that may have impacted student performance during the year of this study was a major institutional change at the university. At the beginning of the first semester of this study, two universities in the region merged. We are curious about whether factors such as the media campaign associated with this merger could have impacted students' attitudes towards their college studies. There were record enrollments in the university during the fall semester of this study. The pass rates for the nine sections of CS0 during this year range from 85% to 100% and are among the best pass rates that we ever experienced in our courses at this university.

This paper presented an implementation of a teaching philosophy where high standards for student participation are balanced with a reasonable amount of flexibility. The results from this study indicate that this philosophy did not have detrimental impacts for a cohort of non-traditional students at a minority serving institution. This teaching philosophy is likely to be useful at other universities because the majority of undergraduate students in the United States are non-traditional students, as well as in other contexts where instructors wish to implement flexible requirements for student participation, such as during the COVID-19 pandemic.

REFERENCES

[1] Anonymous, "Reference omitted for anonymous review."

[2] J. Bennedsen and M. E. Caspersen, "Failure rates in introductory programming: 12 years later," *ACM Inroads*, vol. 10, no. 2, pp. 30–36, Apr 2019.

[3] A. W. Radford, M. Cominole, and P. Skomsvold, "Demographic and enrollment characteristics of nontraditional undergraduates: 2011-12," National Center for Education Statistics, Tech. Rep. NCES 2015025, 2015.

[4] S. Davies, J. A. Polack-Wahl, and K. Anewalt, "A snapshot of current practices in teaching the introductory programming sequence," in *Proceedings of the 42Nd ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '11. New York, NY, USA: ACM, 2011, pp. 625–630.

[5] C. Dierbach, B. Taylor, H. Zhou, and I. Zimand, "Experiences with a CS0 course targeted for CS1 success," in *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*, ser. SIGCSE '05. New York, NY, USA: ACM, 2005, pp. 317–320.

[6] M. desJardins and M. Littman, "Broadening student enthusiasm for computer science with a great insights course," in *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '10. New York, NY, USA: ACM, 2010, pp. 157–161.

[7] C. Marling and D. Juedes, "CS0 for computer science majors at Ohio University," in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, ser. SIGCSE '16. New York, NY, USA: ACM, 2016, pp. 138–143.

[8] S. Freeman, S. L. Eddy, M. McDonough, M. K. Smith, N. Okoroafor, H. Jordt, and M. P. Wenderoth, "Active learning increases student performance in science, engineering, and mathematics," *PNAS*, vol. 111, no. 23, pp. 8410–8415, 6 2014.

[9] L. Barker, C. L. Hovey, and L. D. Thompson, "Results of a large-scale, multi-institutional study of undergraduate retention in computing," in *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*. IEEE, Oct 2014, pp. 1–8.

[10] O. Hazzan, T. Lapidot, and N. Ragonis, *Guide to Teaching Computer Science: An Activity Based Approach*, 2nd ed. London: Springer-Verlag, 2014, ch. 2, pp. 15–22.

[11] A. Kapoor and C. Gardner-McCune, "Understanding cs undergraduate students' professional identity through the lens of their professional development," in *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*. Association for Computing Machinery, 2019, pp. 9–15.

[12] L. Carter and C. Crockett, "An ethics curriculum for CS with flexibility and continuity," in *Frontiers in Education Conference*, 2019.

[13] S. Kumar and N. Kremer-Herman, "Integrating ethics across computing: An experience report of three computing courses engaging ethics and societal impact through roleplaying, case studies, and service learning," in *Frontiers in Education Conference*, 2019.

[14] B. Hanks, S. Fitzgerald, R. McCauley, L. Murphy, and C. Zander, "Pair programming in education: a literature review," *Computer Science Education*, vol. 21, no. 2, pp. 135–173, 2011.

[15] K. S. Choi, "A comparative analysis of different gender pair combinations in pair programming," *Behaviour & Information Technology*, vol. 34, no. 8, pp. 825–837, 2015.

[16] L. H. LeGault, "Understanding and supporting better pairings for CS1 students," in *Proceedings of the 2017 ACM Conference on International Computing Education Research*, ser. ICER '17. New York, NY, USA: ACM, 2017, pp. 267–268.