# PREOP AS A TOOL TO INCREASE STUDENT RETENTION IN CS*

*Briana Lowe Wellman, Monica Anderson and Susan V. Vrbsky*
*Department of Computer Science*
*University of Alabama*
*Tuscaloosa, AL 35487-0290*
*(205) 348-6363*
*{blowe, anderson, vrbsky}@cs.ua.edu*

**ABSTRACT**

The demand for computer scientists is expected to continue to increase irrespective of the current state of the economy. Unfortunately, the supply is not expected to match the demand as the number of computer science majors has decreased substantially since the year 2000. As a result, universities and colleges are attempting to identify new ways to attract and retain prospective students into the field of computer science in order to increase the number of majors. In this paper we describe our approach to increase participation and retention through the use of PREOP (Providing Robotic Experiences through Object-Based Programming), an approach that combines the Alice interface and robots for a CS1 Laboratory. PREOP is an interactive 3D animation programming environment, that allows students to program real robots using a drag-and-drop, syntax-free interface. The goal is to foster student motivation and increase student understanding of the fundamental concepts within the first-year curriculum. Initial results indicate that the students in the PREOP Lab who are eligible for CS2 are more likely to rate their skills and knowledge above average than the students in the non-PREOP Labs, and more likely to be registered for the CS2 course than the students in the non-PREOP Labs.

## INTRODUCTION

In early 2009, careers in computer science continue to appear in the Top Ten lists of Best Careers/Best Jobs in such venues as U.S. News and World Report, the Wall Street

---

Journal, Career Builder and Money Magazine [5, 11, 12, 15, 16]. The job opportunities that are listed for computer scientists include the specialties of software engineers, database administrators and computer/IT analysts. Not only are these jobs identified as currently in demand and recession proof, but the demand for these positions is predicted to continue to increase in the future. In fact, the U.S. Department of Labor has predicted a faster than average growth rate for computer scientists at 37% between 2006 and 2016 [3]. Despite this increase in demand, the future supply of well-trained specialists for these careers has decreased substantially and new approaches are being used to increase participation in computing fields.

The statistics regarding the number of students who are CS majors are not encouraging for the field of computer science. A Computing Research Association study indicated a decline of majors in Computer Science between the fall of 2000 and fall 2005. As a result of a continuing decline, the number of new CS majors in the fall of 2007 was half of the number of new majors in the fall of 2000. More encouraging news in the recent report by CRA is the increase in new majors in 2007-08 [17]. Universities and colleges are attempting to identify new ways to attract and retain prospective students into the field of computer science in order to sustain and grow this increase in majors.

The introductory courses in computer science have historically resulted in a filtering out of all but those students enthused by the rigors of traditional programming in a GUI-based development environment. However, as the field of computer science evolves, an increasing diversity of talents and expertise of the specialists involved has become important to the growth and success of the field. Not all students who are able to succeed and thrive in the field of computer science are being captivated by the first CS course. Presenting computing science in a manner to attract an increasing number of students with diverse talents and interests has become a topic of discussion.

In this paper we describe our approach to increase participation and retention through the use of PREOP (Providing Robotic Experiences through Object-based Programming), an approach that combines the Alice interface and robots. The University of Alabama is currently participating in an NSF funded CCLI project focusing on the first-year CS curriculum by increasing retention and improving learning outcomes. In order to achieve the goals of the project, a series of modules providing a new programming environment is being created using open source and commercial robots platforms. These modules are being used at two additional schools (Berea College and Aliceville High School). In this paper we present the results of using these modules at the University of Alabama for the fall semester 2008 and the current spring semester 2009.

## RELATED WORK

Prior research has incorporated robots or robotic applications into their curriculum to help increase student motivation, confidence, and competence. There are a number of reports addressing the usefulness of robotics in teaching science, math and computing skills at all education levels [4]. Within the computer science area, these approaches have been used in both the introductory sequence courses and senior level courses in the undergraduate study. However, these approaches lack a few key ingredients for teaching

programming with robots. Syntax-free graphical environments have proven useful for teaching programming constructs [6]. Nevertheless, current robotics-based approaches do not mirror any of the well-known programming paradigms well enough to facilitate knowledge transfer to programming. Most graphical robotics-based programming environments focus on “wire-base” graphical interfaces that connect data sources with data consumers. In addition, studies have shown that robotics in introductory sequences requires a simulation component to facilitate initial code generation and testing [7].

The Institute for Personal Education Robotics (IPER) [2] is a consortium of university and industry partners that focuses on an integrated approach that combines software, hardware, and instructional materials. The center of the approach is the scribbler robot, an inexpensive robot that is purchased and owned by the students. Robot control programs are created using an API from traditional languages such as Python and C. Similar to the approach described in this paper are those that were developed for introductory courses. Imberman and Kilbaner [8] focused on allowing students to experiment with real-world problems. They wanted students to stay motivated and interested about the course. Most of the laboratory instructions were text based with one required robotics lab. Students programmed a LEGO® robot using a C programming language called Interactive C.

There are also studies in which simulated robots were used instead of real robots. Anderson and McLoughlin [1] presented a 3-D computer-game-like simulator to show that programming can be enjoyable and rewarding. Ladd and Harcourt [9] discuss the benefits of competitions using a homegrown robot simulator. Designed to help students learn important programming skills, both studies found that students were engaged and motivated by the active-learning environment of a simulator.

The work most closely related is that of Pearce and Nakazawa [14]. They introduce the "multiple entry point funnel" where students can take three entry level classes: web-based, robotics, and Alice. The web-based class was designed to teach students how to debug syntax errors using HTML. The robotics class used a variety of robots and a simulator called RoboLab. The Alice class focused on storytelling. The three courses were compared and the retention rate for the Alice class was the highest at 57.5%. The retention rate for the robotics class was 50% and 36.2% for the web-based class. Components such as simulation and programming paradigms are key to the success of these approaches. However none of the presented approaches combine the important elements with an interesting problem domain that attracts and captivates students.

## APPROACH

At the University of Alabama, the university has been conducting an enrollment drive for the university overall that has resulted in a 41% increase in enrollment in the College of Engineering in the last three years. The Department of Computer Science at the University of Alabama is housed in the College of Engineering. While this increase in enrollment in the College of Engineering has resulted in an increase of 24% in new computer science students, the retention of these students has not increased accordingly. Of the students who declared themselves CS majors, only 52% passed the first CS course in 2008, while 60% of all students who took the first CS course passed the course.

Approximately 50% of these students completing CS1 will enroll in the CS2 course the following semester.

Students from three different computing disciplines at the University of Alabama (Computer Science, Computer Engineering and Management Information Sciences) take a common set of introductory computing courses: CS1 and CS2. These two courses are taught in a technology enabled classroom that utilizes teaming and active learning. Associated with each of these two courses are two supplemental laboratories. the University of Alabama requires a problem solving laboratory to be taken with or before the first CS1 course. Historically, the CS1 Lab was structured so that students spent their time writing code to solve various problems. However, faculty members felt that many students in CS1 were lacking the skills to even determine how to approach many problems presented to them in CS1 and were ultimately unable to solve these problems. Hence, the CS1 Lab was restructured to emphasize problem solving as opposed to writing code. While there has been a focus to improve problem solving skills of the students, the retention rate of Computer Science students has not notably improved.

**Why Combine Alice and Robots**

With the problem of the low retention rates common in CS1 and CS2 courses throughout the country, the use of supplemental laboratories to increase student confidence and competence is an accepted solution. The question now becomes one of content and composition for these laboratories. Two possible existing approaches are to utilize Alice or robots. Because recent results have demonstrated the benefits of each of these approaches [14], we propose that an integration of these approaches may provide an even greater contribution than the individual contributions.

Alice is an introductory teaching tool that uses 3D graphics and a drag-and-drop interface. The motivation of the approach used by Alice is to provide a more positive first programming experience, with less of the frustration resulting from learning traditional computing languages. While Alice has an ease of use, the tasks that can currently be performed by Alice are limited to those that appear on a computer screen. On the other hand, robots are very engaging, with their real-world motions and sound, and robots have been shown to increase student interest and motivation. Research also indicates robots improved students understanding, as students must apply the fundamental concepts presented in CS1 and CS2. However, the effort and skill required to program robots was frustrating for some introductory students.

**PREOP**

A new tool is being developed at the University of Alabama which is extending Alice for use with robots. The tool is called PREOP (Providing Robotic Experiences Through Object-Based Programming) and was formerly called MARE. PREOP is an interactive 3D animation programming environment, that allows students to program real robots using a drag-and-drop, syntax-free interface. The selection of Alice for the modules is designed to add an algorithms-based supplement to our programming-based sequence. Although our current approach is designed to teach strong programming skills,

the focus on programming can often give a myopic view of computer science as programming only and obscure a holistic view of the algorithm development process. PREOP leverages the object-based environment of Alice to expose the students to more transferable programming skills.

PREOP currently allows students to program and control the iRobot Create wirelessly. PREOP is completely expressive, allowing students to control movement and retrieve sensor data to make decisions. The students will utilize the PREOP programming environment to model robot movement and activity in a virtual world. Students learn the functionality and operation of the iRobot Create robot through PREOP in the CS1 laboratory. Students then use the iRobot Create robot directly in the CS2 laboratory. The goal is to foster student motivation and increase student understanding of the fundamental concepts within the first-year curriculum.

The use of technology and an active-learning environment are both known techniques that help ensure student success. However, student motivation and confidence is an even more important factor in student success. A classroom environment alone is not sufficient to guarantee success. A study at Virginia Tech [13] indicates grades may not be the only factor that has an impact on student retention. The study found that the grades earned by students that successfully complete the CS introductory sequence and those that transfer out of CS are comparable. Hence, performance is not the only factor impacting student retention. Instead, confidence plays a large role in attrition. The PREOP tool is designed to increase student confidence.

Different populations of students have been found to respond to different aspects of the field. For example, female students tend to prefer an Alice CS0 course to a robotics CS0 course [14]. Robotics courses tend to have a higher level of minority students persisting in CS, which may suggest that robotics appear more to minorities [10]. The PREOP Lab is being offered as a way to engage a range of students through the use of robots and Alice. Such approaches provide a motivating and engaging learning experience that can transition easily to traditional programming languages.

## IMPACT OF PREOP

The Computer Science department at the University of Alabama has incorporated PREOP in its introductory computing curriculum beginning in the fall semester of 2008 and currently in the spring semester 2009. It is used as supplemental modules to the CS1 introduction programming course. Of the five CS1 labs offered in the fall semester of 2008, one was chosen as the test group for PREOP. The PREOP lab was chosen randomly, and no attempt was made to alter the population of students in this particular lab. There were 40 students registered for the PREOP Lab. The PREOP Lab comprised fewer females than the other non-PREOP Labs (17% vs. 25%) and the PREOP Lab also had fewer transfer students than the non-PREOP Labs (13% vs. 19%), and consequently, a lower average class level (1.46 vs. 1.78), where 1 is freshman, 2 is sophomore, etc., although the number of freshmen was similar in the PREOP Lab and the non-PREOP Labs (50% vs. 48%). 17.5% in the PREOP Lab were CS majors, while there was an average of 13% CS majors in the non-PREOP Labs.

The PREOP Lab consisted of 40 students that worked in pairs, each with a PC and an iRobot Create. Classes began with a fifteen-minute lecture on a programming concept, after which students completed a PREOP assignment illustrating the concept. Topics discussed included: objects, methods, variables, if-statements, loops and other fundamental programming concepts. Students were required to submit a lab report demonstrating their comprehension of the fundamental concepts. During the semester, the PREOP and non-PREOP Lab students were given surveys about their perceptions and performance in their first Computer Science course.

Information regarding the enrollment of students in subsequent CS2 classes was obtained from the university registrar. Students are eligible for CS2 if they have successfully completed CS1 and its corresponding PREOP or non-PREOP Lab. As illustrated in Figure 1, of those students who were eligible for the CS2 course in spring 2009 and took the PREOP Lab, 67% were registered for the CS2 course in the spring 2009 semester. Of the students who took the non-PREOP Lab sections, 53% were enrolled in the CS2 course in spring 2009. We note that although the sample size was small, of the nine CS majors initially registered in the PREOP Lab, 80% of these CS majors who completed CS1 registered for the CS2 course in spring 2009, while 63% of the non-CS majors were registered.

As mentioned earlier, the PREOP tool is designed to increase student confidence. As a measure of such confidence, students in both the PREOP and non-PREOP Labs were given a survey in the 12th week of the semester. Students were asked to rate their computer knowledge compared to their classmates as: above average, average or slightly below average. As illustrated in Figure 2, the students in the PREOP Lab rated their computer knowledge above average at 0.25 (where 0 was assigned to average, 1 to above average, and -1 to below average). The students in the non-PREOP Lab sections rated their knowledge below average at -0.03. The average rating over all students was .007. Students were also asked to rate their programming skills compared to their classmates. Students in the PREOP Lab rated their programming skills as above average at 0.5 (where 0 was assigned to average, 1 to above average and -1 to below average), while the students in the non-PREOP Lab rated theirs below average at -0.18. The average rating over all students was .145. Even more telling, 76% of students in the PREOP Lab would recommend the lab to others, while only 63% of the students in the non-PREOP Labs would recommend their non-PREOP Lab.

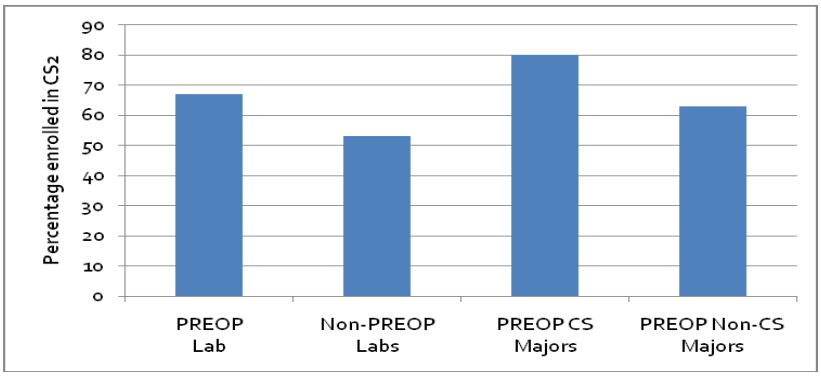


Figure 1. Retention from CS1 to CS2

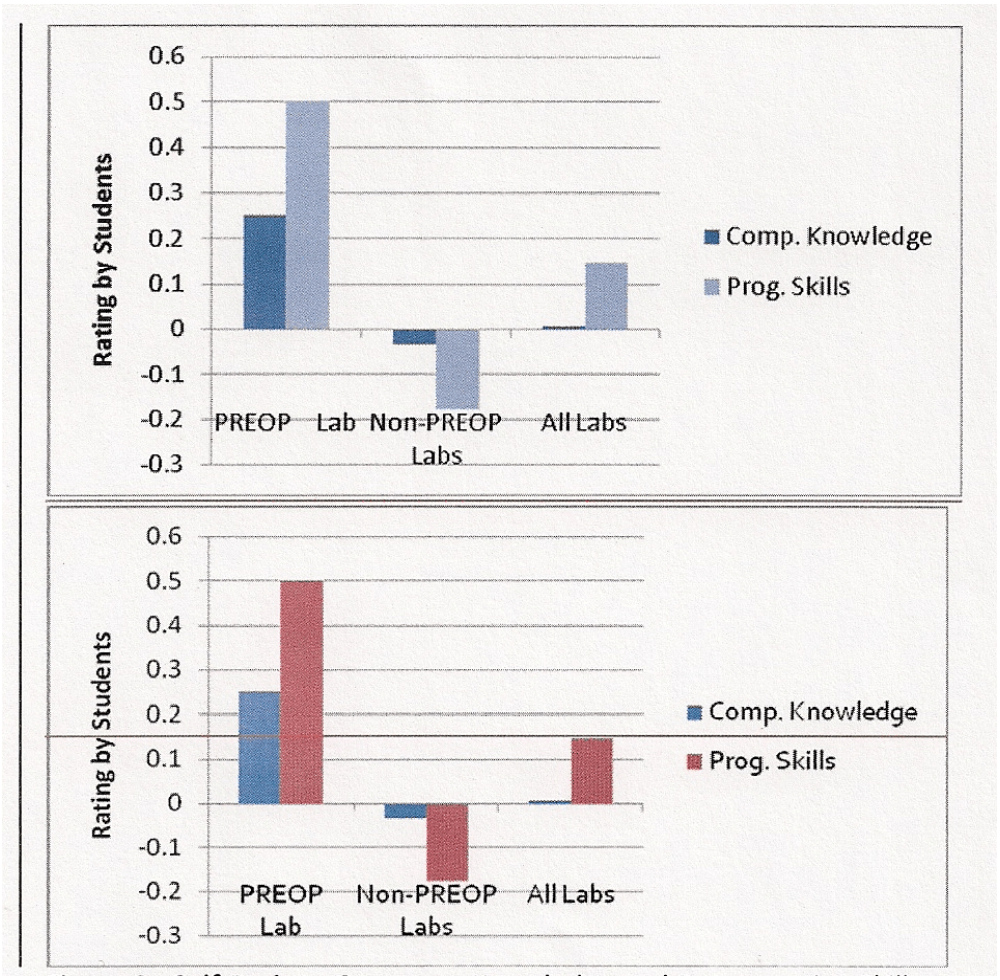


Figure 2. Self-Rating: Computer Knowledge and Programming Skills

## CONCLUSIONS

In order to increase the retention of CS students at the University of Alabama, we are developing a series of modules, called PREOP, which combine the advantages of robotics and programming with Alice. The students utilize PREOP in the first lab, which can be taken prior to or concurrently with our CS1 course. Results indicate that of the students who are eligible for CS2, 27% more of the PREOP students were registered for the CS2 course than the non-PREOP students. These results are consistent for both CS and non-CS majors. Students in the PREOP Lab also rate their programming skills and computer knowledge as 60% higher than their peers in the non-PREOP Labs. We are currently deploying PREOP for use in all lab sections associated with CS1.

## REFERENCES

[1] Anderson, E. and McLoughlin, L., "Critters in the classroom: a 3D computer-game-like tool for teaching programming to computer animation

students," *International Conference on Computer Graphics and Interactive Techniques*, 2007.

[2] Balch, T. and Summet, J., Blank, D. and Kumar, D., "Designing personal robots for education: hardware, software, and curriculum," *IEEE Pervasive Computing*, 7 (2), 2008.

[3] Bureau of Labor Statistics, U.S. Department of Labor, Occupational Outlook Handbook. 2008-09 Edition, Computer Scientists and Database Administrators, http://www.bls.gov/oco/ocos042.htm, retrieved Dec. 2008.

[4] Cannon, K., LaPoint, M., Bird, N., Panciera, K., Veeraraghavan, H., Papanikolopoulos, N. and Gini, M., "No fear: University of Minnesota robotics day camp introduces local youth to hands-on technologies," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 363-368, 2006.

[5] Career Explorer, Ten Hottest Careers: 2009, 2009, http://www.careerexplorer.net/ten-hottest-careers.asp, retrieved Mar. 27, 2009.

[6] Cooper, S. and Dann, W. and Pausch, R., "Teaching objects-first in introductory computer science," *Proceedings 34th SIGCSE technical symposium on CS Education*, pp. 191-195, 2003.

[7] Flowers, T.R. and Gossett, K.A., "Teaching problem solving, computing, and information technology with robots," *Journal of Computing Sciences in Colleges*, 17 (6), pp. 45-55, 2002.

[8] Imberman. S. and Klibaner, R., "A robotics lab for CS1," *Journal of Computing in Small Colleges*, 21 (2), pp. 131-137, 2005.

[9] Ladd, B. and Harcourt, E., "Student competitions and bots in an introductory programming course," *Journal of Computing Sciences in Colleges*, 20 (5), pp. 274-284, 2005.

[10] Layman, L., Williams,L. and Slaten, K., "Note to self: make assignments meaningful," *38th Technical Symposium on Computer Science Education*, pp. 459-463, 2007.

[11] Money Magazine, Best Careers to Have in a Recession, 2008, http://money.cnn.com/2008/04/01/pf/best_careers.moneymag/index.htm?section=money_latest, retrieved Mar. 27, 2009.

[12] Needleman, S., Doing the Math to Find the Good Jobs, 2009, http://online.wsj.com/article/SB123119236117055127.html?mod=rss_US_News, retrieved May 15, 2009.

[13] Patterson, L. and Trasti, J., "Women students in computer science: student perspectives of faculty bias as a possible influence on student retention," *Mid Atlantic Conference on the Scholarship of Diversity*, 2004.

[14] Pearce, J. and Nakazawa, M., "The funnel that grew our CIS majors in the CS desert," *Proceedings of the 39th SIGCSE technical symposium on Computer Science Education*, pp. 503-507, 2008.

[15] Wolgemuth, L., Top 10 Recessionproof Jobs, 2008, http://www.usnews.com/blogs/the-inside-job/2008/10/21/top-10-recessionproof-jobs.html, retrieved Mar. 27, 2009.

[16] Zupek, R., 25 Best Jobs for 2009, 2009, http://www.careerbuilder.com/Article/CB-1087-Job-Search-25-Best-Jobs-for-2009/, retrieved Mar. 27, 2009.

[17] Zweben, S., "Computing degree and enrollment trends," From the 2007-2008 CRA Taulbee Survey, http://cra.org, retrieved Mar. 27, 2009.