Incorporation of an online tutoring tool into programming courses

Ozgur Aktunc
Engineering Department
St. Mary’s University
oaktunc@stmarytx.edu

Abstract

One of the challenges growing institutions face is to maintain the high quality education standards with the growing number of students and class sizes. Finding sufficient number of knowledgeable tutors and supporting the cost of these tutors each semester can be financially challenging for small and medium size programs. This paper reports the efforts to incorporate an online tutoring tool into programming courses of an engineering program. The benefits of this approach are discussed from instructor’s and students’ perspectives. The objective of this ongoing research is to find efficient pedagogical approaches and tools for technology enhanced learning. As institutions of higher learning are utilizing Web-based learning at a larger scale, this study would be of educational value to higher education instructors and administrators that are interested in the utilization of online tutoring systems.

Introduction

Introductory programming courses are part of computer science and many engineering curricula. Engineering programs usually offer two to four programming courses that consists of fundamentals of programming, object-oriented programming, and programming courses geared towards the applications of the discipline, such as assembly language for computer engineering students. The programming courses help students gain problem solving skills, a crucial asset for an engineer, while they develop an algorithmic solution to a problem using the correct syntax. A review of literature published by Palumbo discussed\textsuperscript{1} the transferability of the skills learned in programming courses to problem solving skills. Another study also argued that the fundamental knowledge and study initiative acquire through introductory technology and programming courses are crucial for determining the performance of students in the subsequent advanced courses\textsuperscript{2}.

The purpose of this ongoing study is to find efficient pedagogical approaches and tools for technology enhanced learning for the programming courses in the engineering programs. The study involves finding ways to use instructional technology, including online tutoring systems, to improve the quality and effectiveness of teaching. In the early stages of this study, the author utilized an online tutoring tool, Pearson Education’s MyProgrammingLab\textsuperscript{3} for an object-oriented programming and design course using Java. This course has been taught by the author on a face-to-face basis using a very limited number of student tutors for five years. The author expects the tool to serve as a virtual tutor for students as finding sufficient number of knowledgeable student tutors and supporting the cost of these tutors have been problematic. The concluding part of this research will be to find out the success rate of the students that have been using the MyProgrammingLab. As the author has been teaching this course since 2010, he has sufficient baseline data in the traditional format of this course. The before-and-after comparisons of teaching methodologies will be helpful to redesign the course outcomes. The results will be beneficial as the longitudinal data collected can be
useful to see how well students that complete courses with online tutoring perform in subsequent courses compared with those who completed through the traditional format.

**Problem Statement**

St. Mary’s University’s enrollment has been increasing for several years. The engineering department currently is the second largest department with over 300 students with over 10 programs. One of the challenges the institution is facing is to maintain the high quality teaching while keeping the additional work that the faculty has to undertake with the growing number of students. The engineering department at St. Mary’s offers programming courses to all engineering majors that are enrolled in its seven undergraduate programs. Two of the programming courses are taken by freshman students, many of whom do not have any background in programming. One of these courses, EG 1302 Engineering Programming, focuses on the basics of programming concepts using the C language. There is an emphasis on problem solving, which is crucial for engineering students to get a grasp on. EG 1302 is taken by all engineering majors in their first semester. EG 1305 Object-Oriented Design and Programming is another programming course that focuses on the object-oriented design principles, such as abstraction and encapsulation, using the Java language. This course is taken by the computer and software engineering students in their second semester. EG 1302 is a prerequisite for EG 1305, so students that fail in EG 1302 are not able to register for EG 1305 in spring and have to retake this course.

Currently, a number of students struggle in these programming courses. The causes for this may include the fact that many of the students are not sufficiently prepared to conduct college-level work or because they did not have any computer science courses at the high school level. As a result, a significant number of students fail or withdraw from these courses. Last year, there were over 90 students accepted into the engineering programs which were split into three EG 1302 groups. Due to the number of students failing or withdrawing from the courses, another EG 1302 session had to be added to the spring semester to accommodate these students. As these students cannot register for EG 1305 in the spring semester, this forces them to take the course in their second year and potentially cause a delay in their graduation.

Teaching to a large number of students from different backgrounds is a challenging task for an instructor in any field and this is no different in the area of programming. This challenge becomes more daunting in an area where students have to focus not only on problem solving and abstract thinking, but also on the principles of programming and syntax rules. In the literature, there are reports of numerous challenges instructors face while teaching programming courses:

4. Wide variation in the students’ backgrounds.
5. Majority of the students find programming to be a difficult and complex cognitive task.
6. Excessive amount of time spent teaching the language syntax. Spending too much time learning the syntax without context is detrimental to students’ success.
7. Most programming environments are also confusing as they were developed for professional software engineers, making them difficult for first year students.
8. Inability to see the execution of the program before correcting the syntax and execution errors.
9. Lack of motivation to learn programming, as the programming profession is considered
boring by many students.

St. Mary’s University has been hiring a limited number of student tutors to help the students for EG 1302. This has been helpful for students according to their feedback. However, finding a sufficient number of knowledgeable student tutors and supporting the cost of these tutors for multiple classes requires a larger expenditure of university resources. The overall enrollment in these courses is increasing at a high rate proportional to the enrollment rate, so this issue is a high priority item for the institution.

Integration of Online Tutoring

To address the issues summarized in the previous section, the author is researching the integration of online tutoring and assessment systems to the programming courses in the engineering curricula. The tutoring tools will provide immediate feedback and direction to students and assist the instructor with typical problems, such as basic syntax errors. This will allow the instructor of the course to utilize the class time for higher level concepts, which should result in a more intellectually stimulating environment for students. Students will benefit from the immediate feedback mechanism for their assignments and have the opportunity to practice their programming skills in the virtual environment before they take their midterm tests or the final exam. Students can measure their progress with lab work and assignments without having to wait for the instructor to return their graded assignments. The assessment tool can generate detailed reports showing student progress and the success rates of students on individual exercises. This will help the instructor to spot the weaknesses of students and focus attention on the students who need the most help and the problems that are most challenging for students.

Utilization of Pearson’s MyProgrammingLab

The study consists of the following stages of application, course redesign, data collection, and analysis of the results. The author started using the MyProgrammingLab online tutoring system provided by Pearson in the EG 1305 Object-Oriented Design and Programming course in spring 2014. The students utilized this resource and completed a survey about their experiences with this tool at the end of the semester. The author began a data collection period which will be followed with the analysis of the student grades, test results, and the survey responses. A comparison will be done using the baseline data collected in the traditional settings of this course for the last four years. After the analysis of the course data and reflecting on the student surveys, the author will begin redesigning the course content. Utilization of other online tutoring systems for the languages taught in other courses will be investigated. The redesigned courses are planned to be offered starting spring 2016. The results will be compiled, analyzed, and interpreted to find out the influences of the tutoring systems on the course outcomes. The initial results after one semester use of the tool are summarized in the next section.

Initial Results

The online assignments using the tutoring system were required components of the course. Each week, students had to complete programming challenges that were assigned by the instructor and also complete the required exercises using MyProgrammingLab. The online exercises vary in
difficulty and scope, usually starting with completion of single statements to writing partial programs and complete methods. The programming challenges were completed within one week using the Eclipse IDE. These challenges require students to apply the object-oriented design and programming skills, while the online assignments allow students to practice the basic programming skills required at the low-level, such as writing a method or instantiating an object. The combination of these assessments allows the instructor see the strengths and weaknesses of students in different areas and take necessary measures, such as adjusting lecture content. At the end of the semester, students were also given a survey, where they had to assess the usefulness of the tool and report the issues they faced using the tool. The initial results were collected one section of EG 1305 where 14 students were enrolled. Students had to use the MyProgrammingLab online tutoring system besides the programming assignments given by the author. Student reception to using this tool was positive. Most of the students expressed their positive opinions to the instructor during the semester. There were a few error students faced while using the tutoring system, such as the correct response being counted as incorrect, however these errors did not cause discouragement for the students. In the surveys, 10 out of 14 students strongly agreed that the tool was beneficial to their learning. Only 1 out of 14 student was neutral about the benefits of the tool. The grade comparison to the previous year did not produce conclusive results. Even though the final grade averages were lower than the previous year, there were less number of students dropping the class, only one student compared to four. The author observed that the students were also more successful during the lab exercises in the classroom, which might be contributed by the usage of the tutoring system. A more comprehensive analysis would require a larger data set over a number of years. The data including the averages for the final exam and the in-class quizzes for the last five years of EG 1305 is illustrated in Figure 1 below. It is necessary to note that, the group of students in 2013, the year before the adoption of the tool, included some of the most successful students in the program. This may explain the decline of the averages in the first year of the adoption of the tool.

Summary and Conclusions

The success of this project will be gauged by its effect on the students. In evaluating this project, several evaluation methods will be used: The primary evaluation mechanism will be the monitoring of the D/F/W rate for the programming courses. The author expects to see a lower D/F/W rate compared to previous years due to the utilization of online tutoring systems. This has been
confirmed by the initial observation that was summarized earlier. The author will also compare the test scores of the students who utilize the online tutoring system more often to the students who do not. It is expected to see higher test scores for the students that use the online system. In the EG 1305 course, the author will also administer an initial survey that will test students’ initial knowledge and skill in procedural programming concepts as students that are taking this course have already taken an introductory course in programming with C language. This will allow the instructor to observe the student improvement in procedural concepts on a student-by-student basis to compensate for the high variability among incoming students. The overall goal is to determine the level of improvement through using the online tutoring systems independent of their programming background. Through surveys, student feedback will be collected to assess the usefulness of the tutoring system as well as the suggested improvements to it.

References


OZGUR AKTUNC
Dr. Aktunc currently serves as an Associate Professor of Software Engineering at St. Mary’s University, San Antonio, Texas. His research interests include agile software development, software metrics, dynamic and static software analysis, and usability of Web applications. Prof. Aktunc is the site director of the Prefreshman Engineering Program at St. Mary’s University.