USING ONLINE SELF-ASSESSMENT IN INTRODUCTORY PROGRAMMING CLASSES*

Gita C. Williams, Richard Bialac and Yi Liu
Information Technology and Marketing Department, Georgia College & State University, USA
(478)445-5721
gita.williams@gcsu.edu, rick.bialac@gcsu.edu, yi.liu@gcsu.edu

ABSTRACT

When the Computer Science faculty changed from covering control structures before using classes to an objects-first approach in an introductory programming course, students struggled to learn the material. The end-of-semester grades were notably lower than past years. To help students understand the concepts introduced in the new format, the faculty began requiring students to use two different online self-assessment tools to enhance their learning.

MOTIVATION

There is much debate on the best methods to use when teaching an object-oriented programming language in an introductory computer science course. An instructor typically uses one of two basic approaches: either teach students procedural constructs including loops and arrays before introducing objects or teach objects first [2].

In previous years, discussing control structures in detail before requiring students to write classes for the purpose of object instantiation was our preferred method. However, one semester, *Starting out With Java* by Tony Gaddis [5] was selected as the required textbook. This textbook introduced class design before control structures. When the objects-first approach was first implemented, the students had great difficulty understanding the concepts. Their final averages were significantly lower than past years.

Even after such a "disastrous" semester with the objects-first approach to programming, the faculty decided to continue using the same textbook the following

semester. This time two online assessment tools were chosen to supplement the text. The goal became to help the students become independent learners. Regardless of which programming approach was implemented, the students should be able to learn the overall concepts of programming. This paper will describe how online assessment can be used to promote independent learning.

INTRODUCTION

Online self-assessment tools, available for many content areas, allow students to work independently outside of class. Effective online and computer-based assessment provides both timely and informative feedback. Self-assessment encourages students to take responsibility for their own learning. In addition, assessment tools can be used to provide information to the instructor to guide in-class instruction. This research demonstrates how online assessment tools can be beneficial for computer science students and lessons learned can be applied to many areas.

There are a variety of classroom assessment tools that can supplement instruction and enhance learning [1]. A few of the assessment tools typically used in an introductory computer science course include attitude inventories, concept mapping, one-minute papers and concept tests. This study focuses on implementing interactive conceptual diagnostic tests and performance assessments. *Conceptual diagnostic tests* are multiple choice tests with questions designed to detect common misconceptions. *Performance assessments* require students to solve problems that demonstrate basic knowledge of concepts and theories. Both of these tools can be used by instructors to gauge student understanding of topics. They also can be used by students as a form of self-assessment for monitoring their own learning and to determine if their study habits are effective.

In this study, online assessment tools are introduced to supplement the regular teaching strategies for teaching an introductory programming course in computer science. The faculty continues to provide in-class lectures, closed lab exercises and weekly assignments where students developed a complete program related to the topics discussed in the class. Also, as with previous semesters, comprehensive short answer tests are given during the semester as summative assessments to assess the students understanding of topics discussed. Now, the faculty added a new requirement for the course; a student's performance on online assessments contributes to 10% of his overall grade.

CONCEPTUAL DIAGNOSTIC TESTS

Students have always been encouraged to read the chapter before attending class; however, now to ensure the students read the chapter before coming to class, they are required to complete an online terminology quiz before the lecture. The terminology quizzes are formatted as multiple choice questions containing definitions and lists of similar terms. For example, because students often confuse "relational operators" and "logical operators" both terms will be given as options for the same question (Figure 1).

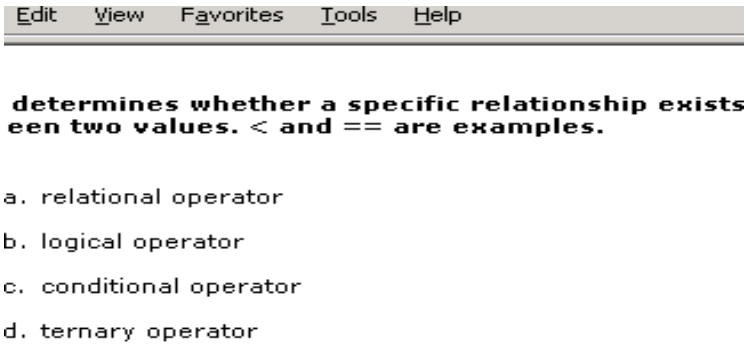

Figure 1: Sample question on terminology review quiz

Each quiz is created by randomly selecting a subset of the terminology in a chapter. These tests could be considered "benchmark tests" in that the students have to score at least 75% on each quiz in order for the quiz to count as a grade. Any grade less than 75% is recorded as a zero. Since the main purpose of these terminology quizzes is for students to become familiar with the terminology in the chapter, they are given an unlimited number of times to retake the test with only the highest score recorded. Each time a student takes a quiz, a different set of terms from the chapter could be selected. It is possible to have students tested on completely different terminology than other students. Only one question is visible at a time and students have only one opportunity to answer the question before proceeding to the next. At the end of the quiz, the student receives the total score and sees all questions asked during the quiz. The incorrectly answered questions are identified; however, the correct solutions are not indicated.

These quizzes are created in WebCT, an e-learning environment available in many higher education institutions, and are made available so that students have flexibility in time and place in which to take the each quiz. The instructor's resources for the required textbook include test questions for each chapter that could be imported into a WebCT database. For this study, the questions provided by the publisher were not used because it was decided that these questions could best be answered only after the instructor discussed the concepts in class. Instead, only definitions of terms from the book were stored in a database of terminology questions, using the faculty's knowledge of misconceptions students had in previous years.

By evaluating the results of the online WebCT terminology tests, students monitor their own learning. They are able to focus on the terms missed and reread that section of the chapter. An awareness of the important terms in the chapter is necessary to promote enhanced discussion of the concepts during the lecture. In addition, the instructor is able to tally the results of the completed quizzes, identify terms that the students have the most difficulty and address these terms in detail during the in-class lecture.

In the instructor's WebCT account, records of all of the submissions for each student are provided. If a student has taken a terminology quiz several times, the score and total time taken on each the quiz is visible. By clicking on the score of a particular quiz, the

instructor can see the questions and student responses for that quiz. Furthermore, in WebCT, the instructor has the option of viewing detailed statistics for each question such as the percentage of time the question was incorrectly answered and even the number of times each option was selected as an answer for that question.

PERFORMANCE ASSESSMENT

A common example of performance assessment in an introductory programming class is requiring students to develop a complete computer program related to the topics discussed in the class. In addition, students were required to purchase a subscription to CodeLab®, a textbook-independent, web-based tutorial developed by Turing's Craft available at http://turingscraft.com/. CodeLab® contains a database of approximately 200 introductory programming exercises. CodeLab® allows students to complete ten free CodeLab® exercises before a subscription purchase of $25 is required. Students submit short code segments and receive instant feedback. Each submission is instantly compiled, executed and evaluated as either correct or incorrect. If a submission is incorrect, helpful hints are provided.

After discussing the concepts of the chapter, the instructor assigns a set of related CodeLab® exercises for the students to complete by the next class. Each question focuses on a specific topic; this means students can concentrate on one aspect of coding instead of creating an entire program. For example, after a discussion on decision structures, the students are asked to write a boolean expression that evaluates to true if a specified variable is equal to zero (See Figure 2). For this problem, students must know that == is the relational operator for equality. Having specific questions and receiving prompt feedback encourages students to try out different possibilities.

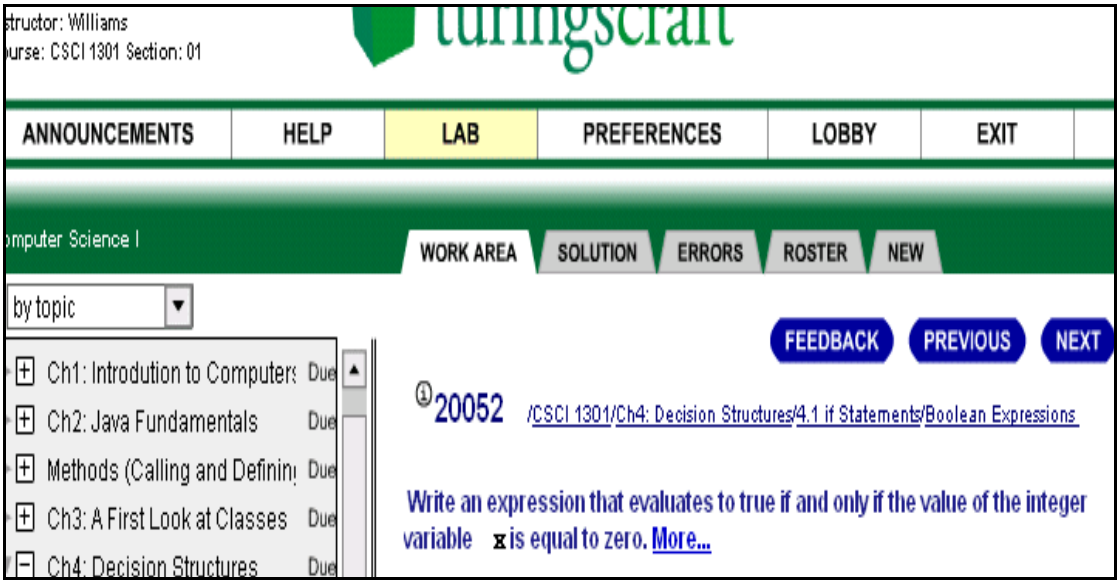

Figure 2: Sample programming exercise in CodeLab®

In the instructor's CodeLab® account, there is an online roster of students registered in the course. She determines at a glance each student's progress; however, she can only access a student's most recent submitted solution for each problem. The roster contains a color code for each student's assigned exercise. A red box indicates the student attempted to solve the problem, but the most recent submission is incorrect. A green box indicates the student's most recent submission is correct. A yellow box indicates the student's most recent submission is correct but it was submitted late. A white box indicates the student has not attempted the problem.

Unfortunately, the instructor does not know the number of submissions a student entered or how long the student worked on the problem. Also, she is able to view and print only the last submission the student entered. After the deadline, the students are able to see the solution. Therefore, in order for students to receive partial credit for an incorrect submission, they must turn in a printout of their submission at the beginning of class before the deadline.

RESULTS

When the CS faculty decided to change from the control structures first approach of teaching an object oriented language to an objects-first approach, in all of the courses taught, by several different professors, there was a marked decrease in student averages. In fact, a class average of 64 was one of the lowest that we can remember. In a class of 14 students there was only 1 A, 2 Bs, 3 Cs, 4 Ds and 4 Fs.

In the following semester, online terminology quizzes and web-based interactive coding exercises supplemented the normal teaching strategies. The results were positive. One class average increased to 86, which is slightly above the typical final class average for this course. In a class of 14 students there were 4 As, 7 Bs and 3 Ds.

When calculating the grade, the lowest grade of each student's terminology quizzes and the lowest grade of the CodeLab® exercises were dropped. Combined the terminology quizzes and CodeLab® exercises comprise only 10% of the final grade. Even though the assessments do not represent a large percentage of their final grade, most students completed them. The average terminology quiz was 91 and the average CodeLab® exercise was 94.

The online assessments are a good indicator of personal initiative as well. Even though the students are given an unlimited number of times to complete the terminology quizzes, some students chose not to retake the quiz until they earned 100. Once they reached a score that was passing, a few decided to stop and wait for the lecture. It was not surprising that two of the three students with the lowest terminology/code lab averages were the students that received the lowest grades in the class. Most students realize the advantage of repeating each quiz several times -- they would be able to view the results and use it as study guide for exams.

Because the online quizzes and exercises are not proctored, it is possible that students talk about the quizzes with their classmates and/or use their textbook to look up answers as they took the quiz. We feel this is acceptable since our goal was for the student to become more familiar with the terms. In fact, finding the answers to the terminology quizzes is a simple process. The important terminology introduced in the

chapter and related definitions are easily identified by their bold or italic formatting in the textbook.

The students in the class completed an attitude inventory regarding their opinions of the CodeLab® exercises and learning strategies. Only 10 of the 14 students completed the survey (See Figure 3). Ninety percent of the students surveyed indicate it is clear how the CodeLab® exercises fit into the course. Eighty percent of students responded that the CodeLab® exercises helped them to understand the concepts discussed in class.

		Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1	The CodeLab® exercises are easy to understand	1	3	3	3	0
2	It is clear how CodeLab® exercises fit in course	4	5	1	0	0
3	The CodeLab® exercises help me understand the concepts discussed in class	3	5	1	1	0
		1-2	3-4	5+		
4	On average how many times do you have to submit an answer in CodeLab® before it is correct?	3	3	4		
		10 minutes or less	20 minutes or less	30 minutes or less	45 minutes or less	more than 1 hour
5	On average how long do you spend on the CodeLab® assignments	1	2	2	1	4
		Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
6a	Doing programming assignments independently is the best learning strategy	3	4	1	0	2
6b	Using interactive CodeLab® exercise is the best learning strategy	2	5	2	0	1
6c	Reading the textbook is the best learning strategy	2	4	1	3	0
6d	Working with a lab partner is the best learning strategy	2	0	3	3	2
6e	Listening to lectures is the best learning strategy	3	4	2	1	0

Figure 3: CodeLab® and Learning Strategies Attitude Inventory.

Interestingly, the attitude inventory revealed a difference with the perceived level of difficulty of the exercises. The faculty would assign a collection of exercises that they thought the students should be able to solve within fifteen minutes. However, of the students surveyed, only 40% found the CodeLab® exercises easy to understand. In fact,

fifty percent of the class reported spending more than 30 minutes on the problems and 40% tried five or more times before they found the correct solution.

CONCLUSION

This paper focused on two different online self-assessment tools, CodeLab® and WebCT quizzes, to supplement instruction and enhance learning. These web-based, interactive exercises provide students with more flexibility in time and place. Students were encouraged to read the chapter before attending lecture and to practice coding exercises to gain mastery of the syntax and language constructs. Overall, online self-assessment tools proved to be effective ways to promote independent learning and ensure accountability.

These two technologies could be applied quickly because they are zero cost to the university and minimal learning curve and investment on the content development by the instructor. Several publishers provide similar supplements either through an online companion or part of a CD included with the text. Depending on the content area, the instructor may decide to use the publishers' resources.

It is important to note that with multiple choice tests you can't determine if a student has guessed the correct answer or his ability to apply the knowledge. However, WebCT quizzes can be effective in ensuring students are familiar with some of the key terminology. A variety of different assessment tools should be used together to evaluate students understanding of the concepts and encourage active learning.

REFERENCES

[1] Angelo, T.A., Cross, P.K., *Classroom Assessment Techniques* (2nd ed.). San Francisco: Jossey-Bass, 1993.

[2] Bruce, K., Controversy on how to teach CS1: a discussion on the SIGCSE-members mailing list. *In Inroads-The SIGCSE Bulletin*, December 2004.

[3] Center for the Study of Higher Education, Assessing learning five practical guides, 2002, http://www.cshe.unimelb.edu.au/assessinglearning/03/online.html, posted 2002.

[4] CodeLab® is a registered trademark of TuringsCraft at http://www.turingscraft.com/

[5] Gaddis, T., *Starting Out With Java.* El Granda, California: Scott/Jones Inc., 2004.

[6] Jacobsen, M., Kremer, R., Online testing and grading using WebCT in computer science, *World Conference on the WWW and Internet 2000,* (1), 263-268, 2000.

[7] Pope, J., Thurber, B., The computer, the discipline and the classroom: two perspectives, *World Conference on Educational Multimedia, Hypermedia and Telecommunications 2002*(1), 1956-1960, 2002.

[8] WebCT is a registered trademark of WebCT, Inc.

[9] Wright, P., A best practices approach to the use of information technology in education. *Society for Information Technology and Teacher Education International Conference 2000* (1), 2143-2150, 2000.