Examples of WeBWorK Programming Assignments[1]

Jackie Baldwin[1]
j-baldwin@cornellcollege.edu

Eileen Crupi[2]
eec91084@yahoo.com

Tabitha Estrellado[2]
tabi119@yahoo.com

Olly Gotel[2]
ogotel@pace.edu

Richard Kline[2]
rkline@pace.edu

Christelle Scharff[2]
cscharff@pace.edu

Andrew Wildenberg[1]
awildenberg@cornellcollege.edu

[1] Department of Computer Science Cornell College
Mt Vernon, IA 52314

[2] School of Computer Science and Information Systems, Pace University
New York, NY 10038

1. What is WeBWorK?

WeBWorK (available for full experimentation at http://webwork.math.rochester.edu) is a free open-source web-based assessment system developed at the University of Rochester to generate, deliver, and (automatically) grade homework problems and distribute their solutions. WeBWorK is currently used to teach mathematics (calculus, pre-calculus, algebra, applied, finite and discrete mathematics) and physics by over fifty institutions. Instructors can select and edit problems from a problem library or write their own customized WeBWorK problem sets (with their solutions, grading schemas and deadlines) in the Problem Generating (PG) macro language that mixes Perl, LaTeX, HTML, and text. PG permits the drawing of graphs and functions, and it can be extended to use JavaScript and interface with Java Applets. Problems that can be defined range from true/false and multiple-choice problems to sophisticated matching problems (evaluated via scripts). The power of PG comes from its ability to recognize and substitute formulae (e.g. WeBWorK would equally accept x+1, (x^2-1)/(x-1) or x+sin(x)^2+cos(x)^2 as an answer), and to generate individualized versions of problems using predefined pseudo-random and Perl functions. Instructors have access to statistics that provide "information on the performance of individual students and the course as a whole," and the system permits them to monitor students' work.

2. Significance and Relevance of the Topic

2.1. WeBWorK for Computer Science

We are developing WeBWorK problems in PG for teaching programming fundamentals in Java: fundamental programming constructs, algorithms and problem-solving, elementary data structures, recursion and event-driven programming. Our problems are available at: http://webwork.csis.pace.edu/webwork/. Problems are organized by topics to facilitate their distribution. We have different types of simple WeBWorK problems: true/false, multiple-choice and parameterized matching problems to test students on terminology, basic programming concepts, and the syntax and semantics of Java. These can be used to implement many traditional homework questions about programming in a system that gives students immediate feedback about their answers.

We are providing an extension to the basic WeBWorK mechanism allowing problems inside the WeBWorK framework whose answers are code fragments (or even entire classes) in Java. Student answers are tested for syntax using a standard lava compiler, and then answers are tested for semantic correctness by using a series of JUnit tests. In teaching programming fundamentals, it is common for beginning students to feel overwhelmed by the complexity of even the simplest programs. Often, much time is wasted because students are unable to hunt down minor syntax

[1] This work is supported by an NSF CCLI AI grant #0511385 entitled 'Collaborative Research: Adapting and Extending WeBWorK for Use in the Computer Science Curriculum'.

errors, especially those where the error as marked by the compiler is a long distance from the actual mistake (e.g, mismatched set of curly braces or incorrect use of the static keyword). This can result in students resenting/refusing to do work outside of supervised lab time or even giving up completely. By providing questions that hone in on program fragments, it is hoped that there will be fewer non-localized errors, and consequently less debugging frustration. Also, the unit tests will assure students they have a truly correct answer instead of one that seems to work most of the time.

We are planning to propose several extensions to the feedback capabilities of WeBWorK designed to benefit both students and faculty.

We are interested in building a community of WeBWorK instructors in the Computer Science field. The WeBWorK community in mathematics and physics is a vibrant collaboration and many institutions have contributed problem sets to a common library which has resulted in a much more useful tool than could have been provided by Rochester alone. It is hoped that we can lead a similar effort in Computer Science.

2.2. Assessment

There has been promising research on the relationship between online feedback and academic performance and the re-testing opportunities to master learning, two features offered by WeBWorK. WeBWorK is an active learning, formative assessment and students' progress monitoring platform. We will present our preliminary results of students' assessment based on the use of control groups. The main questions we are tacking are:

- Does practice with WeBWorK for programming fundamentals foster better performance (in the quizzes, midterms, finals and project work) than conventional homework?
- Does practice with WeBWorK for programming fundamentals help students write higher quality programs (e.g. commented code, fewer bugs) than conventional homework?

We are also developing a series of surveys to gather data on the student experience of using WeBWorK. We aim to gauge their perceptions with regard to its benefits (or not), and to examine (longitudinally) whether it helps students to better integrate the programming fundamentals material across all the core courses.

2.3.Other Existing Systems

CourseMarker, CodeLab and OWL (Online Web-based Learning) are proprietary web-based programming exercise systems. OWL focuses on Java; CourseMarker and CodeLab problems are based on Java, C++ and C. CourseMarker is an interesting system that grades programming assignments with lexical/typographic analysis and automated testing. However, problems may only be selected from fixed (purchased) databases – instructors may not customize problems. CodeLab and OWL deliver short exercises focusing on a particular programming construct or idea. CourseMarker, CodeLab and OWL are proprietary and focus solely on coding. With minor modifications including the extensions to the feedback capabilities of WeBWorK we are proposing, WeBWorK and its (already powerful) problem language could address a wider range of problems associated with programming fundamentals than current commercial alternatives.

3. Content of the Poster

We will prepare a poster that will:

- showcase different types of WeBWorK problems we have written to teach programming fundamentals;
- demonstrate the basic authoring environment for creating new problems, and some basic details of the system architecture; and
- present our preliminary assessment results.

We will also have a laptop available to demonstrate the complete features of WeBWorK.