

WeBWork in CS

ADAPTING AN OPEN-SOURCE WEB-BASED ASSESSMENT SYSTEM FOR THE AUTOMATED ASSESSMENT OF PROGRAMMING PROBLEMS

Dr. Christelle Scharff

Dr. Olly Gotel

Pace University, New York, USA

WeBWork in CS

Outline

- Systems for Automated Assessment of Programming Assignments
- WeBWork
- Pilot Study
- Findings and Lessons Learned from Teaching and Learning
- Conclusions and Future Work

WeBWork in CS

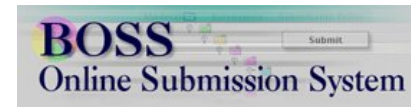
Systems for Automated Assessment of Programming Assignments

- Web-based systems
- Programming as the first skill a computer science undergraduate is expected to master
- To improve, reinforce and improve students' understanding of programming
- Types of problems
 - True / false, matching, multiple-choice, program writing
- Grading
 - Correctness + authenticity + quality

WeBWork in CS

Existing Systems

- BOSS www.dcs.warwick.ac.uk/boss



- CodeLab www.turingscraft.com



- CourseMarker www.cs.nott.ac.uk/CourseMarker



- Gradiance www.gradiance.com



- MyCodeMate www.mycodemate.com



- OWL owl.course.com



- Viope www.viope.com



WeBWork in CS

WeBWork

- *webwork.rochester.edu*
- Web-based, automated problem generation, delivery and grading system
- Free, open-source project funded by NSF
- Initial development and applications in the fields of mathematics and physics
- Currently in use at more than 50 colleges and universities

WeBWork in CS

WeBWork

- Problems are written in the Problem Generating macro language (PG)
 - Text, HTML, Latex, Perl
- Underlying engine dedicated to dealing with mathematical formulae
 - $x+1 = (x^2-1)/(x-1) = x+\sin(x)^2+\cos(x)^2$
- Individualized and parameterized versions of problems

WeBWork in CS

WeBWork for Programming Fundamentals

- Programming fundamentals [*CC2001*]
 - Fundamental programming constructs, algorithms and problem solving, elementary data structures, recursion, event-driven programming
- Extension of WeBWork for use in the core courses of the Computer Science Curriculum
- Interface WeBWork with other tools to facilitate grading of new problem types
- Demo site: atlantis.seidenberg.pace.edu/webwork2/demo
- Work funded by NSF grant

WeBWork in CS

Types of WeBWork Programming Problems

- True / false, matching and multiple choice problems for Java, Python and SML
- Sample problems designed from textbook (with permission)
 - *Java Software Solutions: Foundations of Program Design (4th Edition)*, John Lewis and William Loftus, 2004
- Evaluation of Java programs / fragments by interfacing WeBWork with JUnit [www.junit.org]

WeBWork in CS

WebWork : demo : JavaLoopTest : 1 - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

WeBWork Logged in as practice1. [Log Out](#)

WebWork => demo => JavaLoopTest => 1

▲ Prob. List

JavaLoopTest: Problem 1

(100 pts)

Question 1: Complete the code below such that:

- Ten rows of numbers are printed on two columns
- The first column starts at 2 and doubles the number on each following row
- The second column starts at 32768 and halves the number on each following row
- The tab command is used to line up the columns

The expected output can be observed on this screenshot:

```
C:\Program Files\Xinox Software\J
2      32768
4      16384
8      8192
16     4096
32     2048
64     1024
128    512
256    256
512    128
1024   64
Press any key to continue...
```

View equations as:
 plainText
 formattedText
 images
 jsMath
 asciimath

Show saved answers?
 Yes No

```
public class TwoColumns {
    public static void main(String [] args) {
        int  =  ;
        for (int firstcolumn = 2; firstcolumn <= 1024; firstcolumn*=2) {
            System.out.println( + "\t" +  );
            secondcolumn /= 2;
        }
    }
}
```

WeBWork in CS

WebWork : RUPP2007 : setJava2 : 3 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

WebWork : RUPP2007 : setJava2 ...

WeBWork

Logged in as bonyka.student.
[Log Out](#)

Main Menu

- Courses
- Homework Sets
 - setJava2
 - Problem 3
- Password/Email
- Grades
- Report bugs

Problems

- Problem 1
- Problem 2
- Problem 3
- Problem 4
- Problem 5
- Problem 6

Display Options

View equations as:

- plainText
- formattedText
- images
- jsMath
- asciimath

Show saved answers?

- Yes
- No

WebWork => RUPP2007 => setJava2 => 3

◀ Previous ▲ Prob. List ▶ Next ▶

setJava2: Problem 3

Number Class

We consider a class named *Number* with a public instance variable named *value* of type *int*. *isOdd* is a public instance method of *Number*. Write *isOdd* which takes no parameters and returns *true* if and only if the instance variable *value* is odd.

You have attempted this problem 0 times.
You have unlimited attempts remaining.

Page generated at 1:28am on Mar 13, 2007
WebWork Å© 2000-2006 [The WeBWork Project](#)

WeBWork in CS

Pilot Setting

- Use of WeBWork for homework exercises
 - Multiple attempts
- Use of WeBWork for in-class tests
 - Single attempt
 - 20 multiple choice / matching problems in one browser page
 - 20 multiple choice / matching problems with one problem per browser page
 - 30 to 60 minutes tests
 - Electronic tests only
 - Electronic tests and accompanying hardcopy
 - Tests tackled in the Eclipse IDE before being answered in WeBWork

WeBWork in CS

Pilot Setting

- Audience
 - Students new to programming
 - Fall 2005 - CS 121 / IS 223 (Programming I) – 21 students
 - Spring 2006 – CS 121 (Programming I) – 24 students
 - Python and Java
 - Students experienced in programming
 - Fall 2006 – CS 361 (Programming Languages and Implementation) – 12 students
 - SML

WeBWork in CS

Pilot Study

- Benefits (or not) of WeBWork for homework assignments and in-class tests
- Patterns of use of WeBWork (e.g. whether used, attempts number, etc.)
- Difference of use by students new to or less confident in programming with respect to students more experienced in programming
- How WeBWork affects the way instructors teach

WeBWork in CS

Lesson Learned - Student's Perspective

- Test taking/presentation
 - Consequence of no hardcopy: visibility in the browser, navigation, draft, annotations, answer elimination
 - Matching questions: text box size, multiple answers
 - Context: percentage complete, visibility of progress
- Unexpected solution
 - Variability of the ways to solve the same programming problem
- Feedback
 - Instant feedback and self-assessment
 - Single attempt: answer commitment, shock to see results at the click of a button
 - Multiple attempts: can lead to trial and error, difficulties in assessment
 - Lack of granularity and need of more constructive feedback
- In-class tests and homework exercises
 - Students preferred WeBWork for homework exercises rather than for in-class tests
 - Some kind of code-priming is desirable before going into a question / answer / feedback electronic environment

WeBWork in CS

Lessons Learned – Professor's Perspective

- Crafting WeBWork-ready problems
 - More time upfront and less after
 - Right formulation for a question
 - Importance of quality assurance
- Balance of administrative effort
 - Quite straight forward to change questions to other programming languages
 - Need of libraries of reusable, parameterized and pre-tested problems
 - Logistical tasks (e.g. creation of accounts and preparation of test / homework environments)
- Motivating students to use WeBWork
 - Freshmen and junior / senior do not approach WeBWork the same way for homework exercises
 - Only the best freshmen students or freshmen students with difficulties accessed WeBWork
 - All junior / senior students did the homework exercises
 - Junior / senior students did not proceed by answer elimination
- Impact on regular class sessions
 - In-class tests took longer than planned
 - Technical difficulties (e.g. clock synchronization)
- Randomization
 - Prevention of plagiarism
 - Difficult to go through solutions in class

WeBWork in CS

Conclusions and Future Work

- WeBWork is more favorable for homework exercises than for in-class tests
- Student maturity, familiarity with the topic, confidence and motivation are key factors driving use of WeBWork
- Evaluate the JUnit extension of WeBWork that permits to grade Java programs / fragments
- Tailor problems dynamically to support specific student needs (to challenge or support)
- Compare mechanism for student learning (e.g. references, hints and easier/harder questions)
- Need more granular and visual feedback on performance for students and professors
- Need of a freely available web-based assessment system for programming assignments
- Create a community of contributors to monitor quality and share work

WeBWork in CS

Acknowledgement

- NSF CCLI AI Grant #0511385
- *Collaborative Research: Adapting and Extending WeBWork for Use in the Computer Science Curriculum*
- Drs. Gotel, Kline, Scharff (Pace University, NY)
- Dr. Wildenberg (Cornell College, IA)