Teaching SQA by Encouraging Student Contributions to an Open Source Web-based System for the Assessment of Programming Assignments

Olly Gotel, Pace University, New York, USA
Christelle Scharff, Pace University, New York, USA
Andy Wildenberg, Cornell College, Iowa, USA

---ITiCSE2008---
WeBWorK in CS

Outline

• Pedagogical Context
• Web-based Programming Assessment Systems
• WeBWorK and its WeBWorK-JAG extension
• New Pedagogical Approach: Teaching SQA with WeBWorK
  – Motivation
  – Teaching Model
  – Practical Application
  – Results and Lessons
• Conclusions
WeBWork in CS

Pedagogical Context

• Programming is the first skill a computer science major is expected to master
• Programming fundamentals are taught in CS1 and CS2 courses
  – Fundamental programming constructs, algorithms and problem solving, elementary data structures, recursion, event-driven programming
• Open source for early exposure to collaborative and community-driven development
• Test-driven development to emphasize the criticality of formulating requirements in a testable manner and laying the foundation for quality coding
• Peer-review to give students an awareness of the value of getting an independent person to examine code and detect errors before release and use by others
WeBWorK in CS

Systems for Automated Assessment of Programming Assignments

• Web-based systems
• To encourage practice (with feedback), and improve and reinforce students’ understanding of programming concepts
• Types of questions
  – True / false, matching, multiple-choice, program writing
• Grading
  – Correctness + authenticity + quality
Some Existing Systems

- BOSS (http://www.dcs.warwick.ac.uk/boss, Open Source, Computer Science)
- CodeLab (http://www.turingscraft.com, Commercial, Computer Science)
- CourseMarker (http://www.cs.nott.ac.uk/coursemarker, Commercial, Computer Science)
- DevSquare (http://www.devsquare.com, Commercial, Computer Science)
- Educosoft (https://www.educosoft.com/ecf, Commercial, Mathematics)
- Gradiance (http://www.gradiance.com, Commercial, Computer Science)
- JavaBat (http://javabat.com, Free, Computer Science)
- MathXL (http://www.mathxl.com, Commercial, Mathematics)
- MyCodeMate (http://www.mycodemate.com, Commercial, Computer Science)
- MyMathTest (http://www.mymathtest.com, Commercial, Mathematics)
- OWL (http://owl.course.com, Commercial, Computer Science)
- Quiz PACK (http://www.sis.pitt.edu/~taler/QuizPACK.html, Open Source, Computer Science)
- Viope (http://www.viope.com, Commercial, Computer Science)
- WebAssign (http://www.webassign.net, Commercial, Mathematics)
- WebCAT (http://web-cat.cs.vt.edu, Open Source, Computer Science)
- WeBWorK (http://webwork.rochester.edu and http://atlantis.seidenberg.pace.edu/~scharff/webwork, Open Source, Mathematics and Computer Science)
- WileyPLUS (http://he-cda.wiley.com/WileyCDA/Section.rdr?id=101003, Commercial, Mathematics)
- ASSYST, CFX, CodeWitz, Marmoset, Praktomat, RoboProf, SQL-Tutor, TRAKLA2...

Not an exhaustive list!
WeBWorK: About

• http://webwork.rochester.edu
• Project funded by NSF
• Free, open-source and web-based
• Automated problem delivery and grading
• Initial development and applications in the fields of mathematics and physics
• Currently in use at more than 50 colleges and universities
WeBWorK: Underneath

- 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

• Extension of WebWork for use in programming fundamentals

• True / false, short answer and multiple choice problems for Java, Python and SML

• Evaluation of Java program fragments in real time by interfacing WebWork with JUnit [www.junit.org]
  – WebWork-JAG = WebWork + JUnit
  – Java Auto-Grader

• http://atlantis.seidenberg.pace.edu/webwork2
WebWork in CS

Example “Traditional” Question

Observe the cross in the screenshot below.

![Image of a cross](image)

(Click to enlarge)

Complete the code below to draw the cross above.

```java
public class Cross {
    public static void main(String[] args) {
        for(int i = 0; i < _; i++) {
            for(_ ; t < _ ; t++) {
                if(t == i || (9-t) == _ ) {
                    System.out.print("*");
                } else {
                    System.out.print(" ");
                }
            }
            System.out.println();
        }
    }
}
```
WeBWorK in CS

“Traditional” Questions in WeBWorK

WeBWorK is an open-source Web-based homework system designed for math and science courses. It provides a platform for instructors to create and assign homework problems, and for students to submit answers and receive feedback. The system is particularly useful for mathematics courses, where it can be used to generate a large variety of problem types. WeBWorK supports a wide range of programming languages, including Java, JavaScript, and more. This image demonstrates the use of Java and JavaScript to create dynamic problem sets, allowing for interactive and engaging learning experiences.
WebWork in CS

Simple Example “JAG” Question

JavaMethodReview: Problem 1

Boolean Operator

Write a static method named $flip$ of return type 'boolean' which will take a single boolean parameter and simply return its opposite.
WeBWorK in CS

Answer Entered but not Submitted

JavaMethodReview: Problem 1

Boolean Operator

Write a static method named 'flip' of return type 'boolean' which will take a single boolean parameter and simply return its opposite.

```java
public boolean flip(boolean thingToFlip) {
    return !thingToFlip;
}
```

You have attempted this problem 0 times.
This homework set is closed.

Email instructor
Feedback After Submission

JavaMethodReview: Problem 1
ANSWERS ONLY CHECKED -- ANSWERS NOT RECORDED

<table>
<thead>
<tr>
<th>Entered</th>
<th>Answer</th>
<th>Result</th>
<th>Messages</th>
</tr>
</thead>
</table>
| public boolean flip(boolean thingToFlip) { return thingToFlip; } |        | 60% correct | You only got 3 out of 5 tests right. You need to work on:
* Making the method static
* Making the method return the opposite of its parameter |

The answer above is NOT completely correct.

Boolean Operator

Write a static method named flip of return type boolean which will take a single boolean parameter and simply return its opposite.

```java
public boolean flip(boolean thingToFlip) {
    return !thingToFlip;
}
```
Acknowledgement of Correct Answer
Components of a Problem

• PG file to specify a problem
  – All problems in WeBWorK specified in PG
    • Code to typeset the question and compute an answer
    • Answer evaluator determines if answer matches
  – We provide a new evaluator that calls JUnit

• Template file
  – When correct answer inserted, forms valid .java file

• JUnit test file
  – Provides a series of JUnit tests to assess the response
Boolean Operator

Write a static method named 'flip' of return type 'boolean' which will take a single boolean parameter and simply return its opposite.

```java
class BoolOp {
    public void flip() {
        // your implementation here
    }
}
```

ANS(java_cmp("JavaSampleSet/BoolOp/", "BoolOp"));
General Execution Flow

- Question is displayed by WeBWorK
- User enters answer and submits
- Tmp directory is created
  - Template file with user response inserted
  - JUnit test file
- Both .java files compiled (syntax errors reported)
- JUnit tests are run
- User score is % of tests that are correct
WebWork in CS

New Pedagogical Approach: Teaching SQA with WebWork

• Software Quality Assurance - Motivation
• Writing code with:
  – Standards
  – Requirements
  – Test Cases
  – Peer Review
  – Cycles
WeBWorK in CS
Teaching Model

• Student process for contributing to WeBWorK:
  1. Formulation of problem (Requirements)
  2. Peer-review of problem formulation (Requirements Inspection and Validation -- SQA)
  3. Design of unit tests (Requirements Refinement and Testing)
  4. Peer-review of unit tests (Test Case Inspection and Verification -- SQA)
  5. Integration of problem with its test cases into the Web-based system (Deployment)
  6. Testing of problem and feedback by users (User Acceptance Testing -- SQA)
Template for Problem Formulation

- **Description.** A short description of the method to be written.
- **Method name.** The name of the method.
- **Method signature description.** A description of the method signature in terms of:
  - Modifier, i.e. either static, public, protected, private or package;
  - Type of the method, i.e. either static or instance;
  - Number and type of parameters; and
  - Return type.
- **Exceptions.** A description of the exceptions to be thrown along with the cases in which they are thrown.
- **Code.** Code provided to support writing the method.
- **Notes.** Particular restrictions concerning the method to be written.
Example – Problem Formulation

• Write a method that computes the factorial of a given number
• The method will be called `factorial` and must:
  • Be `public` and `static`
  • Take an `int` as a parameter
  • Return the factorial of that `int` as an `int`
  • Throw an `IllegalArgumentException` for a negative input or an input that would not return a `Java int`
WeBWorK in CS

Example – Expected Solution

```java
public class Factorial {
    public static int factorial(int n) {
        if (n <= 12 && n > 0) {
            return n * factorial(n - 1);
        } else if (n == 0) {
            return 1;
        }
        throw new IllegalArgumentException("Argument "+n+" not in range");
    }
}
```
import java.lang.reflect.*;
import junit.framework.*;

public class FactorialJUnitTest extends TestCase {

    private boolean existsFactorial, isStatic, returnType, paramType;

    // FactorialJUnitTest, setUp, tearDown

    public void testMethodSignature() {
        assertEquals(6,
                    Factorial.factorial(3));
    }  

    public void testFactorial3() {
        try {
            assertEquals(6,
                    Factorial.factorial(3));
        } catch (Exception e) {
            fail("Fail - n = 3");
        }
    }  

    public void testFactorial-4() {
        try {
            Factorial.factorial(-4);
            fail("Fail - n = -4");
        } catch (Exception e) {
            if (e instanceof IllegalArgumentException)
                assertTrue(true);
            else
                fail("Fail - n = -4");
        }
    }
}
WeBWorK in CS

Study Context

• 19 students, CS2 at Pace University
• Reviewed CS1 by undertaking WeBWorK assignments
• Then contributed 9 WeBWorK problems:
  – Problem formulation
  – JUnit tests (cases and feedback)
  – Peer review
  – Integrated into WeBWorK
• Examples: sum of even numbers, checking whether a number is a prime, sorting an array of integers, checking whether 2 arrays contain the same contents, etc.
WebWork in CS

Results

+  
  - Created good review problems (CS1-level questions contributed)  
  - Students had the user in mind when specifying the questions  
  - Formulating the requirements of the problem demands precision and concurrent thinking on tests  
  - Able to formulate normal cases for testing  
  - Quality and granularity of feedback provided  
  - Process and peer review improved quality of problems

-  
  - Students found it hard to come up with topics and questions  
  - User was less in mind when writing the tests and feedback  
  - Often poor initial formulation of questions (scope not clear, modifiers not stated, assumptions)  
  - Difficulty in formulating error/exception cases (and giving feedback)  
  - Large number of tests often overlapping (theme variation)  
  - Time curtails peer review; QA of problems more tricky than apparent and needs guidance too
WeBWorK in CS

Lessons

• Formulating problems with precision is programming
• Writing JUnit tests is programming
• Forced to think about requirements and testing first, and iteration / regression testing
• Learn the crucial role of SQA to catch problems - a practical introduction to an important SE topic
• Did we manage to augment our library of programming questions? Yes and No
• Students proud to contribute to an open-source system that other students have access to - promote learning from examples
• Process to adapt of our teaching model outside the WeBWorK domain - promote learning by teaching
WeBWorK in CS

Conclusions

• Development of a novel pedagogy, encouraging students to contribute their own programming questions to the WeBWorK library, introducing them to crucial practices of software engineering
• Web-based programming assessment systems open up opportunities for professors and students around the globe to transfer and share knowledge, and to extend models of teaching and learning
• Our aim to create a community of contributors to monitor quality and extend the WeBWorK library
• Well-defined processes like this that our students trialled make this possible
WeBWorK in CS

Acknowledgements

• NSF CCLI AI Grants “Collaborative Research: Adapting and Extending WeBWorK for Use in the Computer Science Curriculum” #0511385 and #0511391

• Students:
  – The many computer science students in the US, Cambodia, India, Senegal and Thailand who have used WeBWorK, especially the 19 students of CS2 at Pace University
  – Rick Kline (Pace University Professor)
  – Jacqueline Baldwin, Nathan Baur (JUnit extension)
  – Sophal Chiv (input of problems and help desk)
  – Eileen Crupi, Tabitha Estrellado (input of problems)
  – Allyson Ortiz, Veronica Portas (existing systems)
  – Yue Ma (user acceptance testing)