Providing Accurate and Timely Feedback by Automatically Grading Student Programming Labs

Pete Nordquist

Southern Oregon University

2007

Teaching programming

- Requirements must be understood in terms of input -> output mappings
- Reducing a problem solution to code means ensuring that all tests in the test suite pass
- Programming language constructs can be applied effectively only when they are understood in terms of the resulting inputs and outputs.

Feedback

- Crucial
- Many mechanisms:
  - Compiler
  - Programmer's own testing
  - End-user testing
- The sooner the better -- enter the autograder
- Autograder provides immediate feedback for end-user testing.

Teaching testing

- Often overlooked in programming classes for lack of time
- Autograder teaches testing 'by example'
  - Students test their code as best they can before running the autograder
  - Autograder introduces tests students may not have considered
  - Provides a 'covert' method for teaching good testing technique.

How does it work?

1. Students write code and test in normal environment using their own tests.
2. Students run autograder from a website
3. Autograder
   - Downloads instructor's *test harness*
   - Compiles test harness and student code
   - Runs the test harness
   - Scores results
   - Uploads results
4. Students optionally download instructor solution

Student's code

- Must supply methods with signatures that exactly match the methods specified in the requirements and called by the test harness
- Signature misspellings are easy to spot in the autograder log.
- Must write outputs to be checked to System.out and other outputs to System.err

Test Harness

- Supplied by the instructor
- Calls the methods specified in the requirements
- Captures output from the student code
- Compares captured output with test *oracle*
  - Regular expressions are very handy for doing comparisons
- Writes score for each subtest to a grade file

Let's Try it

CS257 – Computer Science II - lab 1

Local variables and if statements

http://www.sou.edu/cs/Nordquist/cs257/labs/lab1.html

Pedagogical Features

- Learn by doing
- Immediate feedback for individual subtests
- Instructor solution immediately available
- No penalty for multiple submissions – relieves anxiety
- Flexible late fee structure
- Teaches testing strategy by example
- Teaches code reading – students often read the test harness to see what it is doing

Future Work

- The autograder does not work well at all with programming assignments that require graphical user interfaces. (Try JEWL – see J. English reference in paper)
- Creating techniques to present messages that better help beginning programming students understand the source of a failure continues to need improvement.
- It would be profitable to expand the program to handle programming languages other than Java.

Student Survey

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	Response Average
1. The autograder ran test cases for the assignments that, on my own, I would not have considered running.	4% (1)	8% (2)	8% (2)	**48% (12)**	32% (8)	**3.96**
					Total Respondents	**25**
					(skipped this question)	0
2. These test cases helped me gain a better understanding of the programming concepts covered in the assignments.	**Strongly Disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**	**Response Average**
	0% (0)	8% (2)	16% (4)	**48% (12)**	28% (7)	**3.96**
					Total Respondents	**25**
					(skipped this question)	0
3. Being able to retrieve the instructor's solution to the assignment as soon as I was finished submitting my assignment was helpful to me.	**Strongly Disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**	**Response Average**
	0% (0)	0% (0)	24% (6)	28% (7)	**48% (12)**	**4.24**
					Total Respondents	**25**
					(skipped this question)	0
4. Being able to submit my assignment multiple times helped me to more thoroughly learn the programming concepts the assignment was designed to teach.	**Strongly Disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**	**Response Average**
	0% (0)	0% (0)	4% (1)	44% (11)	**52% (13)**	**4.48**
					Total Respondents	**25**
					(skipped this question)	0

SOUTHERN OREGON UNIVERSITY

Student Survey (2)

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	Response Average
5. The autograder's late fee policy allowed me to learn programming concepts more thoroughly than I might have without this policy.	0% (0)	4% (1)	32% (8)	**36% (9)**	28% (7)	**3.88**
					Total Respondents	**25**
					(skipped this question)	0
6. I found the autograder easy to use.	**Strongly Disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**	**Response Average**
	0% (0)	12% (3)	8% (2)	**44% (11)**	36% (9)	**4.04**
					Total Respondents	**25**
					(skipped this question)	0
7. If you have taken (or are taking) a programming course that did (does) not use an autograder, please answer this question: Other things being equal, given the choice between taking a programming course with or without the autograder, I would choose the course that used the autograder.	**Strongly Disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**	**Response Average**
	0% (0)	0% (0)	8% (2)	29% (7)	**63% (15)**	**4.54**
					Total Respondents	**24**
					(skipped this question)	1
8. Please make any comments you would like concerning how the autograder helped or hindered your learning.					**Total Respondents**	**20**
					(skipped this question)	5

SOUTHERN OREGON UNIVERSITY

Student Survey (3)

•It was extremely helpful to be able to run my program against the autograder repeatedly and work out all the little bugs and errors I had on my own. While I do think that one-on-one work with an instructor is still the best way to gain understanding of a concept it's not realistic to go over an assignment like that 5-10 times in the week that you have to work on it. That is the power of the autograder program.
•I think the Autograder is helpful to those that have a good understanding of programming to begin with. I don't have a good understanding of programming and the autograder was just another part I didn't understand. So while helpful I think it is limited to those that have indepth knowledge already and just confuses those that don't.
•Working with the AutoGrader was a little like working with an Analyst in a corporate environment. It is nice to have someone helping to think of test cases that will help eliminate programming bugs. I suppose it could be argued that this dependency lessens the motivation to think of all the needed test cases, but not really, it merely speeds up the learning of the kinds of test cases that tend to be common.
•I found the auto grader very useful in regards to my programming in that it displayed the test parameters used there by giving guidance and direction when debugging. I also liked the instant feed back that I received, both from a troubleshooting aspect when my code didn't work and the satisfaction of a job well done (good, good) when I got it right!
•I found the autograder to be an excellent way to get instant feedback on the lab. The only concern I have had about it is whether the lab was completed in the intended way, or in a functionally equivalent way which doesn't quite address the concepts which were meant to be learned. This concern is mostly addressed by the availability of the instructor solution.
•While the UI can be difficult without training, the ability to have instant feedback and find holes in the code make this a very good tool for students and teachers.
•I really enjoyed using this tool. It did indeed help me learn programming techniques more thoroughly.
•I would recommend adding a test case library that references all test cases and thoroughly explains their (respective) function as well as some work on the GUI aesthetics and overall usability. I suppose I would like things to be a little more intuitive (e.g. intuitive like designs by Apple).
•I thought the autograder was also very nice in that I could submit assignments at any time of day and see my grade at that moment. Also the fact that I could re-submit an assignment for a better grade was much better than the normal way of turning in an assignment once and being marked down for all the little things that i forget and am normally marked down for. I can see my mistakes, correct them, and not be marked down for them.

Related Work

- Computing pioneers N. Wirth and P. Naur wrote automated grading programs in the early 1960s. (Good company)
- Qualities described in [DOUCE, 2005] place the autograder as a third-generation automated grading system.
  - Distance enabled
  - Web based
  - Basic assessment management reporting facility