

WebTasks: Online Programming Exercises Made Easy

Guido Rößling
roessling@acm.org

Sebastian Hartte
sebastian@hartte.de

Technische Universität Darmstadt
Dept. of Computer Science
64289 Darmstadt, Germany

ABSTRACT

Providing appropriate programming tasks for beginning students of CS is challenging - and is further complicated by the need to provide meaningful feedback to solutions. We present the *WebTasks* system, a web-based platform for submitting, testing, and discussing student solutions.

Categories and Subject Descriptors

K.3.2 [Computers and Education]: Computer & Information Science Education - *Computer Science Education*

General Terms

Algorithms

Keywords

Testing, Programming tasks, Online submission, Java

1. WEBTASKS: ONLINE PROGRAMMING EXERCISE SYSTEM

Learning how to properly program in Java is a non-trivial task for many beginning students of Computer Science. Multiple problems need to be faced at the same time, especially for those with little prior programming experience. For example, there are often complaints that there are far too few “easy” programming tasks to get some hands-on experience in programming. Additionally, students often “test” their programs only by using *javac* and a few manually chosen values. The need to submit solutions on time for grading further reduces the chance for training. Especially for large classes, the amount of (helpful) feedback provided by instructors or teaching staff tends to be less than a beginner may require. Finally, inventing new tasks each year takes much time. Reusing tasks from previous years may be unwise if the solution is still available.

To face these problems, we have developed the web-based platform *WebTasks* for online exercises. The platform supports multiple choice questions, “type in the missing method body” tasks, and the upload of a Java class.

Users can log in using the CS department login or after registering with a freely chosen login/password combination. Once a programming solution is submitted, it is passed on to the Java compiler, run through *Checkstyle*, and then

tested using *JUnit*. The solution can fail at any of these stages, leading to appropriate feedback to the student and the chance to re-submit the solution. Most *Checkstyle* issues only result in a warning, not in an error, to prevent discouraging students from reaching working - if not well-written - solutions. If the solution is accepted, it is published on the platform as a valid solution. Users who have passed the same task can see the solution. They can also comment the solution to provide a layer of peer-reviewing, for example to point out alternative or more efficient solution approaches.

Students were informed about the platform in the CS 1/2 courses and invited to give it a try; however, we did not track whether a given student actually used the platform and also did not reward or punish their use or lack thereof. Since its first availability in October 2006, we now have 959 users, 948 of which are regular students, plus 11 tutors and administrators. In addition to the 893 submissions to the multiple choice questions, there have been 5,075 accepted solutions to the currently 168 programming exercises, each of which has received between 2 and 270 correct solutions. Between March 20 and April 4, 1189 rejected solutions that failed one of the test stages were recorded. There are currently 184 comments on other students’ solutions. These numbers reflect the state on April 4, 2008 and will be higher by the time the poster is in print. Similar successes are reported for related tools, such as *Trakla2* [1] (<http://trakla.cs.hut.fi>).

Compared to other systems such as *Marmoset* [2], the goal of the system is not to evaluate typical students’ mistakes or to track their progress for grading purposes, but to provide a rich and risk-free training platform for learning how to program. Therefore, we do not restrict the resubmission using tokens or time-outs: we want to encourage our students to “keep trying and refining” their sources to get them right, without artificial breaks due to time-outs.

The (German) user interface of *Webtasks* is available at <http://webtasks.informatik.tu-darmstadt.de/webtasks>.

2. REFERENCES

- [1] V. Karavirta, A. Korhonen, and L. Malmi. On the use of resubmissions in automatic assessment systems. *Computer Science Education*, 16(3), Sept. 2006.
- [2] J. Spacco, D. Hovemeyer, W. Pugh, J. Hollingsworth, N. Padua-Perez, and F. Emad. Experiences with Marmoset: Designing and Using an Advanced Submission and Testing System for Programming Courses. In *ITiCSE '06: Proceedings of the 11th annual conference on Innovation and technology in computer science education*. ACM Press, 2006.