Computer Programs Grading Computer Programs

Hyrum D. Carroll

TSYS School of Computer Science
Columbus State University

Introduction

A central activity in Computer Science courses is writing computer programs. Several such projects are generally assigned in a semester. Assessing each computer program, multiple times with different inputs, can be tedious. Furthermore, instructors often introduce a new topic and require students to computer several small programs. This allows student to focus on the new topic and incrementally increase their understanding. While the increase in computer programs assigned helps students learn, it greatly increases the amount of time to evaluate and grade. Using automated tools, such as an auto-grader system, can significantly reduce the amount of time it takes to evaluate each program. Researchers have published evaluations of auto-graders studies (Ihanto *et al.*, 2010). Their scope has been general and does not include the latest online platforms that offer free (public) "classrooms". The aim of this study is to evaluate existing auto-grader systems based on criteria relevant to the TSYS School of Computer Science.

Minimum Criteria

Initially, we set the following criteria for each auto-grader for further consideration:

1. Supports Python and Java computer languages
2. Supports custom assignments
3. Does not required advanced technical skills to install
4. Adopted at two or more sites

There are about an order of magnitude more auto-grader systems than those that fit these criteria. These criteria reflect the needs of our school. Namely, we are interested in systems that support the Python (used in our introductory Computer Science course) and Java (used in most of our other courses) computer languages. Furthermore, we filtered out systems that required advanced technical skills to install (and therefore would require significant hardware costs to support 200+ students). Finally, we required that the system be adopted at more than two sites. While many implementations exists at individual universities, their lack of widespread adoption may be an indicator of low quality or difficulty to install and set-up.

Evaluation Criteria

Seven auto-grader systems met our minimum criteria. We evaluated each system based on the following four criteria:

1. Cost
2. Time to create a new assignment
3. Grading Flexibility
4. Integration with CougarVIEW

The first and foremost criterion is financial considerations including who pays for the system. The catalyst for this study was the poor academic performance of students that avoided purchasing required material due to financial burdens. We hoped that there existed a free or significantly cheaper solution that could provide the same education opportunities. At the same time, the University of Georgia System is looking seriously at the cost of required materials and the Open Educational Resources movement is growing. Several systems offer free "classrooms" open to the public with an option to purchase a "private classroom" that requires a password. The second evaluation criterion is the amount of time required to create new assignments. A common selling point for auto-grader systems is that they will reduce the time required to mark and grade assignments. While the time required to install and/or set-up a system is important, hopefully that is a one-time cost. If a system is easy to use and allows an instructor to efficiently create new assignments, then dozens of assignments could be created for each class to benefit students. The third evaluation criterion is the flexibility in grading. Not all auto-graders provide the same options for grading. Our students have complained about the pedantic evaluation of their programs. Consequently, this criterion is broken up into three sub-criterion. It is standard practice for auto-graders to compare the output of an answer key against student submissions. Some auto-graders allow for flexible comparisons (*e.g.*, using regular expressions, a syntax for matching patterns), testing individual modules (or units) or even allowing instructor developed scripts to perform the evaluation (by controlling the inputs and evaluating the output). While the later option requires much more time, it allows for the highest level of flexibility. The final evaluation criterion is the system's ability to integrate with CougarVIEW (D2L).

Table 1. Auto-graders Evaluation Summary

Criterion		AutoGradr	CodeCheck	codio	Mimir	repl.it	stepik	zyBooks
Cost	Source			Student	Student	Department	Department	Student
	Amount	Private: Free	Public: Free	Private: $15/month	Private: $25/semester	Public: Free Private: $1/student per month	Public: Free 1 private: $49.99 10 private: $249	Private: $77 (w/textbook)
Create Assignment Time		~15 minutes	~20 minutes	~15 minutes	~15 minutes	~15 minutes	~15 minutes	~30 minutes
Grading Flexibility	Flexible Output Matching	✗	✓	Case-insensitive, whitespace & substring	Regex	Whitespace & regex	✗	Contains
	Unit Testing	✓	✓	✓	✓	✓	✓	✗
	Scriptable Evaluation	✓	✓	✓	Bash script (with sudo)	✗	Same language	✗
Integrates with CougarVIEW		✗	✗	✓	✓	✗	✗	✗

Results

Table 1 provides the details of our evaluation for each of the seven auto-grader systems based on four evaluation criteria that reflect the needs of our school. Currently (Spring 2018), we are using zyBooks (and therefore our students have access to their auto-grader system) for our introductory Computer Science course. While they offer a good product, as recipients of taxpayers dollars, we are motivated to evaluate more cost-effective solutions. It is worth noting that there exists additional implicit criteria, such as practical issues like the ability to divide students in sections. Moving forward, we will incorporate one or more of the free solutions into Fall programming classes for further evaluation.

References

P. Ihantola, T. Ahoniemi, V. Karavirta, and O. Seppälä. 2010. Review of Recent Systems for Automatic Assessment of Programming Assignments. In *Proceedings of the 10th Koli Calling International Conference on Computing Education Research*. ACM, 86–93.

Contact Information

carroll_hyrum@columbusstate.edu
Center for Commerce and Technology (CCT) 442
Columbus, Georgia, USA
(706) 507-8182