

Automated programming assessment by integrating Brightspace LMS and Gradescope for an undergraduate Python course for non-majors*

Nifty Assignment

Cengiz Günay, Sebastien Siva
Department of Information Technology
School of Science and Technology
Georgia Gwinnett College
Lawrenceville, GA 30043
`{cgunay,ssiva}@ggc.edu`

1 Introduction

Computational and algorithmic thinking is an important skill across all disciplines, which makes introductory programming courses designed for non-majors attractive. A non-majors programming course differs by covering as much material as possible without alienating its audience. Student motivation may be more important than the rigor of the programming assignments. Sustainability of these courses and their assignments depend on the automatic grading capabilities so institutions can support the courses.

Getting non-majors interested in algorithmic thinking and motivating them to finish the course required a carefully designed course philosophy. In our course, we aimed ideally to achieve the following: (1) Every student gets a win every day (some assignment so simple they can all succeed); (2) Every student gets a challenge every day; and (3) They receive continuous feedback. This strategy mandates many small assignments and quick turnaround time, thus automation.

*Copyright is held by the author/owner.

In this paper, we describe an automated grading tool that was applied to an existing introductory programming course for non-majors that showed gains in student learning and attitudes [1, 2]. The tool integrates across the learning management system (LMS) Brightspace (D2L) and to an third party grading platform, Gradescope. The integration between these two platforms is not straightforward, but it provides a way forward for institutions who already have access to them.

2 Example programming assignments

The integration is used for regular take-home assignments and in-class quiz and exam assessment sessions. An example take-home assignment may look like this:

Write a shell (text-based) program, called `travel_time.py`, that prompts the user for their driving distance and speed, and calculates (hint: division) how long a trip will take. An example of someone using the program is shown below. Note: This user chose to enter 200.0 and 50.0.

```
PS C:\Users\ssiva\Desktop> python travel_time.py
How many miles will you drive? 200.0
How fast (mph) will you drive? 50.0
It will take you 4.0 hours to get there.
PS C:\Users\ssiva\Desktop>
```

Submit the file `travel_time.py` when done.

Gradescope has limited integration with D2L for linking grades of students, but not for submissions. Therefore, students were instructed to submit the file on the Gradescope website, which gives them feedback based on instructor-prepared test cases as in screenshot (Fig. 1).

The same approach was used for quizzes and exams. Exam question was displayed in the D2L quiz and students were instructed to submit on Gradescope. We also took advantage of the Github Classroom integration that allowed files to be automatically read from students' Github repositories, which removed the need for students to install code editors and use the online Github editor instead.

3 Drawbacks

Having the students upload files created additional hurdles for providing an environment for them to first create these files. As these were non-major

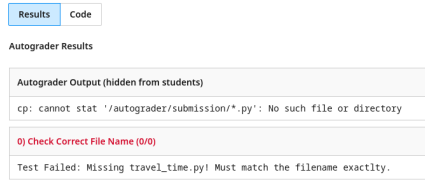


Figure 1: Gradescope feedback for student submission.

students, they had difficulty navigating the computer to create files and upload them. It could be argued that this is an essential skill that they should acquire, although it distracted from learning programming skills – especially when file management problems prevented them from submitting assignments.

Using Github Classroom and its online VS Code editor helped improve on the issues stemming from installing local programs and navigating the computer, but introduced a new problem as VS Code and Git are not beginner-friendly.

Another issue was requiring students to submit on an external website. This precluded us from locking the students into the D2L site for anti-cheating protection. Using Github Classroom exacerbated this problem as the Brightspace Lockdown Browser did not yet have support for including those URLs as of writing of this abstract.

References

- [1] Tacksoo Im, Sebastien Siva, Jason Freeman, Brian Magerko, Greg Hendler, Shelly Engelman, Morgan Miller, Brandi Villa, and Tom McKlin. Incorporating music into an introductory college level programming course for non-majors. In *2017 IEEE Integrated STEM Education Conference (ISEC)*, pages 43–48, 2017.
- [2] Sebastien Siva, Tacksoo Im, Tom McKlin, Jason Freeman, and Brian Magerko. Using music to engage students in an introductory undergraduate programming course for non-majors. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education, SIGCSE '18*, page 975–980, New York, NY, USA, 2018. Association for Computing Machinery.