A Hands-On Guide to Teaching Programming with GitHub, Travis CI, and Python

Gregory M. Kapfhammer

July 28 at PyOhio 2018
Hi! My name is GREGORY M. KAPFHAMMER

@GregKapfhammer

www.gregorykapfhammer.com
Ask and answer questions about effective instruction in the programming field.
Exploring Technologies

GitHub  Travis  Python
Python
Why would I decide to use GitHub and Travis CI?
Important Goals

Clear Status
Key Ideas
Fast Grading
HELP STUDENTS AND FACULTY EFFECTIVELY COLLABORATE
Deliverables to Check

- Source Code
- Technical Writing
- Commit Counts
- Commit Messages
- Program Output
- Data Files
USE INDUSTRY STANDARD TOOLS IN ALL COURSE WORK
DEVELOP NEW SOLUTIONS IN PYTHON WHEN NECESSARY
Benefits of Python

Tools

Options

Testing
How should I setup and use GitHub Classroom?
Create a GitHub organization for your course

Sign up your team

Create an organization account

Organization name

Allegheny-Computer-Science-111-F2018

This will be your organization name on https://github.com/Allegheny-Computer-Science-111-F2018.

Billing email

gkapfham@allegheny.edu

We'll send receipts to this inbox.

Choose your plan

Organization accounts allow your team to plan, build, review, and ship software — all while tracking bugs and discussing ideas.
Now you have an empty organization
Let GitHub Classroom access the organization

Select an organization for your new classroom

[C1] @Allegheny...
[C8] @Allegheny...
[C6] @Allegheny...
[C3] @Allegheny...
[C2] @Allegheny...

[Grant access to an organization that is not listed]

ProTip! Don't see your organization? You need to grant us access first.

with by GitHub
GitHub Classroom is open source.
Create a student roster for the Classroom

Add students

Next, let's set up a "roster", so you can easily track student progress on your dashboard.

To create your roster, you'll need to share a way to identify your students. These "identifiers" can be:

- Student last names
- Student email addresses
- Any other unique piece of information

You can always come back and complete this step later.

What information would you like to use to identify your students? (emails, student ID #, etc.)

Paste your list of identifiers below (one per line, please):
Create an assignment for your course

This organization doesn’t have any assignments yet.

Create your first assignment
CREATE TWO REPOSITORIES FOR EACH ASSIGNMENT
Using Travis CI

Solution  Starter  Check
THE STARTER REPOSITORY SHOULD NOT PASS THE TESTS
Create an assignment using the starter
A classroom will contain many assignments

Computer Science 111 Fall 2017 at Allegheny College
Allegheny-Computer-Science-111-F2017

Assignments

- **Computer Science 111 Fall 2017 Lab 1**
  - Individual assignment
  - [https://classroom.github.com/a/Uzt3ijLV](https://classroom.github.com/a/Uzt3ijLV)

- **Computer Science 111 Fall 2017 Lab 10**
  - Group assignment for Computer Science 111 Lab 10 Teams Formed on November 9, 2018
  - [https://classroom.github.com/g/G1VJvVkJG](https://classroom.github.com/g/G1VJvVkJG)

- **Computer Science 111 Fall 2017 Lab 11**
  - Group assignment for Computer Science 111 Lab 11 Teams Formed on November 16, 2017
  - [https://classroom.github.com/g/BNsbCdVC](https://classroom.github.com/g/BNsbCdVC)
How do you automatically check student submissions?
Flexible Checking

Local

Travis
Use Java and Python

1. # use both Python and Java
2. sudo: required
3. dist: trusty
4. language: python
5. python:
6. - 'Java is the main language for this course'
7. before_install:
8. - jdk_switcher use oraclejdk8
Decide if Check Passes

1. #!/bin/bash
2.
3. # assume that gatorgrader.py exits correctly
4. GATORGRADER_EXIT=0
5.
6. # assume that the human-readable answer is "No"
7. GATORGRADER_EXIT_HUMAN_PASS="No"
8.
9. # determine if the exit code is always failing
10. determine_exit_code() {
11.   if [ "$1" -eq 1 ]; then
12.     GATORGRADER_EXIT=1
13.   else
14.     if [ "$2" ]; then
15.       echo "$2 was successful"
16.     fi
17. fi
18.}
19.
20. The build passes if all of the checks pass
Running GatorGrader on a Correct Submission
Running GatorGrader on an Incorrect Project
Previous Courses and Topics

- Computational Expression
- Data Abstraction
- Web Development
- Software Engineering

Different topics, goals, languages, and levels
Exciting New Features

Script → Gradle
What do the students think about this idea?
This approach ensures that the source code and GitHub repositories are organized. It is easier for me to help the students who are struggling in an introductory course.

- SAEJIN MAHLAU-HEINERT
This tool suite made it easier for me to talk with students about technical requirements. It helped me to make complex assignments more accessible to students.

- MARIA KIM
GatorGrader encouraged me to add better code comments and try out language constructs that I would not have otherwise investigated. The tool was a big help this semester!

- SAMATHA DARRIS
GatorGrader is like having a constant coach! I liked receiving feedback on the quality of my source code and writing before turning in the final version of my lab.

- ANNA YEAGER
How can we improve and study this approach?
Ideas for Experiments

Deliverables

Insights
WHAT CODE AND CONCEPTS CAUSE STUDENT FRUSTRATION?
WHAT FEATURES BEST SUPPORT EMERGING PROGRAMMERS?
HOW DO TEAM MEMBERS INFLUENCE LEARNING?
Let's Collaborate

Pull Requests  Bug Reports
GitHub Classroom developers are great!

Internal Server Error 500 When Accessing an Assignment in GitHub Classroom #1117

Closed

gkapfham opened this issue on Aug 31, 2017 · 10 comments

Hello! Thank you so much for developing GitHub Classroom. So far, my students and I have really enjoyed using the system. In advance of running the laboratory session today, I have tested it repeatedly with small class sizes for our first laboratory assignment and everything always worked correctly.

That is, it was possible to:

- Instructor: Create an organization for the class
Review these slides on SpeakerDeck

[Image of SpeakerDeck page]

Focusing on software engineering, software testing, and data science, I program, teach, research, and write.

www.gregorykapfhammer.com
See slide source code on GitHub
Check out GatorGrader on GitHub!

Automated Grading Tool that Checks the Work of Writers and Programmers

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>internal/java</td>
<td>Renaming the DisplayOutput file to Sample instead.</td>
<td>11 months</td>
</tr>
<tr>
<td>tests</td>
<td>added test and check for empty paragraph list in counting words</td>
<td>4 months</td>
</tr>
<tr>
<td>.coverage</td>
<td>Adding the first version of the coveragec file for pytest-cov testing.</td>
<td>6 months</td>
</tr>
<tr>
<td>.gitignore</td>
<td>adding .pytest_cache to .gitignore</td>
<td>4 months</td>
</tr>
<tr>
<td>style.yapf</td>
<td>The yapf style is now using pep8.</td>
<td>a year ago</td>
</tr>
<tr>
<td>.travis.yml</td>
<td>fixing flake8 and md1 problems</td>
<td>4 months</td>
</tr>
<tr>
<td>README.md</td>
<td>fixing flake8 and md1 problems</td>
<td>4 months</td>
</tr>
<tr>
<td>gatorgrader.py</td>
<td>Added non-existent check error exit code</td>
<td>4 months</td>
</tr>
<tr>
<td>gatorgrader_comments.py</td>
<td>Fixing a docstring in the gatorgrader_comments.</td>
<td>6 months</td>
</tr>
<tr>
<td>gatorgrader_entities.py</td>
<td>One more docstring fix in the gatorgrader_entities.</td>
<td>7 months</td>
</tr>
</tbody>
</table>
Main Contributions

- Clear Status
- Key Ideas
- Fast Grading

GatorEducator/GatorGrader