Using Python, Travis CI, and GitHub to Effectively Teach Programming

Gregory M. Kapfhammer

October 5 at PyGotham 2018
Hi! My name is GREGORY M. KAPFHAMMER

@GregKapfhammer

www.gregorykapfhammer.com
Introduce strategies and tools for effectively teaching programming
USE INDUSTRY STANDARD SOFTWARE IN ALL COURSES
SUITABLE FOR INSTRUCTORS IN INDUSTRY OR ACADEMIA
Exploring Technologies

GitHub

Travis

Python

Discuss in greater detail!
Important Goals

- Clear Status
- Key Ideas
- Fast Grading
HELP STUDENTS AND TEACHERS EFFECTIVELY COLLaborate
Deliverables to Check

- Source Code
- Technical Writing
- Commit Counts
- Commit Messages
- Program Output
- Data Files
LEVERAGE EXISTING TOOLS WHENEVER POSSIBLE
Tools to Support Checks

- Source code linting
- Markdown linting
- Prose checking
- JUnit test suite
- Pytest test suite
- Automated build tools
DEVELOP NEW SOLUTIONS IN PYTHON WHEN NECESSARY
Benefits of Python

Tools

Packages

Testing
Tools for Development

- Black
- Flake8
- Pipenv
- Pylint
- Pytest
- Pytest Plugins (e.g., Codecov)
DESIGN PRINCIPLES THAT GUIDED DEVELOPMENT

Inspired by John Ousterhout's
A Philosophy of Software Design
Design Principles

- Working code is not sufficient: aim for simplicity
- Code that hasn't been executed does not work
- Simple interfaces over simple implementations
- Great documentation encourages contributions
Flexible Checking

Local

Travis

See GatorGrader and GatorGradle in the GatorEducator organization on GitHub
CREATE TWO REPOSITORIES FOR EACH ASSIGNMENT
Using Travis CI

Solution  Starter  Check
THE STARTER REPOSITORY SHOULD NOT PASS THE TESTS
Supporting different programming languages, the Gradle plugin runs GatorGrader checks in parallel.
Let's Configure Travis CI!

1. # use Java and non-root
2. dist: trusty
3. sudo: false
4. language: Java
5. jdk: oraclejdk8
6. Travis CI runs private builds for every student
7. # ignore the virtualenv that Travis creates
8. env:
9. global:
Let's Configure GatorGrader!

1. ---
2. name: cmpsc-100-fall-2018-lab3
3. break: true
4. indent: 4
5. ---

6. # -- Check the source code for various characteristics
7. # note that without an "--exact" the check is an "at least"

Configure GatorGrader for use through Gradle
GatorGrader's Output

× Repository has at least 14 commit(s)
  → Found 9 commit(s) in the Git repository

× The writing has at least 100 word(s)
  → Found 12 word(s) in a paragraph

× The output has one of the '28.75'
  → Found 0 fragment(s) in the output

Passed 6/13 (46%) of checks for f2018-lab3
Courses and Topics

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

Different topics, goals, languages, and levels
Used during laboratory, practical, and class
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.

-安娜·伊杰尔
Ideas for Experiments

Deliverables

Insights
WHAT CODE AND CONCEPTS CAUSE STUDENT FRUSTRATION?
Let's Collaborate

New Checks

Bug Reports

See GatorGrader and GatorGradle in the GatorEducator organization on GitHub