

Python module for automatically assessing homeworks

<https://github.com/macobo/python-grader>

INTRODUCTION

grader module allows lecturers to write unit tests for introductory programming courses, which can be used to assess homeworks and tests and to give fast and quality feedback to the student.

ADVANTAGES

Traditional programming exercise testing tools lack one or more of the following properties:

1. Helpful feedback
2. Extendable to support new task types
3. Independence from specific grading systems

grader module achieves these properties through:

1. The module allows test writers to check for errors and word feedback for those errors. The module also contains functions which check for common errors in different task types.
2. The module can be extended by writing custom test methods. It has in-built functions for testing both input-output based and function based tasks, as well as some less common task types (e.g. “fill in the blanks with code”).
3. Output from a test run is a JSON dictionary, which can be further processed by other programs. This has been used to integrate the module with Moodle grading system.

EXAMPLES

```
1 from grader import *
2
3 @test cases(
4     [1, 10000, 5000, 3, 7265, 8724, 2861],
5     description="Searching for number {0}"
6 )
7 def search_tester(m, users_number):
8     found = False
9     guesses = []
10
11     while len(guesses) < 14 and not found:
12         guess = int(m.stdout.new())
13         guesses.append(guess)
14
15         if guess < users_number:
16             m.stdin.put("too small")
17         elif guess > users_number:
18             m.stdin.put("too large")
19         elif guess == users_number:
20             m.stdin.put("correct")
21             found = True
22
23     assert found, (
24         "Program made too many guesses.\n" +
25         "Guesses were: {}".format(guesses))
```

Tester for program which tries to guess a number picked by user.

✓ add(2, 4) == 6

add(-3, 1) == -2

Function should return -2 but returned 3.

add(9, 0) == 9

Function returned the correct value, but changed the following global variables:
changed global variable arr which was [1] and now is [1, 2]
created global variable k = 'value'

Function must work on strings - add('another', 'string') == 'anotherstring'

Function printed out the correct result instead of returning it.
Hint: replace print with return.

Feedback for an addition fuction.

METHODS

Security against malicious programs is achieved through the use of Docker, which allows to create virtual machine-like sandboxes on demand within a fraction of a second.

Testing input-output is supported through the use of fake streams. The tester code which runs in parallel with the tested program can insert data to the input stream. As a side-effect of this design, the tester also has access to variables and functions of the tested program at the time of execution.

USAGE

The module has been successfully used in courses *Computer Programming* and *Introduction to Programming II* to help assistants grade midterms and to grade homeworks.

Future plans for the module include integration with Aivar Annamaa's programming book at programmeerimine.cs.ut.ee.

AUTHOR

Karl-Aksel Puulmann
Bachelor's curriculum of Computer Science
Institute of Computer Science
Faculty of Computer Science and Mathematics
University of Tartu

Project was created as a part of author's Bachelor's thesis, accessible via project page.