

Autograding

All projects in this course will be autograded after you submit your code. For all projects you can submit as many times as you like until the deadline. Every project's release includes its autograder for you to run yourself. This is the recommended, and fastest, way to test your code, but keep in mind you need to submit your files to have your grade registered.

To get you familiarized with the autograder, we will ask you to code, test, and submit solutions for three questions. You will submit your files on the last tab in this sequence (scroll up and click on the right most tab).

You can download all of the files associated the autograder tutorial as a zip archive (in Files on Canvas) `tutorial.zip` Save the file in your cs325 folder. Unzip this file and examine its contents:

```
[cs325]$ unzip tutorial.zip
[cs325]$ cd tutorial
[cs325/tutorial]$ ls
addition.py
autograder.py
buyLotsOfFruit.py
grading.py
projectParams.py
shop.py
shopSmart.py
testClasses.py
testParser.py
test_cases
tutorialTestClasses.py
```

This contains a number of files you'll edit or run:

- `addition.py`: source file for question 1
- `buyLotsOfFruit.py`: source file for question 2
- `shop.py`: source file for question 3
- `shopSmart.py`: source file for question 3
- `autograder.py`: autograding script (see below)

and others you can ignore:

- `test_cases`: directory contains the test cases for each question
- `grading.py`: autograder code
- `testClasses.py`: autograder code
- `tutorialTestClasses.py`: test classes for this particular project
- `projectParams.py`: project parameters

The command `python autograder.py` grades your solution to all three problems. If we run it before editing any files we get a page or two of output:

```
[cs325/tutorial]$ python autograder.py
Starting on 1-21 at 23:39:51
```

Question q1

=====

```
*** FAIL: test_cases/q1/addition1.test
*** add(a,b) must return the sum of a and b
*** student result: "0"
*** correct result: "2"
*** FAIL: test_cases/q1/addition2.test
*** add(a,b) must return the sum of a and b
*** student result: "0"
*** correct result: "5"
*** FAIL: test_cases/q1/addition3.test
*** add(a,b) must return the sum of a and b
*** student result: "0"
*** correct result: "7.9"
*** Tests failed.
```

Question q1: 0/1

Question q2

=====

```
*** FAIL: test_cases/q2/food_price1.test
*** buyLotsOfFruit must compute the correct cost of the order
*** student result: "0.0"
*** correct result: "12.25"
*** FAIL: test_cases/q2/food_price2.test
*** buyLotsOfFruit must compute the correct cost of the order
*** student result: "0.0"
*** correct result: "14.75"
*** FAIL: test_cases/q2/food_price3.test
*** buyLotsOfFruit must compute the correct cost of the order
*** student result: "0.0"
*** correct result: "6.4375"
*** Tests failed.
```

Question q2: 0/1

Question q3

=====

```
Welcome to shop1 fruit shop
Welcome to shop2 fruit shop
*** FAIL: test_cases/q3/select_shop1.test
*** shopSmart(order, shops) must select the cheapest shop
*** student result: "None"
*** correct result: "<FruitShop: shop1>"
Welcome to shop1 fruit shop
Welcome to shop2 fruit shop
*** FAIL: test_cases/q3/select_shop2.test
```

```
*** shopSmart(order, shops) must select the cheapest shop
*** student result: "None"
*** correct result: "<FruitShop: shop2>"
Welcome to shop1 fruit shop
Welcome to shop2 fruit shop
Welcome to shop3 fruit shop
*** FAIL: test_cases/q3/select_shop3.test
*** shopSmart(order, shops) must select the cheapest shop
*** student result: "None"
*** correct result: "<FruitShop: shop3>"
*** Tests failed.
```

Question q3: 0/1

Finished at 23:39:51

```
Provisional grades
=====
Question q1: 0/1
Question q2: 0/1
Question q3: 0/1
-----
Total: 0/3
```

.....

For each of the three questions, this shows the results of that question's tests, the questions grade, and a final summary at the end. Because you haven't yet solved the questions, all the tests fail. As you solve each question you may find some tests pass while other fail. When all tests pass for a question, you get full marks.

Looking at the results for question 1, you can see that it has failed three tests with the error message "add(a,b) must return the sum of a and b". The answer your code gives is always 0, but the correct answer is different. We'll fix that in the next step.