

Assignment 1

This assignment description assumes that you have watched all the week 2 videos and also done the week 2 exercise. If you read this assignment description before you've done all of that, please come back and reread it after you've completed the week 2 exercise.

Learning activities

Here are the learning activities for this assignment:

- Think about and choose tests for three separate functions.
- Write `unittest` test cases (not doctests) for each function based on the tests you chose.

The Phases of A1

A1 will be peer assessed. That means that you will be grading other people's submissions. **If you do not complete the peer assessments, you will earn at most 50% on A1.**

Here are the phases:

1. Submission Phase: write your unit tests and submit them.

You will plan and write unit tests for the three functions, as described below. You may begin working. You can create and save multiple drafts any time during the submission process, but your draft enters the system to be graded only when you click the "Enter" button. Only entered drafts will be evaluated by your peers. This means if you haven't clicked "Enter" by the deadline, your saved draft will not be evaluated.

Note: You may enter work for evaluation multiple times before the submission deadline without penalty. Only the last entry prior to the deadline will be peer-assessed.

Submission Phase Due Date: Wednesday 10 April 11:59pm Toronto time.

You can no longer enter work after this time. If you do not enter work, you will not be able to receive an evaluation for this assignment, nor will you be allowed to evaluate your classmates' work.

2. Evaluation Phase: Peer Assessment of One Another's Submissions

After the Submission Phase due date has passed, you may begin evaluating your peers' work.

You must evaluate three submissions, but you may evaluate more submissions if you wish.

You must also assess your own submission.

The evaluation phase starts Thursday 11 April. You will not be able to do your peer assessments until then.

Evaluation Phase Due Date: Wednesday 17 April 11:59pm Toronto time

You can no longer see or evaluate any work. If you do not complete the required number of evaluations, you will receive a penalty of 50%.

3. Results Phase: Getting Feedback from Your Peers

After the Evaluation Phase due date, you will be able to see the feedback from your peers.

You will not receive feedback from your peers unless you have completed three evaluations of the work of others plus the evaluation of your own.

What to do

Step 1: Download the starter code.

What to do

Step 1: Download the starter code.

In this assignment, we are providing starter code.

Here is a file containing headers and docstrings for the three functions that you will test:

- [a1.py](#)

Here are three files, each containing one `unittest` class header. You will write your `unittest` methods in these files.

- [test_num_buses.py](#)
- [test_stock_price_summary.py](#)
- [test_swap_k.py](#)

Step 2: Read and implement the three functions in

`a1.py`.

The starter code contains a header and a docstring for each function, so the descriptions are not repeated here. Each docstring contains one example call. **To gain a better understanding of each function, you should add additional examples to the docstring.**

You will **not** submit `a1.py`, but we recommend that you implement these functions.

In fact, we encourage you to post **BUGGY** implementations for the functions. (Do not post your test cases, of course.) We have created an [A1 buggy function implementations board](#) for this.

Step 3: Choose test cases.

Next, you should choose your tests for each of the three functions. As we did in the Choosing Test Cases video, write down (on paper) a table containing:

- a description of each test,
- the values of the arguments, and
- the expected result.

As you build the table, consider the various factors (size, dichotomies, boundaries, and order) to make sure that you have sufficient test coverage. Avoid duplicate tests.

You will not submit this part and it will not be marked. It's important to do this, though, in order to make sure you have a thorough set of tests.

One of the three functions mutates a list. In your tests for this function, be sure to test that the list was mutated properly.

One of the three functions mutates a list. In your tests for this function, be sure to test that the list was mutated properly.

Step 4: Write test method headers, including docstrings.

Once you have finished planning your tests on paper, write one method header for each test in the appropriate test file.

You must follow these guidelines; you will be marked on them:

- Give the test method a meaningful name that includes the name of the function and briefly describes the test case.
- Include only one of your tests in each method you write.
- Include a docstring description for the method that clearly describes the purpose of this test case. You do **not** need to include the values of the arguments in this description, but you **should** include an appropriate description of those values such as "An odd length word that begins with a vowel."

You will not be marked on grammar and spelling, but you will be marked on clarity. Please make an effort to be clear.

Step 5: Implement the test method bodies.

Now that you have chosen your tests and written the test method headers, write the method bodies to carry out the tests. Be sure to run the tests to make sure they work. Visit the [A1 buggy function implementations board](#) for example buggy functions to try your code on.

Step 6: Submit your work.

Click on the "Go to assignment" button to submit your work.

You can re-submit as many times as you like. The last submission before the deadline will be marked.