

Lab 5 – Test-Driven Development

(Fall 2024)

Lab Description:

In this lab, you will write a basic Python class following a Test-Driven Development approach. A small test suite is provided in the lab materials. Using the unit tests provided in the test file and the information contained within this document, you will create the code necessary to satisfy the tests. You will not modify the test suite itself; your goal is to write code that will satisfy it.

Class Description:

The **Potion** class represents a collection of five different potions that can be created via the combination of exactly two ingredients. These potions are named as follows:

- Elixir of Halitosis
- Philter of Amphibiosity
- Draught of Eavesdropping
- Potion of Regret
- Failed Decoction

The recipe for each potion can be inferred from the tests provided in **test_potion.py**. If the ingredients provided for the potion are invalid (i.e. they do not produce one of the first four potions listed above), then the resulting potion should always produce a Failed Decoction.

In addition to the names listed above, each potion should possess a **value** (i.e. monetary value) to be generated at random when the potion is created. The potion's random value should exist within a given range as determined by the type of potion produced. The range of values for each potion can be inferred from the tests provided.

Lab Instructions:

1. Download the `test_potion.py` file provided with the lab materials. This file contains the tests from which you will derive your code.
2. Create a new (empty) file named `potion.py` to contain the class or classes for this lab assignment.

3. Design the **Potion** class (with necessary attributes, methods, etc.) according to the specifications given in this document and in the test suite. Your **Potion** class will be written in the `potion.py` file.
4. As you are designing the **Potion** class, incrementally run the tests in `test_potion.py` to check that your code is fulfilling the requirements of the program.
5. When you have designed the **Potion** class according to the appropriate specifications and all of your tests are passing, you are ready to submit the assignment.

Notes:

If you have trouble, the following notes may help you in completing the assignment:

- The **Potion** class need not be very large. Although the wall of tests in **test_potion.py** may look intimidating, your class will only need a handful of attributes and methods in order to satisfy them. This can be accomplished in fewer than 20 lines of code, depending on your chosen implementation.
- One way to generate a random number in Python is to import the **random** module. This will give you access to several methods associated with random number generation.
- Keep in mind that the tests in **test_potion.py** are not comprehensive. They exist only to provide you with the guidelines for your class. It is possible to produce code that passes the tests but does not actually meet the specifications described in the document. Even if all your tests pass, double-check the Class Description section to ensure that your code aligns with the description provided.

Submission

Place your **potion.py** file in a .zip file named `[user_id]_lab05.zip`, where `[user_id]` is your Tech username (for example, `jstrickler_lab05.zip`). Submit the .zip file to the Lab 5 Submission dropbox in iLearn.

You do not need to resubmit **test_potion.py**, as myself and the graders will already have access to this file and it will be the same for each student. Any changes you make to your own test file will not be applied or considered in the grading process, so make sure your code satisfies the original tests provided.