Lab 4 – Test-Driven Development

(Fall 2023)

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.

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 are 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). This value should be generated at random when the potion is created, but 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 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.
- 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.
- Remember that you can use the word **in** to check whether a given value is in a list. This can be a quick way to determine whether a particular variable is one of multiple different options. For example, `x in [1, 2, 3]` would be true as long as the value of **x** was 1, 2, or 3.
- 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

Submit your **potion.py** file to the Lab 04 Submission dropbox in iLearn. You do not need to resubmit **test_potion.py**, as the graders will already have access to this file and it will be the same for each student.