**Assignment Description:**

Sometimes you will be given a program that someone else has written, and you will be asked to fix, update and enhance that program.   In this assignment you will start with an existing implementation of the classify triangle program that will be given to you.   You will also be given a starter test program that tests the classify triangle program, but those tests are not complete.

* These are the two files:  Triangle.py and TestTriangle.py
  + [***Triangle.py***](https://sit.instructure.com/courses/34748/files/5082668/download?wrap=1)is a starter implementation of the triangle classification program.
  + [***TestTriangle.py***](https://sit.instructure.com/courses/34748/files/5082660/download?wrap=1)**c**ontains a starter set of unittest test cases to test the classifyTriangle() function in the file Triangle.py file.

In order to determine if the program is correctly implemented, you will need to update the set of test cases in the test program.  You will need to update the test program until you feel that your tests adequately test all of the conditions.   Then you should run the complete set of tests against the original triangle program to see how correct the triangle program is.    Capture and then report on those results in a formal test report described below.   For this first part you should not make any changes to the classify triangle program.  You should only change the test program.

Based on the results of your initial tests, you will then update the classify triangle program to fix all defects.  Continue to run the test cases as you fix defects until all of the defects have been fixed.   Run one final execution of the test program and capture and then report on those results in a formal test report described below.

**Author:**

Erik Bornako

**Summary:**

*Summary of Test Runs:*



Reflection:

I learned the domain application of the triangle.py was accurate; it was a few typo’s, incomplete coverage, and order of execution errors that caused the initial tests to fail. The only test that worked was the “testInvalidInputA” with the sides greater than 200. Otherwise, most of the results were ‘InvalidInput’ where they were not suppose to be.

Honor Pledge:

Yes

Detailed Results:

I assumed there will be 3 inputs always into classifyTriangle function that would be of any type. I used the unittest module to automatically execute and present each test result. Source code can be found out <https://github.com/Erik-Bornako/HW02_SSW567_Triangle>

Below is the summary of each test run:

*First Test Run:*



*Second Test Run After Updates to triangle.py:*

