Assignment Description:

The objective of this assignment is for you to (a) develop a set of tests for an existing triangle classification program, (b) use those tests to find and fix defects in that program, and (c) report on your testing results for the Triangle problem

Description of assignment:

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* is a starter implementation of the triangle classification program.
    - *TestTriangle.py* contains 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.

Note that you should NOT simply replace the logic with your logic from Assignment 1. Test teams typically don't have the luxury of rewriting code from scratch and instead must fix what's delivered to the test team.

Author: Douglas Chi

Summary:

```
TestTriangle.py:33: AssertionError
=========================================== short test summary info ============================================
FAILED TestTriangle.py::TestTriangles::testEquilateralTriangles - AssertionError: 'InvalidInput' != 'Equilateral'
FAILED TestTriangle.py::TestTriangles::testFloatNumbers - AssertionError: 'InvalidInput' != 'Invalid_Input_Using_
Float'
FAILED TestTriangle.py::TestTriangles::testInputLimitMax - AssertionError: 'InvalidInput' != 'InvalidInput_Reache
d_Limit'
FAILED TestTriangle.py::TestTriangles::testInputLimitMin - AssertionError: 'InvalidInput' != 'InvalidInput_Reache
d_Limit'
FAILED TestTriangle.py::TestTriangles::testIsATriangle - AssertionError: 'InvalidInput' != 'InvalidInput_Not_A_Tr
iangle'
FAILED TestTriangle.py::TestTriangles::testIsosceles - AssertionError: 'InvalidInput' != 'isosceles'
FAILED TestTriangle.py::TestTriangles::testRightTriangleA - AssertionError: 'InvalidInput' != 'Right'
FAILED TestTriangle.py::TestTriangles::testRightTriangleB - AssertionError: 'InvalidInput' != 'Right'
FAILED TestTriangle.py::TestTriangles::testScalene - AssertionError: 'InvalidInput' != 'scalene'
============================================ 9 failed in 0.26s =============================================
```

```
PS C:\Users\Douglas Chi\Desktop\STEVENS F2023\SSW 567\Homework\Homework 2> pytest TestTriangle.py
============================================================ test session starts ============================================================
platform win32 -- Python 3.11.4, pytest-7.4.0, pluggy-1.0.0
rootdir: C:\Users\Douglas Chi\Desktop\STEVENS F2023\SSW 567\Homework\Homework 2
plugins: anyio-3.5.0
collected 9 items

TestTriangle.py .........                                                                                                          [100%]

============================================================= 9 passed in 0.04s =============================================================
PS C:\Users\Douglas Chi\Desktop\STEVENS F2023\SSW 567\Homework\Homework 2>
```

Test ID	Input	Expected Results	Actual Results	Pass or Fail
testRightTriangleA	(3,4,5)	Right	InvalidInput	Fail
testRightTriangleB	(5,3,4)	RIght	InvalidInput	Fail
testEquilateralTriangles	(1,1,1)	Equilateral	InvalidInput	Fail
testIsosceles	(3,3,4)	Isosceles	InvalidInput	Fail
testScalene	(5,7,9)	Scalene	InvalidInput	Fail
testInputLimitMax	(300,78,360)	InvalidInput	InvalidInput	Fail
testInputLimitMin	(-10,10,10)	InvalidInput	InvalidInput	Fail
TestIsATriangle	(1,10,23)	NotATriangle	InvalidInput	Fail
testFloatNumbers	(1.1,2.2,3.3)	InvalidInput	InvalidInput	Fail
Test ID	Input	Expected Results	Actual Results	Pass or Fail
testRightTriangleA	(3,4,5)	Right	Right	Pass
testRightTriangleB	(5,3,4)	RIght	RIght	Pass
testEquilateralTriangles	(1,1,1)	Equilateral	Equilateral	Pass
testIsosceles	(3,3,4)	Isosceles	Isosceles	Pass
testScalene	(5,7,9)	Scalene	Scalene	Pass
testInputLimitMax	(300,78,360)	InvalidInput	InvalidInput	Pass
testInputLimitMin	(-10,10,10)	InvalidInput	InvalidInput	Pass
TestIsATriangle	(1,10,23)	NotATriangle	NotATriangle	Pass
testFloatNumbers	(1.1,2.2,3.3)	InvalidInput	InvalidInput	Pass

	Test Run 1	Test Run 2
Tests Planned	Test: Right Triangle, Test Equilateral Triangle, Test Isosceles, Test Scalene, Test Is A Triangle	Test: Right Triangle, Test Equilateral Triangle, Test Isosceles, Test Scalene, Test Is A Triangle, Test InputLimMax, Test InputLimitMin , Test Float Numbers
Tests Executed	All	All
Tests Passed	None	All
Defects Found	b <= b, Addition of (;), Wrong function used to deteremine if inputs make a Triangle	None
Defects Fixed	All	None

Honor Pledge: I Pledge On My Honor I Have Abided By The Stevens Honor System

Detailed Results:

Being fairly non-proficient with Python this assignment poised a much harder challenge than expected as understanding GitHub and repositories took much of the initial investigation of this assignment. After establishing the basis for which updates and understanding the scope of the problem, the assignment was broken down into a series of tasks. First, a detailed understanding of what it meant for three inputs (a,b,c) to be a triangle was needed (much of this is carried over from the previous homework 1) and then a series of test cases needed to be established. Looking at the Triangle.py given in this assignment, tests were made in accordance to what was defined such as “NotATriangle”. It was seen that most of the test cases failed (if not all) due to a lack of (intentional) review of the Triangle.py code. For example, instead of b<=0, the code was written as b<=b which created a failure in the test case for Equilateral Triangle. Additionally, the function for deriving what is a Triangle did not cover the scope of what it meant to be a triangle therefore a new function (based on the previous homework 1) was derived to replace that function. After removing the semi-colon in the unnecessary spot and fixing the minute errors, the test cases that were added to the TestTriangle.py all became functional and passed the pyTest TestTriangle.py.