HW2a

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 **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 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 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: Shivank Srivastava

GitHub: HW2a

SUMMARY

Initial result

Testcase ID	Input	Expected Result	Actual result	Pass/Fail
testRightTriangleA	3,4,5	Right Triangle	Invalid Input	Fail
testRightTriangleB	5,3,4	Right Triangle	Invalid Input	Fail
testEquilateralTriangles	1,1,1	Equilateral	Invalid Input	Fail

Final result

Test Case ID	Input	Expected Result	Actual Result	Pass/Fail
testCase1	2,2,2	Equilateral	Equilateral	Pass
testCase2	15,15,15	Equilateral	Equilateral	Pass
testCase3	5,5,3	Isosceles	Isosceles	Pass
testCase4	4,6,6	Isosceles	Isosceles	Pass
testCase5	10,11,12	Scalene	Scalene	Pass
testCase6	4,2,3	Scalene	Scalene	Pass
testCase7	-1,-1,-1	InvalidInput	InvalidInput	Pass
testCase8	200,0,0	InvalidInput	InvalidInput	Pass
testCase9	5,1,1	NotATriangle	NotATriangle	Pass
testCase10	1,5,1	NotATriangle	NotATriangle	Pass
testCase11	3,4,5	Right	Right	Pass
testCase12	5,3,4	Right	Right	Pass

Matrix

	Test Run 1	Test Run 2	Test Run 3
Tests Planned	**3**	**10**	**12**
Tests Executed	**3**	**10**	**12**
Tests Passed	**0**	**10**	**12**
Defects Found	**3**	**0**	**0**
Defects Fixed	**0**	**0**	**0**

Honor pledge

“I pledge on my honor that I have not given or received any unauthorized assistance on this assignment. I further pledge that I have not copied any material from a book, article, the Internet, or any other source except where I have expressly cited the source.”