1. 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 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.

2. Author: Vittali Vamshi Krishna

Github: https://github.com/vamshivittali76/SSW-576-Fall-23/tree/main/HW%2002

3. Initial Testing Results

Test Id	input	Expected output	Actual Output	Pass or Fail
testvalidtriangle1	(3, 4, 5)	Right	Invalid Input	Fail
testvalidtriangle2	(3, 4, 5)	Equilateral	Invalid Input	Fail
testvalidtriangle3	(3, 3, 2)	Isosceles	Invalid Input	Fail
testvalidtriangle4	(2, 3, 4)	Scalene	Invalid Input	Fail
testinvalidtriangle1	(1, 2, 3)	Not a Triangle	Invalid Input	Fail

testinvalidtriangle2	(0, 1, 1)	Invalid Input	Invalid Input	Fail
testinvalidtriangle3	(-1, 1, 1)	Invalid Input	Invalid Input	Fail
testinvalidtriangle4	(200, 300, 500)	Invalid Input	Invalid Input	Fail
testnonnumericinput	("a", "b", "c")	Right		Fail
testvalidtriangle5	(5, 3, 4)	Right	Invalid Input	Fail
testvalidtriangle6	(3, 5, 4)	Right	Invalid Input	Fail
Testvalidtriangle7	(5, 12, 13)	Right	Invalid Input	Fail
Testvalidtriangle8	(8, 15, 17)	Right	Invalid Input	Fail
Testvalidtriangle9	(7, 24, 25)	Right	Invalid Input	Fail
Testvalidtriangle10	(2, 2, 3)	Isosceles	Invalid Input	Fail
Testvalidtriangle11	(3, 2, 2)	Isosceles	Invalid Input	Fail
Testvalidtriangle12	(6, 8, 10)	Right	Invalid Input	Fail
Testvalidtriangle13	(2, 2, 2)	Equilateral	Invalid Input	Fail
Testvalidtriangle14	(3, 3, 3)	Equilateral	Invalid Input	Fail
testinvalidtriangle5	(1, 1, 2)	Not a triangle	Invalid Input	Fail

4. Test Run Matrix:

	Test ID 1	Test ID 2	Test ID 3
Tests Planned	20	20	20
Tests Executed	20	20	20
Tests passed	18	19	20
Defects Found	2	1	0
Defects Fixed	0	1	2

5. Final Tests:

Test Id	input	Expected output	Actual Output	Pass or Fail
testvalidtriangle1	(3, 4, 5)	Right	Right	Pass
testvalidtriangle2	(3, 4, 5)	Equilateral	Equilateral	Pass
testvalidtriangle3	(3, 3, 2)	Isosceles	Isosceles	Pass
testvalidtriangle4	(2, 3, 4)	Scalene	Scalene	Pass

testinvalidtriangle1	(1, 2, 3)	Not a Triangle	Not a Triangle	Pass
testinvalidtriangle2	(0, 1, 1)	Invalid Input	Invalid Input	Pass
testinvalidtriangle3	(-1, 1, 1)	Invalid Input	Invalid Input	Pass
testinvalidtriangle4	(200, 300, 500)	Invalid Input	Invalid Input	Pass
testnonnumericinput	("a", "b", "c")	Right	Right	Pass
testvalidtriangle5	(5, 3, 4)	Right	Right	Pass
testvalidtriangle6	(3, 5, 4)	Right	Right	Pass
Testvalidtriangle7	(5, 12, 13)	Right	Right	Pass
Testvalidtriangle8	(8, 15, 17)	Right	Right	Pass
Testvalidtriangle9	(7, 24, 25)	Right	Right	Pass
Testvalidtriangle10	(2, 2, 3)	Isosceles	Isosceles	Pass
Testvalidtriangle11	(3, 2, 2)	Ispsceles	Ispsceles	Pass
Testvalidtriangle12	(6, 8, 10)	Right	Right	Pass
Testvalidtriangle13	(2, 2, 2)	Equilateral	Equilateral	Pass
Testvalidtriangle14	(3, 3, 3)	Equilateral	Equilateral	Pass
testinvalidtriangle5	(1, 1, 2)	Not a triangle	Not a triangle	Pass

6. While using the agile approach of Test-Driven Development in this assignment I initially faces issues writing the test cases without writing the code first. It gave me a chance to step out of my comfort zone and explore a little bit. Overall, I would say that this was a dun assignment that helped me enhance my testing skills.

7. Honor Pledge: I have not received any assistance while completing this assignment. I pledge my honor that I have abided by the stevens honor system.