Testing a legacy program and reporting on testing results

# 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 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.

# Author

Shoaib Kalawant

# Summary

# Initial Test Report

Results of running test set against the initial buggy implementation of classifyTriangle

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Input | Expected Result | Actual Result | Pass/Fail |
| T1 | (3,4,5) | Right | InvalidInput | Fail |
| T2 | (5,3,4) | Right | InvalidInput | Fail |
| T3 | (1,1,1) | Equilateral | InvalidInput | Fail |
| T4 | (1,2,3) | NotATriangle | InvalidInput | Fail |
| T5 | (5,9,14) | NotATriangle | InvalidInput | Fail |
| T6 | (5,5,8) | Isosceles | InvalidInput | Fail |
| T7 | (7,8,9) | Scalene | InvalidInput | Fail |
| T8 | (0,0,0) | InvalidInput | InvalidInput | Pass |
| T9 | (-1,-1,-1) | InvalidInput | InvalidInput | Pass |
| T10 | (201,1,1) | InvalidInput | InvalidInput | Pass |
| T11 | (1.5,2.5,3.5) | InvalidInput | InvalidInput | Pass |
|  |  |  |  |  |

# Screen Dump

# 

# Final Test Report

Results of running the test set against the improved implementation of classifyTriangle:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Input | Expected Result | Actual Result | Pass/Fail |
| T1 | (3,4,5) | Right | Right | Pass |
| T2 | (5,3,4) | Right | Right | Pass |
| T3 | (1,1,1) | Equilateral | Equilateral | Pass |
| T4 | (1,2,3) | NotATriangle | NotATriangle | Pass |
| T5 | (5,9,14) | NotATriangle | NotATriangle | Pass |
| T6 | (5,5,8) | Isosceles | Isosceles | Pass |
| T7 | (7,8,9) | Scalene | Scalene | Pass |
| T8 | (0,0,0) | InvalidInput | InvalidInput | Pass |
| T9 | (-1,-1,-1) | InvalidInput | InvalidInput | Pass |
| T10 | (201,1,1) | InvalidInput | InvalidInput | Pass |
| T11 | (1.5,2.5,3.5) | InvalidInput | InvalidInput | Pass |

# Matrix

|  |  |  |
| --- | --- | --- |
|  | Test Run 1 | Test Run 2 |
| Tests Planned | 11 | 11 |
| Tests Executed | 11 | 11 |
| Tests Passed | 4 | 11 |
| Defects Found | 7 | 0 |
| Defects Fixed | 0 | 7 |

# Github

# [Repository](https://github.com/SKele10/SSW_567)

# Honor Pledge

I pledge my honor that I have abided by the Stevens Honor System.