**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/58515/files/9197661/download?wrap=1)is a starter implementation of the triangle classification program.
  + [***TestTriangle.py***](https://sit.instructure.com/courses/58515/files/9197670/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:** Frank DiGiacomo

**Summary:** Before fixing all the bugs in the classifyTriangle function, all test cases failed due to a false invalid input. After fixing the bugs, all test cases passed except one. The one that failed was testing a 1,2,3 triangle to be Scalene. 1,2,3 returned “NotATrianlge” which is actually correct, so I had to fix that test case.

**Honor Pledge:** I pledge my honor that I have abided by the stevens honor system – Frank DiGiacomo

**Detailed Results:**

I assumed before fixing the bugs in the main file, that all my test cases would work and pass. I this case, it didn’t because I made an error in my test. This just shows that constant testing is necessary to make sure the code is 100% correct before publishing to master code base or a system.

For inputs, I just used simple integers since any other input will throw an error.

**Initial Test Report:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Input | Expected | Actual | Pass or Fail |
| testEquilateralTriangles | 1,1,1 | Equilateral | InvalidInput | Fail |
| testIsocelesTriangles | 1,2,2 | Isosceles | InvalidInput | Fail |
| testScaleneTriangles | 1,2,3 | Scalene | InvalidInput | Fail |
| testRightTriangle | 3,4,5 | Right | InvalidInput | Fail |

**Improved Test Report:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Input | Expected | Actual | Pass or Fail |
| testEquilateralTriangles | 1,1,1 | Equilateral | Equilateral | Pass |
| testIsocelesTriangles | 1,2,2 | Isosceles | Isosceles | Pass |
| testScaleneTriangles | 1,2,3 | Scalene | NotATriangle | Fail |
| testRightTriangle | 3,4,5 | Right | Right | Pass |

**Results Summary:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test Run 1 | Test Run 2 | Test Run 3 |
| Tests Planned | 3 | 17 | 17 |
| Tests Executed | 3 | 17 | 17 |
| Tests Passed | 0 | 16 | 17 |
| Defects Found | 4 | 1 | 0 |
| Defects Fixed | 4 | 1 | 0 |

Github Repo: <https://github.com/frankied003/sw567> (I put everything in main repo since I don’t want so many repos in my account. I would’ve created a different account, but since I do a lot of work outside of school, it would be tedious to keep switching back and forth between accounts)