**Test Report on Classify Triangle Program**

**1. Description:** In this assignment you start with an existing implementation of the classify triangle program given to you. You are given a starter test program that tests the classify triangle program, but these 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 classifyTriangle() function in the file Triangle.py file.

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 the 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.

**2. Author:** Nikhil Kumar G

**3. Summary:**

1. Test report table before modifying the ‘Triangle.py’ file with improved ‘TestTriangle.py’:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Input** | **Expected Results** | **Actual Result** | **Pass or Fail** |
| 1 | (1,1,1) | Equilateral | InvalidInput | Fail |
| 2 | (10,4,3) | NotATriangle | InvalidInput | Fail |
| 3 | (3,4,5) | Right | InvalidInput | Fail |
| 4 | (5,3,4) | Right | InvalidInput | Fail |
| 5 | (7,4,5) | Scalene | InvalidInput | Fail |

1. Test report table after modifying the ‘Triangle.py’ file with improved ‘TestTriangle.py’:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Input** | **Expected Results** | **Actual Result** | **Pass or Fail** |
| 1 | (1,1,1) | Equilateral | Equilateral | Pass |
| 2 | (10,4,3) | NotATriangle | NotATriangle | Pass |
| 3 | (3,4,5) | Right | Right | Pass |
| 4 | (5,3,4) | Right | Right | Pass |
| 5 | (7,4,5) | Scalene | Scalene | Pass |

**Matrix**:

|  |  |  |
| --- | --- | --- |
|  | Test Run 1 | Test Run 2 |
| Tests Planned | 5 | 5 |
| Tests Executed | 5 | 5 |
| Tests Passed | 0 | 5 |
| Defects Found | 5 | 0 |
| Defects Fixed | 5 | 0 |

**Description of the strategy:**

1. I added a couple test cases in the ‘TestTriangle.py’ file which I think would make my unit testing more vigorous.
2. Upon finding the buggy code in ‘Triangle.py’ failing all 5 tests, I started with manually picking the inputs and started inserting them in the function using the ‘Code Tracing’ method.
3. Going line by line of the code I came across a few obvious bugs which I fixed just by changing some literals.
4. There was a section in the code for ‘Right angled Triangle’ where if the inputs aren’t given in a specific order, then it would fail all the time. So, I just added some code over there to fix it.

**4. Reflection:** Personally, this assignment helped me a lot in getting flexible with GitHub code commits. It made me think about the code from a tester’s perspective where in a finished code is given to me and I must debug it line by line to pass all the test cases. I was able to debug it till all the tests passed.

**5. 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 where I have expressly cited the source.