HW 02A - Testing a legacy program and reporting on testing results

Name – Jashil Mehta

1. **Assignment Description**: The objective of this assignment is for you to (a) develop a set of tests for an existing triangle classification program, (b) use those tests to find and fix defects in that program, and (c) report on your testing results for the Triangle problem

2. **Author**: Jashil Mehta

3. **Summary**: The

4. **Honor** – I pledge my honor that I have abided my work according to Stevens Policy

**Test Report 1 :-**

This test case includes many bugs counting upto 8 of them.

/usr/local/bin/python3 "/Users/jashilmehta/Desktop/CODES/Software Testing /TestTriangle.py"

jashilmehta@Jashils-MacBook-Air Desktop % /usr/local/bin/python3 "/Users/jashilmehta/Desktop/CODES/Software Testing /TestTriangle.py"

Running unit tests

FF.FFFFF.F

======================================================================

FAIL: testEquilateralTriangleA (\_\_main\_\_.TestTriangles)

----------------------------------------------------------------------

Traceback (most recent call last):

File "/Users/jashilmehta/Desktop/CODES/Software Testing /TestTriangle.py", line 21, in testEquilateralTriangleA

self.assertEqual(classifyTriangle(1,23,1),'Equilateral')

AssertionError: 'NotATriangle' != 'Equilateral'

- NotATriangle

+ Equilateral

======================================================================

FAIL: testEquilateralTriangleB (\_\_main\_\_.TestTriangles)

----------------------------------------------------------------------

Traceback (most recent call last):

File "/Users/jashilmehta/Desktop/CODES/Software Testing /TestTriangle.py", line 24, in testEquilateralTriangleB

self.assertEqual(classifyTriangle(10,23,10),'Equilateral')

AssertionError: 'NotATriangle' != 'Equilateral'

- NotATriangle

+ Equilateral

======================================================================

FAIL: testIsocelesTriangleB (\_\_main\_\_.TestTriangles)

----------------------------------------------------------------------

Traceback (most recent call last):

File "/Users/jashilmehta/Desktop/CODES/Software Testing /TestTriangle.py", line 30, in testIsocelesTriangleB

self.assertEqual(classifyTriangle(4, 5, 6),'Isoceles')

AssertionError: 'Scalene' != 'Isoceles'

- Scalene

+ Isoceles

======================================================================

FAIL: testNotATriangleB (\_\_main\_\_.TestTriangles)

----------------------------------------------------------------------

Traceback (most recent call last):

File "/Users/jashilmehta/Desktop/CODES/Software Testing /TestTriangle.py", line 33, in testNotATriangleB

self.assertEqual(classifyTriangle(1, -5, 1),'NotATriangle')

AssertionError: 'InvalidInput' != 'NotATriangle'

- InvalidInput

+ NotATriangle

======================================================================

FAIL: testNotATriangleC (\_\_main\_\_.TestTriangles)

----------------------------------------------------------------------

Traceback (most recent call last):

File "/Users/jashilmehta/Desktop/CODES/Software Testing /TestTriangle.py", line 36, in testNotATriangleC

self.assertEqual(classifyTriangle(1,-2, -5),'NotATriangle')

AssertionError: 'InvalidInput' != 'NotATriangle'

- InvalidInput

+ NotATriangle

======================================================================

FAIL: testRightTriangleA (\_\_main\_\_.TestTriangles)

----------------------------------------------------------------------

Traceback (most recent call last):

File "/Users/jashilmehta/Desktop/CODES/Software Testing /TestTriangle.py", line 9, in testRightTriangleA

self.assertEqual(classifyTriangle(3,9,1),'Right')

AssertionError: 'NotATriangle' != 'Right'

- NotATriangle

+ Right

======================================================================

FAIL: testRightTriangleB (\_\_main\_\_.TestTriangles)

----------------------------------------------------------------------

Traceback (most recent call last):

File "/Users/jashilmehta/Desktop/CODES/Software Testing /TestTriangle.py", line 12, in testRightTriangleB

self.assertEqual(classifyTriangle(5,3,9),'Right')

AssertionError: 'NotATriangle' != 'Right'

- NotATriangle

+ Right

======================================================================

FAIL: testRightTriangleD (\_\_main\_\_.TestTriangles)

----------------------------------------------------------------------

Traceback (most recent call last):

File "/Users/jashilmehta/Desktop/CODES/Software Testing /TestTriangle.py", line 18, in testRightTriangleD

self.assertEqual(classifyTriangle(0, 6, 10), 'Right')

AssertionError: 'InvalidInput' != 'Right'

- InvalidInput

+ Right

----------------------------------------------------------------------

Ran 10 tests in 0.001s

FAILED (failures=8)

jashilmehta@Jashils-MacBook-Air Desktop %

**Test Report 2 :-**

This test successfully gave no bugs after fixing the code and test cases.

jashilmehta@Jashils-MacBook-Air Desktop % /usr/local/bin/python3 "/Users/jashilmehta/Desktop/CODES/Software Testing /TestTriangle.py"

Running unit tests

..........

----------------------------------------------------------------------

Ran 10 tests in 0.000s

OK

jashilmehta@Jashils-MacBook-Air Desktop %

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Input | Expected Result | Actual Result | Pass/Fail |
| test\_RightTriangleA | (3,4,5) | Right Angled | Right Angled | Pass |
| test\_RightTriangleB | (5,12,13) | Right Angled | Right Angled | Pass |
| test\_RightTriangleC | (3,4,5) | Right Angled | Right Angled | Pass |
| test\_RightTriangleD | (8,15,17) | Right Angled | Right Angled | Pass |
| test\_EquilateralTriangleA | (1,1,1) | Equilateral | Equilateral | Pass |
| test\_EquilateralTriangleA | (10,10,10) | Equilateral | Equilateral | Pass |
| test\_IsocelesTriangleA | (3,3,2) | Iscocles | Iscocles | Pass |
| test\_IsocelesTriangleB | (4,5,5) | Iscocles | Iscocles | Pass |
| test\_NotATriangleA | (1,5,1) | Not a Triangle | Not a Triangle | Pass |
| test\_NotATriangleB | (1,1,5) | Not a Triangle | Not a Triangle | Pass |

**Final Test Summary**

|  |  |  |
| --- | --- | --- |
|  | TestRun 1 | TestRun2 |
| Test Planned | 10 | 10 |
| Test Executed | 10 | 10 |
| Test Passed | 2 | 10 |
| Defects Found | 8 | 0 |
| Defects Fixed | 0 | 0 |

After the first test run it gave 8 defects and only two of them were passed out of 10

The test run 2 gave a successful result of all the test passed i.e., 10.