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

By Vaibhav Vashisht

1.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. 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.Name/Author : Vaibhav Vashisht
Github :: https://github.com/vaibhavvashisht16/Triangle567

3. Initial Testing

testsInvalidInput1	-2,1,4	InvalidInput	InvalidInput	Pass
testsInvalidInput2	0,0,0	InvalidInput	InvalidInput	Pass
testsInvalidInput3	200,400,300	Valid	InvalidInput	Fail
testNotaTriangle1	1,10,12	NotATriangle	InvalidInput	Fail
testNotaTriangle2	3,1.5,1.5	NotATriangle	InvalidInput	Fail
testNotaTriangle3	1,1,2	NotATriangle	InvalidInput	Fail
testNotaTriangle4	4,2,6	NotATriangle	InvalidInput	Fail
testRightTriangleA1	3,4,5	Right	InvalidInput	Fail
testRightTriangleA2	12,13,5	Right	InvalidInput	Fail
testRightTriangleA3	10,6,8	Right	InvalidInput	Fail
testRightTriangleA4	4,3,5	Right	InvalidInput	Fail
testRightTriangleA5	12,13,5	Right	InvalidInput	Fail
testEquilateralTriangles1	1,1,1	Equilateral	InvalidInput	Fail
testEquilateralTriangles2	6,6,6	Equilateral	InvalidInput	Fail
testEquilateralTriangles3	7,7,7	Equilateral	InvalidInput	Fail
testEquilateralTriangles4	1.5,1.5,1.5	Equilateral	InvalidInput	Fail

testIsoscelesTriangle1	3,3,5	Isosceles	InvalidInput	Fail
testIsoscelesTriangle2	4,2,2	Isosceles	InvalidInput	Fail
testIsoscelesTriangle3	5,9,5	Isosceles	InvalidInput	Fail
testIsoscelesTriangle4	5,5,3	Isosceles	InvalidInput	Fail
testscaleneTriangle1	1,3,2	Scalene	InvalidInput	Fail
testscaleneTriangle2	3,4,6	Scalene	InvalidInput	Fail
testscaleneTriangle3	4,2,6	Scalene	InvalidInput	Fail
testscaleneTriangle4	3,4,7	Scalene	InvalidInput	Fail

4. Run Matrix :

	Test 1	Test 2	Test 3	Test 4	Test 5
Test Planned	24	24	24	24	24
Test Executed	23	24	24	24	24
Tests passed	2	12	15	18	24
Defects found	2	3	3	2	0
Defects fixed	2	3	3	2	0

5. After making correction to the code

testNotaTriangle1	1,10,12	InvalidInput	InvalidInput	Pass
testNotaTriangle2	3,1.5,1.5	InvalidInput	InvalidInput	Pass
testNotaTriangle3	1,1,2	NotATriangle	NotATriangle	Pass
testNotaTriangle4	4,2,6	NotATriangle	NotATriangle	Pass
testRightTriangleA1	3,4,5	Right	Right	Pass
testRightTriangleA2	12,13,5	Right	Right	Pass
testRightTriangleA3	10,6,8	Right	Right	Pass
testRightTriangleA4	4,3,5	Right	Right	Pass
testRightTriangleA5	12,13,5	Right	Right	Pass
testEquilateralTriangles1	1,1,1	Equilateral	Equilateral	Pass
testEquilateralTriangles2	6,6,6	Equilateral	Equilateral	Pass
testEquilateralTriangles3	7,7,7	Equilateral	Equilateral	Pass
testEquilateralTriangles4	1.5,1.5,1.5	Equilateral	Equilateral	Pass
testIsoscelesTriangle1	3,3,5	Isosceles	Isosceles	Pass

testIsoscelesTriangle2	4,2,2	Isosceles	Isosceles	Pass
testIsoscelesTriangle3	5,9,5	Isosceles	Isosceles	Pass
testIsoscelesTriangle4	5,5,3	Isosceles	Isosceles	Pass
testscaleneTriangle1	1,3,2	Isosceles	Isosceles	Pass
testscaleneTriangle2	3,4,6	Isosceles	Isosceles	Pass
testscaleneTriangle3	4,2,6	Scalene	Scalene	Pass
testscaleneTriangle4	3,4,7	Scalene	Scalene	Pass

Test-driven troubleshooting is an exceptionally powerful method of fixing incorrect code. As I fixed deformities in the code and ran the tests, more imperfections became evident. Notwithstanding, composing tests as you compose code would be a more viable method of mistake checking than composing all the code, then, at that point every one of the tests as I would like to think.

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