Shantanu(Shaan) T K SSW-567 Steven's Institute of Technology
CWID: 20012027

HW02a
Testing a Legacy program and Reporting on testing Results

1. **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.

   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.

   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.

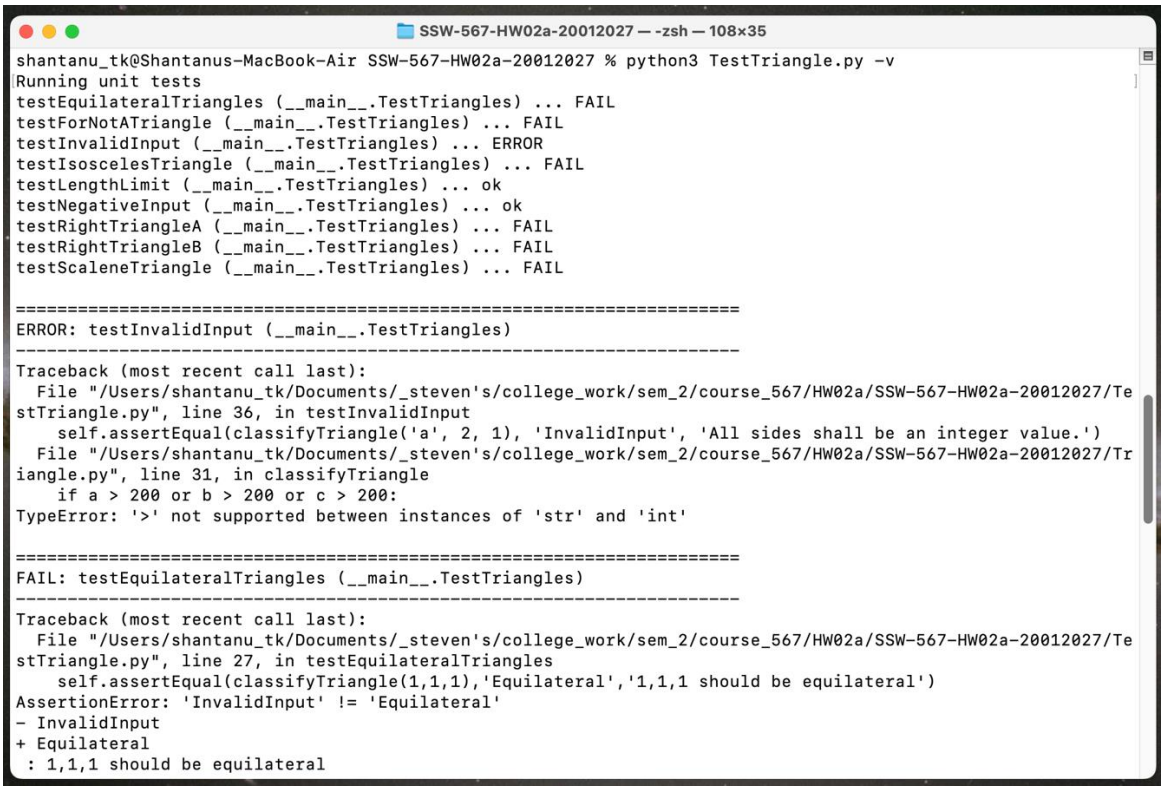
2. **Author:** Shantanu T Kshatriya

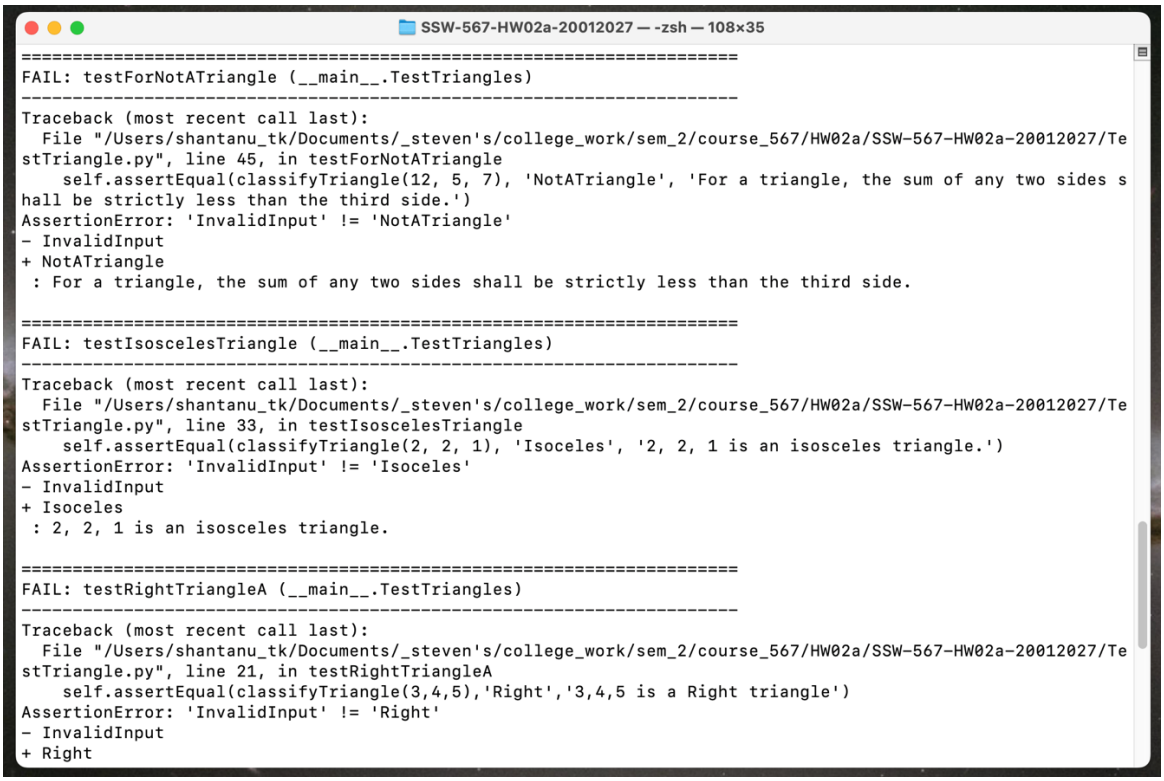
3. Summary:

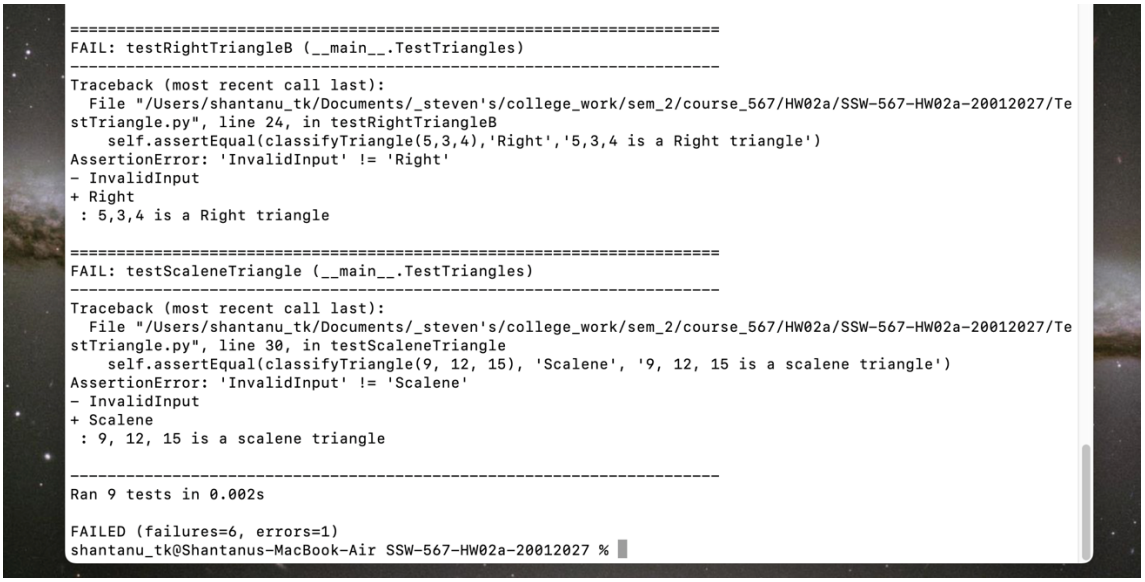
Test Report 1:

Number of test cases passed: 2
Number of test cases failed: 6
Error(s): 1
Total: 9

Test ID	Input	Expected Result	Actual Result	Pass or Fail
testRightTriangleA	3, 4, 5	Right	InvalidInput	Fail
testRightTriangleB	5, 4, 3	Right	InvalidInput	Fail
testEquilateralTriangles	1, 1, 1	Equilateral	InvalidInput	Fail
testScaleneTriangle	9, 12, 15	Scalene	InvalidInput	Fail
testIsoscelesTriangle	2, 2, 1	Isosceles	InvalidInput	Fail
testInvalidInput	'a', 2, 1	InvalidInput	TypeError: '>' not supported between instances of 'str' and 'int'	Fail
testLengthLimit	201, 9, 100	InvalidInput	InvalidInput	Pass
testNegativeInput	-1, 2, -9	InvalidInput	InvalidInput	Pass
testForNotATriangle	12, 5, 7	NotATriangle	InvalidInput	Fail


SSW-567-HW02a-20012027 — -zsh — 108×35
shantanu_tk@Shantanus-MacBook-Air SSW-567-HW02a-20012027 % python3 TestTriangle.py -v
Running unit tests
testEquilateralTriangles (__main__.TestTriangles) ... FAIL
testForNotATriangle (__main__.TestTriangles) ... FAIL
testInvalidInput (__main__.TestTriangles) ... ERROR
testIsoscelesTriangle (__main__.TestTriangles) ... FAIL
testLengthLimit (__main__.TestTriangles) ... ok
testNegativeInput (__main__.TestTriangles) ... ok
testRightTriangleA (__main__.TestTriangles) ... FAIL
testRightTriangleB (__main__.TestTriangles) ... FAIL
testScaleneTriangle (__main__.TestTriangles) ... FAIL
ERROR: testInvalidInput (__main__.TestTriangles)
Traceback (most recent call last):
File "/Users/shantanu_tk/Documents/_steven's/college_work/sem_2/course_567/HW02a/SSW-567-HW02a-20012027/TestTriangle.py", line 36, in testInvalidInput
self.assertEqual(classifyTriangle('a', 2, 1), 'InvalidInput', 'All sides shall be an integer value.')
File "/Users/shantanu_tk/Documents/_steven's/college_work/sem_2/course_567/HW02a/SSW-567-HW02a-20012027/Triangle.py", line 31, in classifyTriangle
if a > 200 or b > 200 or c > 200:
TypeError: '>' not supported between instances of 'str' and 'int'
FAIL: testEquilateralTriangles (__main__.TestTriangles)
Traceback (most recent call last):
File "/Users/shantanu_tk/Documents/_steven's/college_work/sem_2/course_567/HW02a/SSW-567-HW02a-20012027/TestTriangle.py", line 27, in testEquilateralTriangles
self.assertEqual(classifyTriangle(1,1,1),'Equilateral','1,1,1 should be equilateral')
AssertionError: 'InvalidInput' != 'Equilateral'
- InvalidInput
+ Equilateral
: 1,1,1 should be equilateral

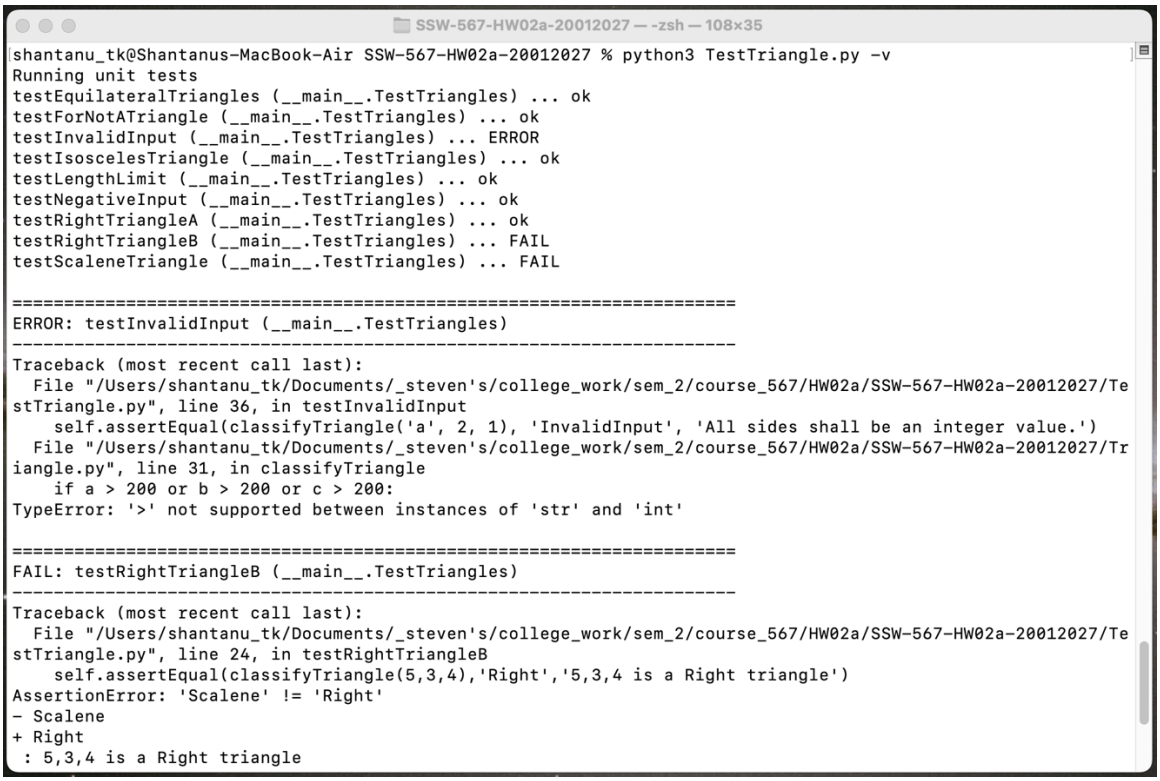

SSW-567-HW02a-20012027 — -zsh — 108×35
FAIL: testForNotATriangle (__main__.TestTriangles)
Traceback (most recent call last):
File "/Users/shantanu_tk/Documents/_steven's/college_work/sem_2/course_567/HW02a/SSW-567-HW02a-20012027/TestTriangle.py", line 45, in testForNotATriangle
self.assertEqual(classifyTriangle(12, 5, 7), 'NotATriangle', 'For a triangle, the sum of any two sides shall be strictly less than the third side.')
AssertionError: 'InvalidInput' != 'NotATriangle'
- InvalidInput
+ NotATriangle
: For a triangle, the sum of any two sides shall be strictly less than the third side.
FAIL: testIsoscelesTriangle (__main__.TestTriangles)
Traceback (most recent call last):
File "/Users/shantanu_tk/Documents/_steven's/college_work/sem_2/course_567/HW02a/SSW-567-HW02a-20012027/TestTriangle.py", line 33, in testIsoscelesTriangle
self.assertEqual(classifyTriangle(2, 2, 1), 'Isoceles', '2, 2, 1 is an isosceles triangle.')
AssertionError: 'InvalidInput' != 'Isoceles'
- InvalidInput
+ Isoceles
: 2, 2, 1 is an isosceles triangle.
FAIL: testRightTriangleA (__main__.TestTriangles)
Traceback (most recent call last):
File "/Users/shantanu_tk/Documents/_steven's/college_work/sem_2/course_567/HW02a/SSW-567-HW02a-20012027/TestTriangle.py", line 21, in testRightTriangleA
self.assertEqual(classifyTriangle(3,4,5),'Right','3,4,5 is a Right triangle')
AssertionError: 'InvalidInput' != 'Right'
- InvalidInput
+ Right

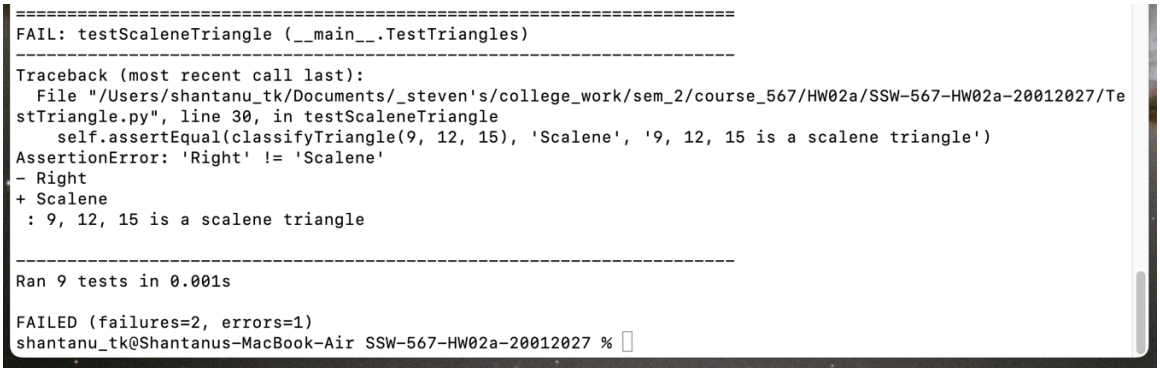


Test Report 2:

Number of test cases passed: 6
Number of test cases failed: 2
Error(s):1
Total: 9

Test ID	Input	Expected Result	Actual Result	Pass or Fail
testRightTriangleA	3, 4, 5	Right	Right	Pass
testRightTriangleB	5, 4, 3	Right	InvalidInput	Fail
testEquilateralTriangles	1, 1, 1	Equilateral	Equilateral	Pass
testScaleneTriangle	9, 12, 15	Scalene	InvalidInput	Fail
testIsoscelesTriangle	2, 2, 1	Isosceles	Isosceles	Pass
testInvalidInput	'a', 2, 1	InvalidInput	TypeError: '>' not supported between instances of 'str' and 'int'	Fail
testLengthLimit	201, 9, 100	InvalidInput	InvalidInput	Pass
testNegativeInput	-1, 2, -9	InvalidInput	InvalidInput	Pass
testForNotATriangle	12, 5, 7	NotATriangle	NotATriangle	Pass

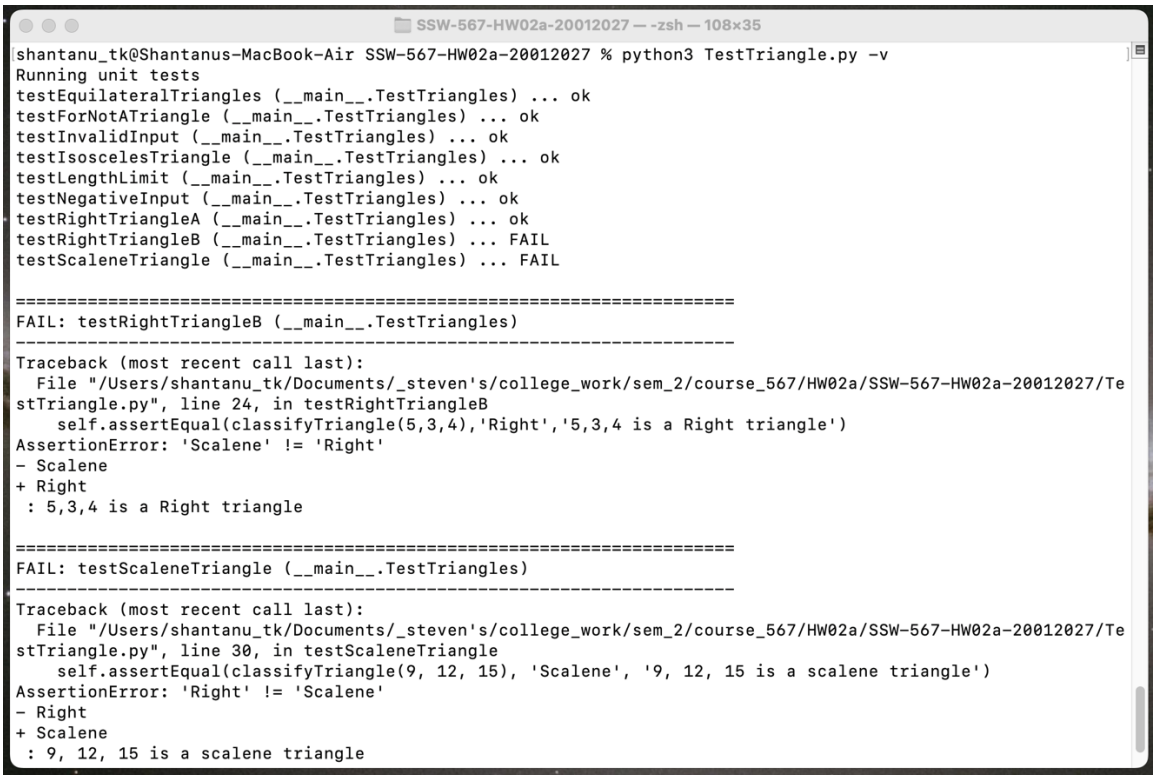

SSW-567-HW02a-20012027 — -zsh — 108×35
shantanu_tk@Shantanus-MacBook-Air SSW-567-HW02a-20012027 % python3 TestTriangle.py -v
Running unit tests
testEquilateralTriangles (__main__.TestTriangles) ... ok
testForNotATriangle (__main__.TestTriangles) ... ok
testInvalidInput (__main__.TestTriangles) ... ERROR
testIsoscelesTriangle (__main__.TestTriangles) ... ok
testLengthLimit (__main__.TestTriangles) ... ok
testNegativeInput (__main__.TestTriangles) ... ok
testRightTriangleA (__main__.TestTriangles) ... ok
testRightTriangleB (__main__.TestTriangles) ... FAIL
testScaleneTriangle (__main__.TestTriangles) ... FAIL
ERROR: testInvalidInput (__main__.TestTriangles)
Traceback (most recent call last):
File "/Users/shantanu_tk/Documents/_steven's/college_work/sem_2/course_567/HW02a/SSW-567-HW02a-20012027/TestTriangle.py", line 36, in testInvalidInput
self.assertEqual(classifyTriangle('a', 2, 1), 'InvalidInput', 'All sides shall be an integer value.')
File "/Users/shantanu_tk/Documents/_steven's/college_work/sem_2/course_567/HW02a/SSW-567-HW02a-20012027/Triangle.py", line 31, in classifyTriangle
if a > 200 or b > 200 or c > 200:
TypeError: '>' not supported between instances of 'str' and 'int'
FAIL: testRightTriangleB (__main__.TestTriangles)
Traceback (most recent call last):
File "/Users/shantanu_tk/Documents/_steven's/college_work/sem_2/course_567/HW02a/SSW-567-HW02a-20012027/TestTriangle.py", line 24, in testRightTriangleB
self.assertEqual(classifyTriangle(5,3,4),'Right','5,3,4 is a Right triangle')
AssertionError: 'Scalene' != 'Right'
- Scalene
+ Right
: 5,3,4 is a Right triangle

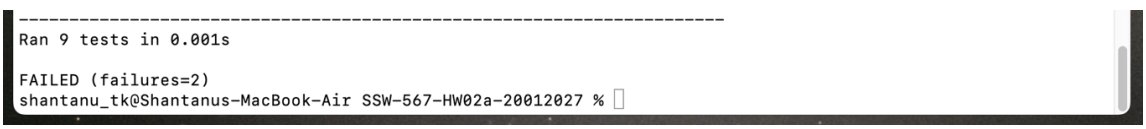

FAIL: testScaleneTriangle (__main__.TestTriangles)
Traceback (most recent call last):
File "/Users/shantanu_tk/Documents/_steven's/college_work/sem_2/course_567/HW02a/SSW-567-HW02a-20012027/TestTriangle.py", line 30, in testScaleneTriangle
self.assertEqual(classifyTriangle(9, 12, 15), 'Scalene', '9, 12, 15 is a scalene triangle')
AssertionError: 'Right' != 'Scalene'
- Right
+ Scalene
: 9, 12, 15 is a scalene triangle
Ran 9 tests in 0.001s
FAILED (failures=2, errors=1)
shantanu_tk@Shantanus-MacBook-Air SSW-567-HW02a-20012027 %

Test Report 3:

Number of test cases passed: 7
Number of test cases failed: 2
Error(s):0
Total: 9

Test ID	Input	Expected Result	Actual Result	Pass or Fail
testRightTriangleA	3, 4, 5	Right	Right	Pass
testRightTriangleB	5, 4, 3	Right	InvalidInput	Fail
testEquilateralTriangles	1, 1, 1	Equilateral	Equilateral	Pass
testScaleneTriangle	9, 12, 15	Scalene	InvalidInput	Fail
testIsoscelesTriangle	2, 2, 1	Isosceles	Isosceles	Pass
testInvalidInput	'a', 2, 1	InvalidInput	InvalidInput	Pass
testLengthLimit	201, 9, 100	InvalidInput	InvalidInput	Pass
testNegativeInput	-1, 2, -9	InvalidInput	InvalidInput	Pass
testForNotATriangle	12, 5, 7	NotATriangle	NotATriangle	Pass





Test Report 4:

Number of test cases passed: 9
Number of test cases failed: 0
Error(s):0
Total: 9

Test ID	Input	Expected Result	Actual Result	Pass or Fail
testRightTriangleA	3, 4, 5	Right	Right	Pass
testRightTriangleB	5, 4, 3	Right	Right	Pass
testEquilateralTriangles	1, 1, 1	Equilateral	Equilateral	Pass
testScaleneTriangle	9, 12, 15	Scalene	Scalene	Pass
testIsoscelesTriangle	2, 2, 1	Isosceles	Isosceles	Pass
testInvalidInput	'a', 2, 1	InvalidInput	InvalidInput	Pass
testLengthLimit	201, 9, 100	InvalidInput	InvalidInput	Pass
testNegativeInput	-1, 2, -9	InvalidInput	InvalidInput	Pass
testForNotATriangle	12, 5, 7	NotATriangle	NotATriangle	Pass

Reflection:

During the course of this assignment, I learnt a lot about how to create and write testcases. To me it was a learning experience as I have not worked on testing code before.

I learned that to write testcases, a tester needs to think about the requirements of the software as well as think from a perspective of the developer in order to test the logic the developers have written. I learned that is important to think about the values for testcases as well, because I even after all bugs from the Triangle.py were cleared I was unable to pass one testcase as the values were not right and the program returned before my testcase could reach the part to the code I was testing. Additionally, I understood the importance of writing concise, effective and efficient testcases. Most importantly I learned the importance and role of software testing in the overall software development cycle. Apart from learning the technical aspects of creating test scenarios and writing testcase, I also understood the importance of documentation and how to write test reports.

4. Detailed Result:

a. **Assumptions:** All lengths for triangle are of DATA TYPE <int>.

5. Final Result:

	Test Run 1	Test Run 2	Test 3	Test 4
Tests Planned	9	9	9	9
Tests Executed	9	9	9	9
Tests Passed	2	6	7	9
Tests Failed	7	3	2	0
Defects Found	5	1	1	0
Defects Fixed	-	5	1	1

6. Honor Pledge:

"I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination. I further pledge that I have not copied any material from a book, article, the Internet, or any other source except where I have expressly cited the source."

GitHub Link: https://github.com/binaryBodhi/SSW-567-HW02a-20012027