Prateek Singh Chauhan
CWID: 20016291

HW 2a - Testing a legacy program and reporting on testing results

Description:

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 is a starter implementation of the triangle classification program.

TestTriangle.py contains 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: Prateek Singh Chauhan

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."

Things Learned:

- Learned to think from the perspective of the tester
- Need to establish all oracles before testing
- Need to clarify all conditions and criteria of logic before testing
- Planning/Strategy is key to successful testing

Assumptions made:

All the values of length of the triangle is assumed to be integer as there's was no description in the problem statement.

Prateek Singh Chauhan
CWID: 20016291

Summary:

Test Run 1:

Test ID	Input	Expected Results	Actual Results	Pass or Fail
testRightTriangleA	3,4,5	Right	InvalidInput	Fail
testRightTriangleB	5,3,4	Right	InvalidInput	Fail
testEquilateralTriangles	1,1,1	Equilateral	InvalidInput	Fail
testIsoscelesTriangleA	2,1,2	Isoceles	InvalidInput	Fail
testScaleneTriangleA	3,1,5	NotATriangle	InvalidInput	Fail
testTriangleLengthGreaterThan200	201,201,1	InvalidInput	InvalidInput	Pass
testTriangeLengthForNegatoveValues	-1,1,1	InvalidInput	InvalidInput	Pass
testTriangleLengthForInputString	'a',1,1	InvalidInput	TypeError: '>' not supported between instances of 'str' and 'int'	Fail

- Total Test cases: 8
  - Number of Passed cases: 2
  - Number of Failed cases: 6
  - Errors: 1

```
PS C:\Users\chauh\Documents\SSW 567\ssw567_hw_software_eng_test\HW02a Testing Legacy Problem> python -m unittest TestTriangle.py
FFFFFCase passed: -1,1,1 should not be triangle
.ECase passed: 201,201,1 should not be triangle
.
======================================================================
ERROR: testTriangleLengthForInputString (TestTriangle.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\chauh\Documents\SSW 567\ssw567_hw_software_eng_test\HW02a Testing Legacy Problem\TestTriangle.py", line 45, in testTriangleLengthForInputString
    self.assertEqual(classifyTriangle('a',1,1), 'InvalidInput', 'All length sides should be numeric')
  File "C:\Users\chauh\Documents\SSW 567\ssw567_hw_software_eng_test\HW02a Testing Legacy Problem\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 (TestTriangle.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\chauh\Documents\SSW 567\ssw567_hw_software_eng_test\HW02a Testing Legacy Problem\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

======================================================================
FAIL: testIsoscelesTriangleA (TestTriangle.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\chauh\Documents\SSW 567\ssw567_hw_software_eng_test\HW02a Testing Legacy Problem\TestTriangle.py", line 30, in testIsoscelesTriangleA
    self.assertEqual(classifyTriangle(1,1,2),'Isoceles', '1,1,2 shold be isoceles triangle')
AssertionError: 'InvalidInput' != 'Isoceles'
- InvalidInput
+ Isoceles
 : 1,1,2 shold be isoceles triangle
```

Prateek Singh Chauhan
CWID: 20016291

```
======================================================================
FAIL: testRightTriangleA (TestTriangle.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\chauh\Documents\SSW 567\ssw567_hw_software_eng_test\HW02a Testing Legacy Problem\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
 : 3,4,5 is a Right triangle

======================================================================
FAIL: testRightTriangleB (TestTriangle.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\chauh\Documents\SSW 567\ssw567_hw_software_eng_test\HW02a Testing Legacy Problem\TestTriangle.py", line 24, in testRightTriangleB
    self.assertEqual(classifyTriangle(5,3,4),'Right','5,3,4 is a Right triangle')
AssertionError: 'InvalidInput' != 'Right'
- InvalidInput
+ Right
 : 5,3,4 is a Right triangle

======================================================================
FAIL: testScaleneTriangleA (TestTriangle.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\chauh\Documents\SSW 567\ssw567_hw_software_eng_test\HW02a Testing Legacy Problem\TestTriangle.py", line 33, in testScaleneTriangleA
    self.assertEqual(classifyTriangle(3,1,5), 'Scalene', '3,1,5 shold be scalene triangle')
AssertionError: 'InvalidInput' != 'Scalene'
- InvalidInput
+ Scalene
 : 3,1,5 shold be scalene triangle

----------------------------------------------------------------------
Ran 8 tests in 0.002s

FAILED (failures=5, errors=1)
PS C:\Users\chauh\Documents\SSW 567\ssw567_hw_software_eng_test\HW02a Testing Legacy Problem>
```

Prateek Singh Chauhan
CWID: 20016291

Test Run 2:

Code Changes:

- Moved Data type validation before any triangle numeric length validation
- Corrected validation from b<=b to b<=0

```
-    if a <= 0 or b <= b or c <= 0:
         return 'InvalidInput'

39
40   +    if a <= 0 or b <= 0 or c <= 0:
41            return 'InvalidInput'
```

Test ID	Input	Expected Results	Actual Results	Pass or Fail
testRightTriangleA	3,4,5	Right	NotATriangle	Fail
testRightTriangleB	5,3,4	Right	NotATriangle	Fail
testEquilateralTriangles	1,1,1	Equilateral	NotATriangle	Fail
testIsoscelesTriangleA	2,1,2	Isoceles	NotATriangle	Fail
testScaleneTriangleA	3,1,5	NotATriangle	NotATriangle	Pass
testTriangleLengthGreaterThan200	201,201,1	InvalidInput	InvalidInput	Pass
testTriangeLengthForNegatoveValues	-1,1,1	InvalidInput	InvalidInput	Pass
testTriangleLengthForInputString	'a',1,1	InvalidInput	InvalidInput	Pass

- After Code Changes: Total Test cases: 8
  - Number of Passed cases: 4
  - Number of Failed cases: 4
  - Errors: 0

```
if (a > (b + c)) or (b > (a + c)) or (c > (a + b)):
```

```
PS C:\Users\chauh\Documents\SSW 567\ssw567_hw_software_eng_test\HW02a Testing Legacy Problem> python -m unittest TestTriangle.py
FFFFCase passed: 3,1,5 should not be scalene triangle
.Case passed: -1,1,1 should not be triangle
.Case passed: All input should be numeric
.Case passed: 201,201,1 should not be triangle
.
======================================================================
FAIL: testEquilateralTriangles (TestTriangle.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\chauh\Documents\SSW 567\ssw567_hw_software_eng_test\HW02a Testing Legacy Problem\TestTriangle.py", line 29, in testEquilateralTriangles
    self.assertEqual(classifyTriangle(1,1,1),'Equilateral','1,1,1 should be equilateral')
AssertionError: 'NotATriangle' != 'Equilateral'
- NotATriangle
+ Equilateral
 : 1,1,1 should be equilateral

======================================================================
FAIL: testIsoscelesTriangleA (TestTriangle.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\chauh\Documents\SSW 567\ssw567_hw_software_eng_test\HW02a Testing Legacy Problem\TestTriangle.py", line 33, in testIsoscelesTriangleA
    self.assertEqual(classifyTriangle(1,1,2),'Isoceles', '1,1,2 shold be isoceles triangle')
AssertionError: 'NotATriangle' != 'Isoceles'
- NotATriangle
+ Isoceles
 : 1,1,2 shold be isoceles triangle

======================================================================
FAIL: testRightTriangleA (TestTriangle.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\chauh\Documents\SSW 567\ssw567_hw_software_eng_test\HW02a Testing Legacy Problem\TestTriangle.py", line 21, in testRightTriangleA
    self.assertEqual(classifyTriangle(3,4,5),'Right','3,4,5 is a Right triangle')
AssertionError: 'NotATriangle' != 'Right'
- NotATriangle
+ Right
 : 3,4,5 is a Right triangle
```

Prateek Singh Chauhan
CWID: 20016291

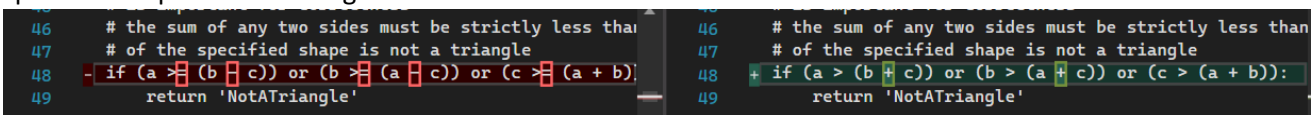
```
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\chauh\Documents\SSW 567\ssw567_hw_software_eng_test\HW02a Testing Legacy Problem\TestTriangle.py", line 21, in testRightTriangleA
    self.assertEqual(classifyTriangle(3,4,5),'Right','3,4,5 is a Right triangle')
AssertionError: 'NotATriangle' != 'Right'
- NotATriangle
+ Right
 : 3,4,5 is a Right triangle

======================================================================
FAIL: testRightTriangleB (TestTriangle.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\chauh\Documents\SSW 567\ssw567_hw_software_eng_test\HW02a Testing Legacy Problem\TestTriangle.py", line 25, in testRightTriangleB
    self.assertEqual(classifyTriangle(5,3,4),'Right','5,3,4 is a Right triangle')
AssertionError: 'NotATriangle' != 'Right'
- NotATriangle
+ Right
 : 5,3,4 is a Right triangle

----------------------------------------------------------------------
Ran 8 tests in 0.002s

FAILED (failures=4)
PS C:\Users\chauh\Documents\SSW 567\ssw567_hw_software_eng_test\HW02a Testing Legacy Problem>
```

Prateek Singh Chauhan
CWID: 20016291

Test Run 3:

Code Changes:

- Corrected validation for the sum of any two sides must be strictly less than the third side of the specified shape is not a triangle



Test ID	Input	Expected Results	Actual Results	Pass or Fail
testRightTriangleA	3,4,5	Right	NotATriangle	Fail
testRightTriangleB	5,3,4	Right	NotATriangle	Fail
testEquilateralTriangles	1,1,1	Equilateral	Equilateral	Pass
testIsoscelesTriangleA	2,1,2	Isoceles	NotATriangle	Fail
testScaleneTriangleA	3,1,5	NotATriangle	NotATriangle	Pass
testTriangleLengthGreaterThan200	201,201,1	InvalidInput	InvalidInput	Pass
testTriangeLengthForNegatoveValues	-1,1,1	InvalidInput	InvalidInput	Pass
testTriangleLengthForInputString	'a',1,1	InvalidInput	InvalidInput	Pass

- After Code Changes: Total Test cases: 8
  - Number of Passed cases: 5
  - Number of Failed cases: 3
  - Errors: 0

```
PS C:\Users\chauh\Documents\SSW 567\ssw567_hw_software_eng_test\HW02a Testing Legacy Problem> python -m unittest TestTriangle.py
Case passed: 1,1,1 should be equilateral
.FFFCase passed: 3,1,5 should not be scalene triangle
.Case passed: -1,1,1 should not be triangle
.Case passed: All input should be numeric
.Case passed: 201,201,1 should not be triangle
.
======================================================================
FAIL: testIsoscelesTriangleA (TestTriangle.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\chauh\Documents\SSW 567\ssw567_hw_software_eng_test\HW02a Testing Legacy Problem\TestTriangle.py", line 33, in testIsoscelesTriangleA
    self.assertEqual(classifyTriangle(1,1,2),'Isoceles', '1,1,2 shold be isoceles triangle')
AssertionError: 'Equilateral' != 'Isoceles'
- Equilateral
+ Isoceles
 : 1,1,2 shold be isoceles triangle

======================================================================
FAIL: testRightTriangleA (TestTriangle.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\chauh\Documents\SSW 567\ssw567_hw_software_eng_test\HW02a Testing Legacy Problem\TestTriangle.py", line 21, in testRightTriangleA
    self.assertEqual(classifyTriangle(3,4,5),'Right','3,4,5 is a Right triangle')
AssertionError: 'Scalene' != 'Right'
- Scalene
+ Right
 : 3,4,5 is a Right triangle

======================================================================
FAIL: testRightTriangleB (TestTriangle.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\chauh\Documents\SSW 567\ssw567_hw_software_eng_test\HW02a Testing Legacy Problem\TestTriangle.py", line 25, 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
```

```
----------------------------------------------------------------------
Ran 8 tests in 0.001s

FAILED (failures=3)
PS C:\Users\chauh\Documents\SSW 567\ssw567_hw_software_eng_test\HW02a Testing Legacy Problem>
```

Prateek Singh Chauhan
CWID: 20016291

Test Run 4:

Code Changes:

- Corrected validation for Equilateral Triangle

- Corrected validation for Right Triangle

- Corrected validation for Scalene Triangle

Test ID	Input	Expected Results	Actual Results	Pass or Fail
testRightTriangleA	3,4,5	Right	Right	Pass
testRightTriangleB	5,3,4	Right	Right	Pass
testEquilateralTriangles	1,1,1	Equilateral	Equilateral	Pass
testIsoscelesTriangleA	2,1,2	Isoceles	Isoceles	Pass
testScaleneTriangleA	3,1,5	NotATriangle	NotATriangle	Pass
testTriangleLengthGreaterThan200	201,201,1	InvalidInput	InvalidInput	Pass
testTriangeLengthForNegatoveValues	-1,1,1	InvalidInput	InvalidInput	Pass
testTriangleLengthForInputString	'a',1,1	InvalidInput	InvalidInput	Pass

- After Code Changes: Total Test cases: 8
  - Number of Passed cases: 8
  - Number of Failed cases: 0
  - Errors: 0

```
PS C:\Users\chauh\Documents\SSW 567\ssw567_hw_software_eng_test\HW02a Testing Legacy Problem> python -m unittest TestTriangle.py
Case passed: 1,1,1 should be equilateral
.Case passed: 2,1,2 should be isoceles triangle
.Case passed: 3,4,5 is a Right triangle
.Case passed: 5,3,4 is a Right triangle
.Case passed: 3,1,5 should not be scalene triangle
.Case passed: -1,1,1 should not be triangle
.Case passed: All input should be numeric
.Case passed: 201,201,1 should not be triangle
.
----------------------------------------------------------------------
Ran 8 tests in 0.000s

OK
```

Prateek Singh Chauhan
CWID: 20016291

Final Report:

	Test Run 1	Test Run 2	Test Run 3	Test Run 4
Tests Planned	8	8	8	8
Tests Executed	8	8	8	8
Tests Passed	2	4	5	8
Tests Failed	6	4	3	0
Defects Found	2	1	3	0
Defects Fixed	–	2	1	3

Code Repo:
https://github.com/sabudanakichdi/ssw567_hw_software_eng_test/tree/main/HW02a%20Testing%20Legacy%20Problem