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

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. 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.

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.

Triangle.py contains an implementation of the classifyTriangle() function with a few bugs.

TestTriangle.py contains the initial set of test cases

Author: Meghana Arumilli

GitHub Repo: https://github.com/Meghana-Arumilli/Testing567

Summary:

For initial test set -

Test ID	Input	Expected Results	Actual Result	Pass or Fail
testRightTriangle1()	3,4,5	Right	InvalidInput	Fail
testRightTriangle2()	5,3,4	Right	InvalidInput	Fail
testRightTriangle3()	8,6,10	Right	InvalidInput	Fail
testRightTriangle4()	21,6,10	InvalidInputOne	InvalidInput	Fail
testEquilateralTriangle1()	1,1,1	Equilateral	InvalidInput	Fail
testEquilateralTriangle2()	30,30,30	Equilateral	InvalidInput	Fail
testEquilateralTriangle3()	16,16,16	Equilateral	InvalidInput	Fail
testEquilateralTriangle4()	10,1,10	InvalidInputTwo	InvalidInputTwo	Pass
testIsoscelesTriangle1()	5,5,3	Isosceles	InvalidInput	Fail
testIsoscelesTriangle2()	4,6,6	Isosceles	InvalidInput	Fail
testIsoscelesTriangle3()	8,8,7	Isosceles	InvalidInput	Fail
testIsoscelesTriangle4()	6,6,4	Isosceles	InvalidInput	Fail
testScaleneTriangle1()	9,10,11	Scalene	InvalidInput	Fail
testScaleneTriangle2()	30,40,52	Scalene	InvalidInput	Fail
testScaleneTriangle3()	100,110,112	Scalene	InvalidInput	Fail
testScaleneTriangle4()	25,25,14	InvalidInputThree	InvalidInputTwo	Pass
testNotATriangle1()	2,2,8	NotATriangle	InvalidInput	Fail
testNotATriangle2()	1,19,5	NotATriangle	InvalidInput	Fail
testNotATriangle3()	1,1,5	NotATriangle	InvalidInput	Fail
testNotATriangle4()	1,17,5	NotATriangle	InvalidInput	Fail

For Final test set-

Test ID	**Input**	**Expected Results**	**Actual Result**	**Pass or Fail**
testRightTriangle1()	3,4,5	Right	Right	Pass
testRightTriangle2()	5,3,4	Right	Right	Pass
testRightTriangle3()	8,6,10	Right	Right	Pass
testRightTriangle4()	21,6,10	InvalidInputOne	InvalidInputOne	Pass
testEquilateralTriangle1()	1,1,1	Equilateral	Equilateral	Pass
testEquilateralTriangle2()	30,30,30	Equilateral	Equilateral	Pass
testEquilateralTriangle3()	16,16,16	Equilateral	Equilateral	Pass
testEquilateralTriangle4()	10,1,10	InvalidInputTwo	InvalidInputTwo	Pass
testIsoscelesTriangle1()	5,5,3	Isosceles	Isosceles	Pass
testIsoscelesTriangle2()	4,6,6	Isosceles	Isosceles	Pass
testIsoscelesTriangle3()	8,8,7	Isosceles	Isosceles	Pass
testIsoscelesTriangle4()	6,6,4	Isosceles	Isosceles	Pass
testScaleneTriangle1()	9,10,11	Scalene	Scalene	Pass
testScaleneTriangle2()	30,40,52	Scalene	Scalene	Pass
testScaleneTriangle3()	100,110,112	Scalene	Scalene	Pass
testScaleneTriangle4()	25,25,14	InvalidInputThree	InvalidInputThree	Pass
testNotATriangle1()	2,2,8	NotATriangle	NotATriangle	Pass
testNotATriangle2()	1,19,5	NotATriangle	NotATriangle	Pass
testNotATriangle3()	1,1,5	NotATriangle	NotATriangle	Pass
testNotATriangle4()	1,17,5	NotATriangle	NotATriangle	Pass

	Test Run 1	Test Run2	Test Run3
Tests Planned	20	20	20
Tests Executed	20	20	20
Tests Passed	2	14	20
Defects Found	18	6	0
Defects Fixed	0	12	6

- **Reflection – this is where you actually think about the assignment after it is over – what did you learn? What worked, what didn't?**
  Following are the learning from the assignment:
  - Proper implementation of unit testing on some other person's code.
  - Including the unit test for edge cases
  - Fixing buggy code through unit testing.

Conditions mentioned in the Triangle.py had issues so the program was working with few inputs but was failing for most of the inputs.

5. Honor pledge

6. Detailed results, if any:

- Details of any assumptions or constraints made

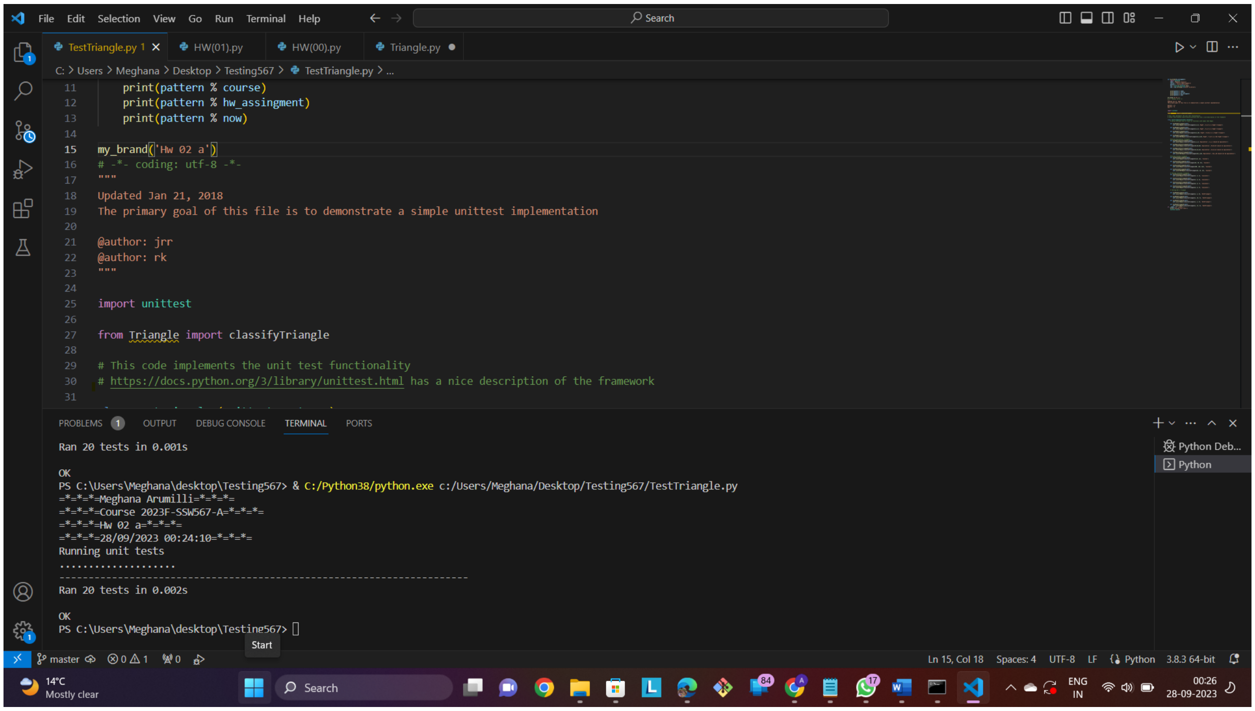
Assumption about the upper limit of side length.
Constraints put over the datatype i.e., int datatype.

- **A description of whatever data inputs you used**

Data inputs were used for every necessary conditions like
- float datatype, alphabet were used as data set for input condition
- Upper limit for the length of triangle was checked
- Lower limit for the length of triangle was checked
- Different values for NotATriangle condition
- Different values for different triangle conditions

- **An explanation of the results of your work. In this case, upload copies of the code, the sanity tests, and the results, plus any other relevant results about the tools selection/installation/usage.**

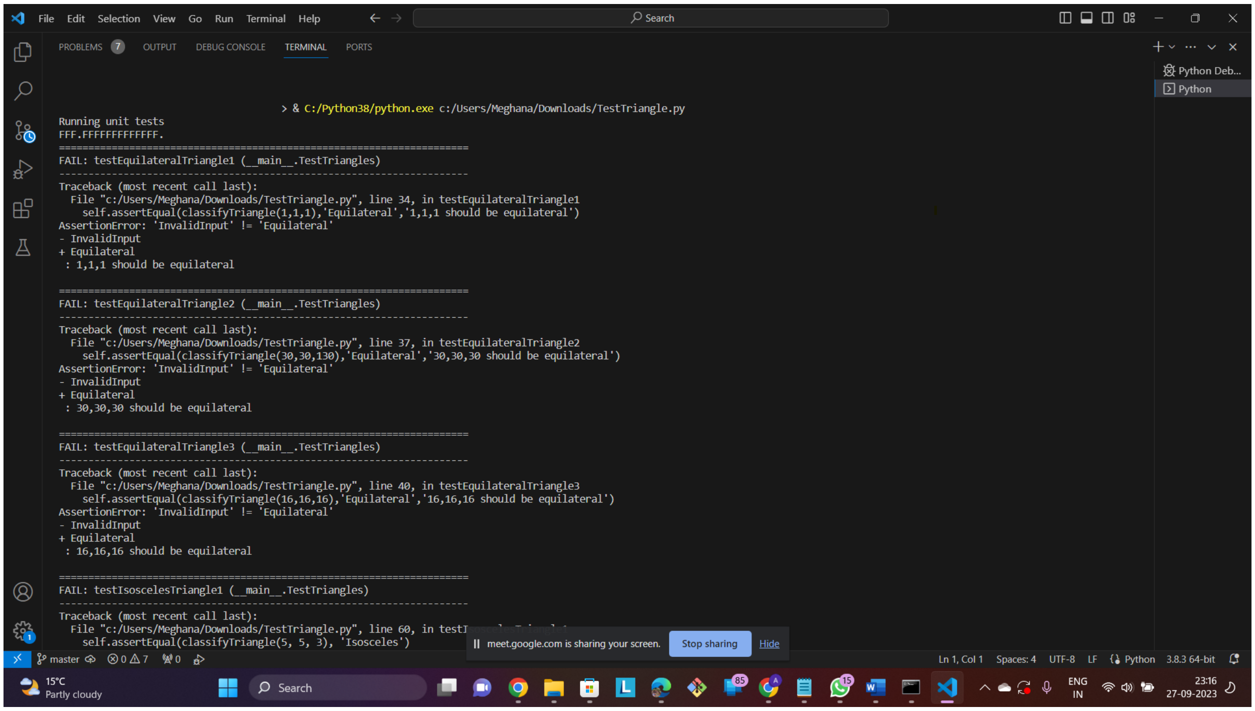
Final Test Set run:



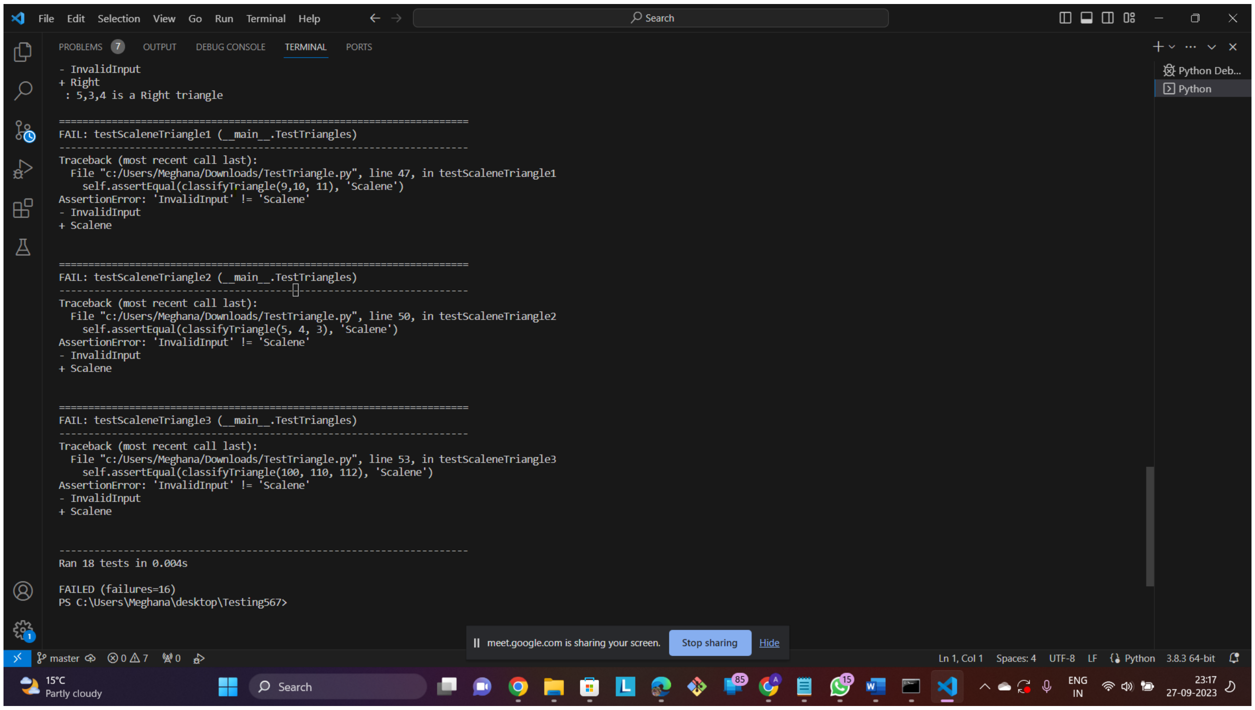
```python
    print(pattern % course)
    print(pattern % hw_assingment)
    print(pattern % now)

my_brand('Hw 02 a')
# -*- coding: utf-8 -*-
"""

Updated Jan 21, 2018
The primary goal of this file is to demonstrate a simple unittest implementation

@author: jrr
@author: rk
"""

import unittest

from Triangle import classifyTriangle

# This code implements the unit test functionality
# https://docs.python.org/3/library/unittest.html has a nice description of the framework

```

```
Ran 20 tests in 0.001s

OK
PS C:\Users\Meghana\desktop\Testing567> & C:/Python38/python.exe c:/Users/Meghana/Desktop/Testing567/TestTriangle.py
=*=*=*=Meghana Arumilli=*=*=*=
=*=*=*=Course 2023F-SSW567-A=*=*=*=
=*=*=*=Hw 02 a=*=*=*=
=*=*=*=28/09/2023 00:24:10=*=*=*=
Running unit tests
....................
----------------------------------------------------------------------
Ran 20 tests in 0.002s

OK
PS C:\Users\Meghana\desktop\Testing567> 
```

Initial Test Set Run1 :

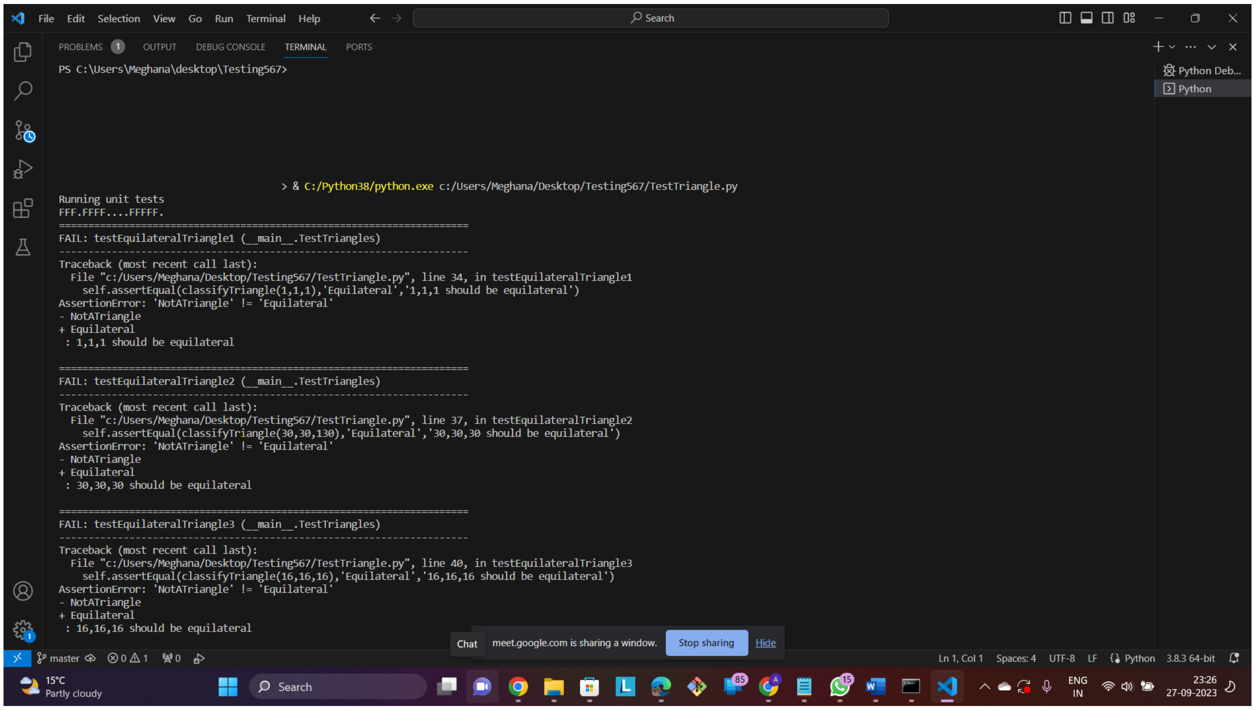


```
                                          > & C:/Python38/python.exe c:/Users/Meghana/Downloads/TestTriangle.py
Running unit tests
FFF.FFFFFFFFFFFF.
======================================================================
FAIL: testEquilateralTriangle1 (__main__.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "c:/Users/Meghana/Downloads/TestTriangle.py", line 34, in testEquilateralTriangle1
    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: testEquilateralTriangle2 (__main__.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "c:/Users/Meghana/Downloads/TestTriangle.py", line 37, in testEquilateralTriangle2
    self.assertEqual(classifyTriangle(30,30,130),'Equilateral','30,30,30 should be equilateral')
AssertionError: 'InvalidInput' != 'Equilateral'
- InvalidInput
+ Equilateral
 : 30,30,30 should be equilateral

======================================================================
FAIL: testEquilateralTriangle3 (__main__.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "c:/Users/Meghana/Downloads/TestTriangle.py", line 40, in testEquilateralTriangle3
    self.assertEqual(classifyTriangle(16,16,16),'Equilateral','16,16,16 should be equilateral')
AssertionError: 'InvalidInput' != 'Equilateral'
- InvalidInput
+ Equilateral
 : 16,16,16 should be equilateral

======================================================================
FAIL: testIsoscelesTriangle1 (__main__.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "c:/Users/Meghana/Downloads/TestTriangle.py", line 60, in testI
    self.assertEqual(classifyTriangle(5, 5, 3), 'Isosceles')
```



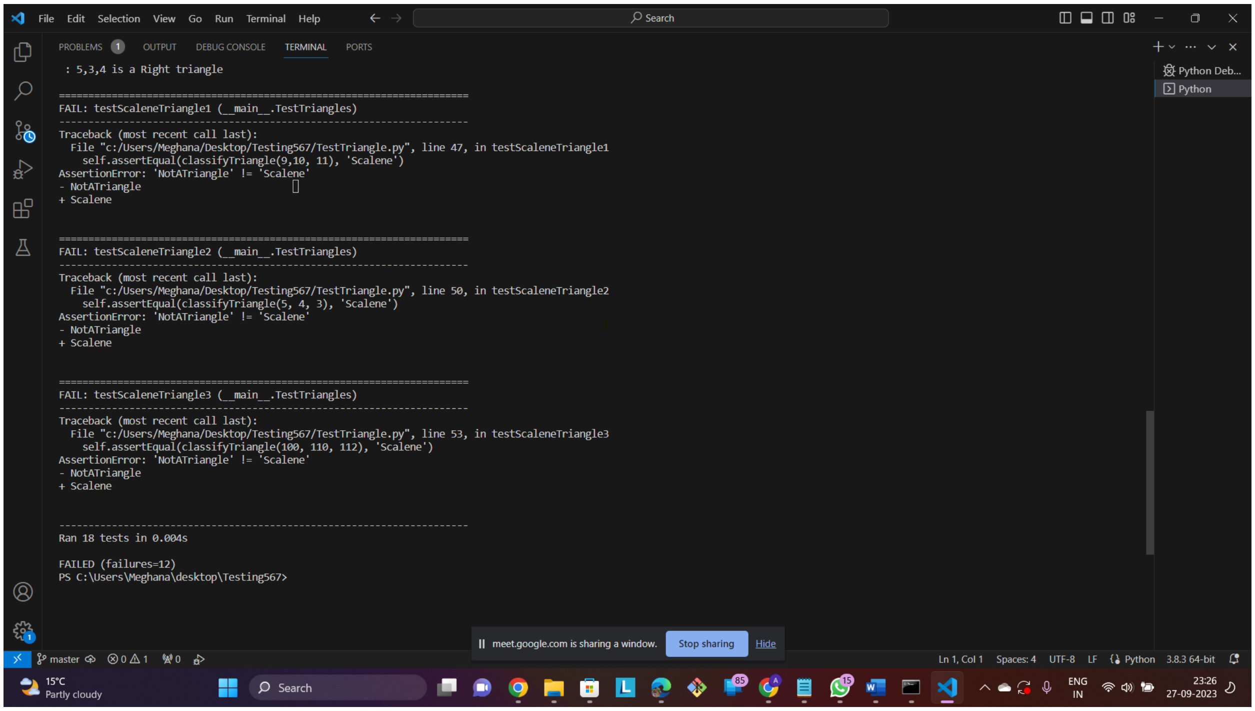
```
- InvalidInput
+ Right
 : 5,3,4 is a Right triangle

======================================================================
FAIL: testScaleneTriangle1 (__main__.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "c:/Users/Meghana/Downloads/TestTriangle.py", line 47, in testScaleneTriangle1
    self.assertEqual(classifyTriangle(9,10, 11), 'Scalene')
AssertionError: 'InvalidInput' != 'Scalene'
- InvalidInput
+ Scalene


======================================================================
FAIL: testScaleneTriangle2 (__main__.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "c:/Users/Meghana/Downloads/TestTriangle.py", line 50, in testScaleneTriangle2
    self.assertEqual(classifyTriangle(5, 4, 3), 'Scalene')
AssertionError: 'InvalidInput' != 'Scalene'
- InvalidInput
+ Scalene


======================================================================
FAIL: testScaleneTriangle3 (__main__.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "c:/Users/Meghana/Downloads/TestTriangle.py", line 53, in testScaleneTriangle3
    self.assertEqual(classifyTriangle(100, 110, 112), 'Scalene')
AssertionError: 'InvalidInput' != 'Scalene'
- InvalidInput
+ Scalene


----------------------------------------------------------------------
Ran 18 tests in 0.004s

FAILED (failures=16)
PS C:\Users\Meghana\desktop\Testing567>
```

Initial Test Set Run 2:



File Edit Selection View Go Run Terminal Help
Search
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python Deb...
Python
Chat
meet.google.com is sharing a window.
Stop sharing
Hide
master
Ln 1, Col 1 Spaces: 4 UTF-8 LF Python 3.8.3 64-bit
15°C
Partly cloudy
ENG
IN
23:26
27-09-2023

```
PS C:\Users\Meghana\desktop\Testing567>




                                        > & C:/Python38/python.exe c:/Users/Meghana/Desktop/Testing567/TestTriangle.py
Running unit tests
FFF.FFFF....FFFFF.
======================================================================
FAIL: testEquilateralTriangle1 (__main__.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "c:/Users/Meghana/Desktop/Testing567/TestTriangle.py", line 34, in testEquilateralTriangle1
    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: testEquilateralTriangle2 (__main__.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "c:/Users/Meghana/Desktop/Testing567/TestTriangle.py", line 37, in testEquilateralTriangle2
    self.assertEqual(classifyTriangle(30,30,130),'Equilateral','30,30,30 should be equilateral')
AssertionError: 'NotATriangle' != 'Equilateral'
- NotATriangle
+ Equilateral
 : 30,30,30 should be equilateral

======================================================================
FAIL: testEquilateralTriangle3 (__main__.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "c:/Users/Meghana/Desktop/Testing567/TestTriangle.py", line 40, in testEquilateralTriangle3
    self.assertEqual(classifyTriangle(16,16,16),'Equilateral','16,16,16 should be equilateral')
AssertionError: 'NotATriangle' != 'Equilateral'
- NotATriangle
+ Equilateral
 : 16,16,16 should be equilateral
```



File Edit Selection View Go Run Terminal Help
Search
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python Deb...
Python
meet.google.com is sharing a window.
Stop sharing
Hide
master
Ln 1, Col 1 Spaces: 4 UTF-8 LF Python 3.8.3 64-bit
15°C
Partly cloudy
ENG
IN
23:26
27-09-2023

```
 : 5,3,4 is a Right triangle

======================================================================
FAIL: testScaleneTriangle1 (__main__.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "c:/Users/Meghana/Desktop/Testing567/TestTriangle.py", line 47, in testScaleneTriangle1
    self.assertEqual(classifyTriangle(9,10, 11), 'Scalene')
AssertionError: 'NotATriangle' != 'Scalene'
- NotATriangle
+ Scalene


======================================================================
FAIL: testScaleneTriangle2 (__main__.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "c:/Users/Meghana/Desktop/Testing567/TestTriangle.py", line 50, in testScaleneTriangle2
    self.assertEqual(classifyTriangle(5, 4, 3), 'Scalene')
AssertionError: 'NotATriangle' != 'Scalene'
- NotATriangle
+ Scalene


======================================================================
FAIL: testScaleneTriangle3 (__main__.TestTriangles)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "c:/Users/Meghana/Desktop/Testing567/TestTriangle.py", line 53, in testScaleneTriangle3
    self.assertEqual(classifyTriangle(100, 110, 112), 'Scalene')
AssertionError: 'NotATriangle' != 'Scalene'
- NotATriangle
+ Scalene


----------------------------------------------------------------------
Ran 18 tests in 0.004s

FAILED (failures=12)
PS C:\Users\Meghana\desktop\Testing567>
```