Triangle Classification

TriangleClassification.py

```python
import unittest

# Function to determine if a triangle is scalene, isosceles, equilateral, or right triangles
def classify_triangle(a, b, c):
    # Ensure that all sides have positive numbers that are greater than zero
    if a <= 0 or b <= 0 or c <= 0:
        return "Sides should be greater than 0"

    if not (a + b > c and a + c > b and b + c > a):
        return "Not a triangle"

    # right triangle logic
    if round(a**2 + b**2, 5) == round(c**2, 5) or \
       round(a**2 + c**2, 5) == round(b**2, 5) or \
       round(b**2 + c**2, 5) == round(a**2, 5):
        triangle_type = "Right"
    else:
        triangle_type = ""

    # Classify based on side lengths
    if a == b == c:
        triangle_type += " Equilateral"
    elif a == b or b == c or a == c:
        triangle_type += " Isosceles"
    else:
        triangle_type += " Scalene"

    return triangle_type.strip()
```

TriangleClassification.py

```python

# Function to run Unit Test
class TestClassifyTriangle(unittest.TestCase):
    def test_equilateral(self):
        self.assertEqual(classify_triangle(8, 8, 8), "Equilateral")

    def test_isosceles(self):
        self.assertEqual(classify_triangle(10, 10, 15), "Isosceles")
        self.assertEqual(classify_triangle(10, 15, 10), "Isosceles")
        self.assertEqual(classify_triangle(15, 10, 10), "Isosceles")


    def test_scalene(self):
        self.assertEqual(classify_triangle(8, 10, 12), "Scalene")

    def test_right_triangle(self):
        self.assertEqual(classify_triangle(6, 8, 10), "Right Scalene")
        self.assertEqual(classify_triangle(9, 12, 15), "Right Scalene")

    def test_invalid_triangle(self):
        self.assertEqual(classify_triangle(1, 2, 10), "Not a triangle")
        self.assertEqual(classify_triangle(5, 9, 20), "Not a triangle")

    def test_invalid_input(self):
        self.assertEqual(classify_triangle(-2, 4, 5), "Sides should be greater than 0")
        self.assertEqual(classify_triangle(0, 8, 7), "Sides should be greater than 0")
        self.assertEqual(classify_triangle(8, 8, 0), "Sides should be greater than 0")

if __name__ == "__main__":
    unittest.main()
```

Output of Unit Test Cases

TriangleClassification.py

```
TriangleClassification.py > ...
32  class TestClassifyTriangle(unittest.TestCase):
43          self.assertEqual(classify_triangle(8, 10, 12), "Scalene")
44
45      def test_right_triangle(self):
46          self.assertEqual(classify_triangle(6, 8, 10), "Right Scalene")
47          self.assertEqual(classify_triangle(9, 12, 15), "Right Scalene")
48
49      def test_invalid_triangle(self):
50          self.assertEqual(classify_triangle(1, 2, 10), "Not a triangle")
51          self.assertEqual(classify_triangle(5, 9, 20), "Not a triangle")
52
53      def test_invalid_input(self):
54          self.assertEqual(classify_triangle(-2, 4, 5), "Sides should be greater than 0")
55          self.assertEqual(classify_triangle(0, 8, 7), "Sides should be greater than 0")
56          self.assertEqual(classify_triangle(8, 8, 0), "Sides should be greater than 0")
57
58  if __name__ == "__main__":
59      unittest.main()
60
```

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

```
PS C:\Users\Admin\Documents\GitHub\SSW-567> python -m unittest TriangleClassification.py
......
----------------------------------------------------------------------
Ran 6 tests in 0.001s

OK
PS C:\Users\Admin\Documents\GitHub\SSW-567>
```

Breaking Down the Result

1. **... :** The dots (.) represent successful tests. Each dot corresponds to one test case that passed.
2. **Ran 6 tests:** This indicates the total number of tests that were executed here was 6.
3. **in 0.001s:** This shows how long the test suite took to run, which is useful for performance tracking.
4. **OK:** This means that all the tests passed successfully. If there were failures or errors, this message would change.

Types of Results

1. **Success (OK):** All tests passed.
2. **Failure (FAILED):** One or more tests failed because the actual output didn’t match the expected output.
3. **Error (ERROR):** An error occurred in one or more tests (e.g., invalid input or exceptions being raised that weren't caught).

Akshay Kumar Talur Narasimmulu
SSW 567
https://github.com/AkshayKumarTN/SSW-567