

SSW567 HW01

Although the assignment wasn't particularly difficult, there were a few issues encountered with the testing framework. We were required to adjust the format of the triangle implementation multiple times to better suit the requirements of unittest, first using a single function for all of the triangle detection code but then resorting to a class with multiple functions.

The requirements specification was very self-explanatory. It concisely listed what was required for the assignment, and our code should accurately reflect the requirements.

We encountered a few issues with the unittest. It initially did not work properly with the first implementation of our code. This involved us first using functions and then resorting to a method that used classes. After the code was altered accordingly, the tests seemed to be functional but refused to return a log of results with the docstrings. We eventually got it to work properly.

We determined that three tests for each case were sufficient. Each aspect of the code had two different, but valid, test cases; this demonstrates the ability of the test cases to verify the purpose of the function. Furthermore, the last test case of each triangle detection set of tests was an invalid one to prove that the code truly and effectively evaluates the inputs. Overall, we knew this would

```

HW01.py M X tim.py
HW01.py > TestTriangles > test_invalidTriangle
48 self.assertEqual(BuggyTriangle(3, 4, 5).right_check())
49 self.assertTrue(BuggyTriangle(8, 8, math.sqrt(128)).right_check())
50
51 def test_equilateralTriangles(self):
52     """ test equilateral triangle detection """
53     self.assertEqual(BuggyTriangle(7, 7, 7).classify_triangle(), "Equilateral Triangle")
54     self.assertEqual(BuggyTriangle(5, 5, 5).classify_triangle(), "Equilateral Triangle")
55     self.assertNotEqual(BuggyTriangle(1, 2, 3).classify_triangle(), "Equilateral Triangle")
56
57 def test_isoscelesTriangles(self):
58     """ test isosceles triangle detection """
59     self.assertEqual(BuggyTriangle(4, 4, 6).classify_triangle(), "Isosceles Triangle")
60     self.assertEqual(BuggyTriangle(2, 3, 2).classify_triangle(), "Isosceles Triangle")
61     self.assertNotEqual(BuggyTriangle(6, 5, 3).classify_triangle(), "Isosceles Triangle")
62
63 def test_scaleneTriangles(self):
64     """ test scalene triangle detection """
65     self.assertEqual(BuggyTriangle(4, 5, 6).classify_triangle(), "Scalene Triangle")
66     self.assertEqual(BuggyTriangle(4, 7, 5).classify_triangle(), "Scalene Triangle")
67     #This is the one i changed
68     self.assertNotEqual(BuggyTriangle(5, 4, 3).classify_triangle(), "Scalene Triangle")
69
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Alan Atrach\OneDrive - stevens.edu\SSW 567\SSW567_HW01> & "C:\Users\Alan Atrach\AppData\Local\Microsoft\Windows\
/HW01.py"
test_equilateralTriangles (__main__.TestTriangles.test_equilateralTriangles)
test equilateral triangle detection ... ok
test_invalidTriangle (__main__.TestTriangles.test_invalidTriangle)
test invalid triangle detection ... ok
test_isoscelesTriangles (__main__.TestTriangles.test_isoscelesTriangles)
test isosceles triangle detection ... ok
test_rightTriangle (__main__.TestTriangles.test_rightTriangle)
test right triangle detection ... ok
test_scaleneTriangles (__main__.TestTriangles.test_scaleneTriangles)
test scalene triangle detection ... FAIL

=====
FAIL: test_scaleneTriangles (__main__.TestTriangles.test_scaleneTriangles)
test scalene triangle detection
=====
Traceback (most recent call last):
  File "c:\Users\Alan Atrach\OneDrive - stevens.edu\SSW 567\SSW567_HW01\HW01.py", line 68, in test_scaleneTriangles
    self.assertNotEqual(BuggyTriangle(5, 4, 3).classify_triangle(), "Scalene Triangle")
AssertionError: 'Scalene Triangle' == 'Scalene Triangle'

=====
Ran 5 tests in 0.002s

FAILED (failures=1)
PS C:\Users\Alan Atrach\OneDrive - stevens.edu\SSW 567\SSW567_HW01>

```

This is a screenshot of the code with an intentional testing error to confirm that the unittest is working correctly.

```
HW01.py M X tim.py
HW01.py > TestTriangles > test_scaleneTriangles
48 self.assertEqual(BuggyTriangle(3, 4, 5).right_check())
49 self.assertTrue(BuggyTriangle(8, 8, math.sqrt(128)).right_check())
50
51 def test_equilateralTriangles(self):
52     """ test equilateral triangle detection """
53     self.assertEqual(BuggyTriangle(7, 7, 7).classify_triangle(), "Equilateral Triangle")
54     self.assertEqual(BuggyTriangle(5, 5, 5).classify_triangle(), "Equilateral Triangle")
55     self.assertNotEqual(BuggyTriangle(1, 2, 3).classify_triangle(), "Equilateral Triangle")
56
57 def test_isoscelesTriangles(self):
58     """ test isosceles triangle detection """
59     self.assertEqual(BuggyTriangle(4, 4, 6).classify_triangle(), "Isosceles Triangle")
60     self.assertEqual(BuggyTriangle(2, 3, 2).classify_triangle(), "Isosceles Triangle")
61     self.assertNotEqual(BuggyTriangle(6, 5, 3).classify_triangle(), "Isosceles Triangle")
62
63 def test_scaleneTriangles(self):
64     """ test scalene triangle detection """
65     self.assertEqual(BuggyTriangle(4, 5, 6).classify_triangle(), "Scalene Triangle")
66     self.assertEqual(BuggyTriangle(4, 7, 5).classify_triangle(), "Scalene Triangle")
67     #This is the one i changed
68     self.assertNotEqual(BuggyTriangle(5, 5, 3).classify_triangle(), "Scalene Triangle")
69
70 def test_invalidTriangle(self):
71     """ test invalid triangle detection """
72     self.assertEqual(BuggyTriangle(9, 4, 4).classify_triangle(), "Invalid Triangle")
73     self.assertEqual(BuggyTriangle(2, 7, 2).classify_triangle(), "Invalid Triangle")
74     self.assertNotEqual(BuggyTriangle(3, 4, 5).classify_triangle(), "Invalid Triangle")
75
76
77 if __name__ == '__main__':
78     unittest.main(exit=False, verbosity=2)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\Alan Atrach\OneDrive - stevens.edu\SSW 567\SSW567_HW01> & "C:/Users/Alan Atrach/AppData/Local/Microsoft/
/HW01.py"
test_equilateralTriangles (_main__.TestTriangles.test_equilateralTriangles)
test equilateral triangle detection ... ok
test_invalidTriangle (_main__.TestTriangles.test_invalidTriangle)
test invalid triangle detection ... ok
test_isoscelesTriangles (_main__.TestTriangles.test_isoscelesTriangles)
test isosceles triangle detection ... ok
test_rightTriangle (_main__.TestTriangles.test_rightTriangle)
test right triangle detection ... ok
test_scaleneTriangles (_main__.TestTriangles.test_scaleneTriangles)
test scalene triangle detection ... ok

-----
Ran 5 tests in 0.001s

OK
○ PS C:\Users\Alan Atrach\OneDrive - stevens.edu\SSW 567\SSW567_HW01>
```

These are the test cases in working order.