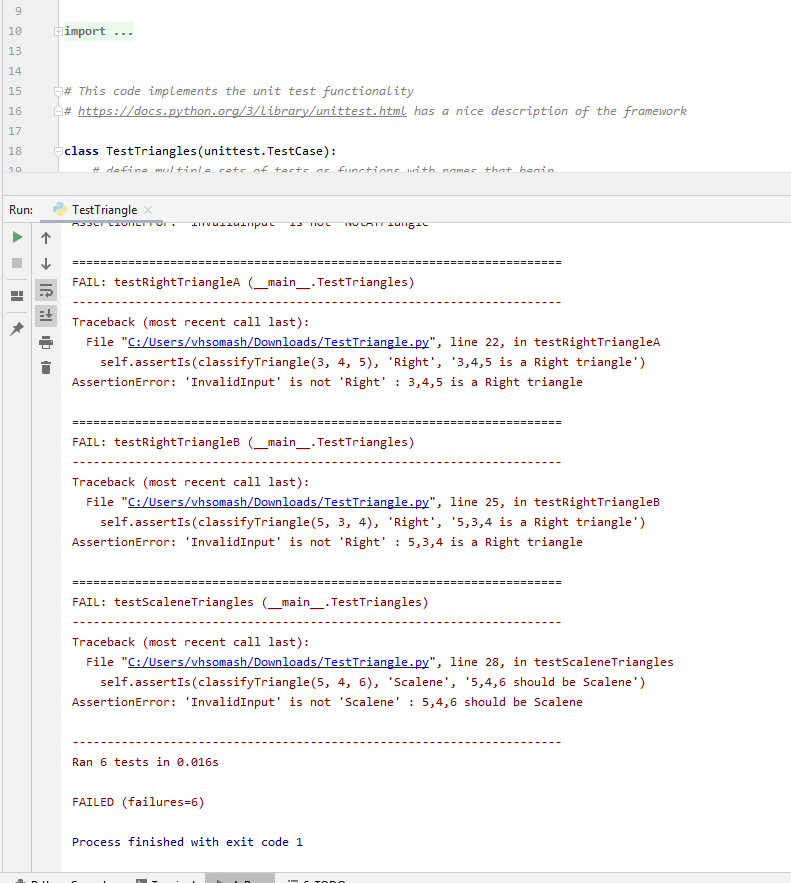
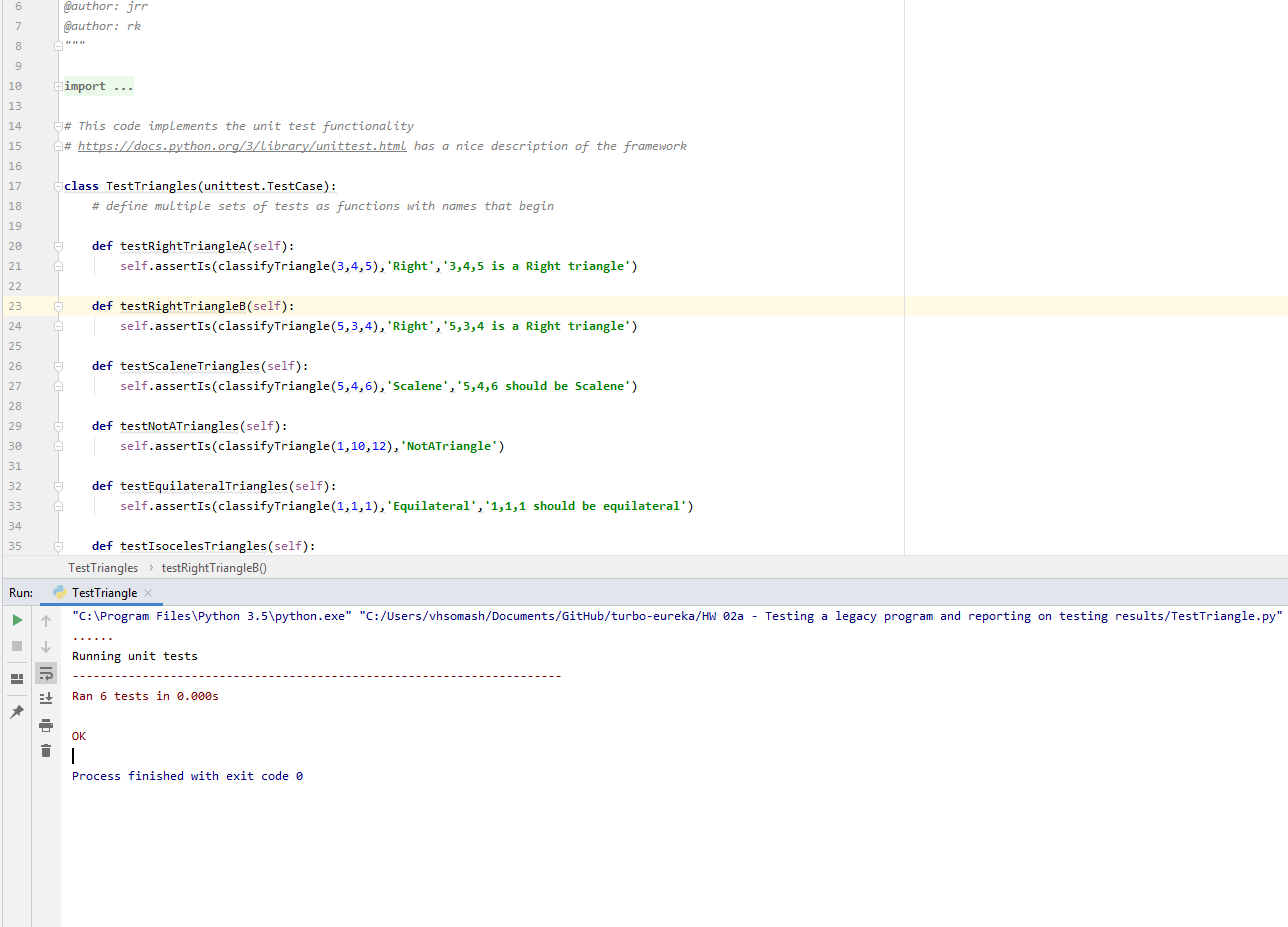
Following screenshot shows test failure when changes made to TestTriangle.py and Traingle.py is untouched.

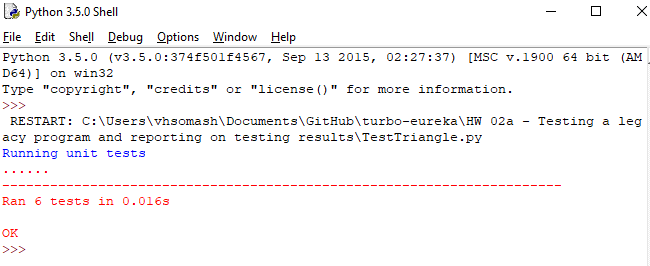


**Test Report of failed execution:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Input** | **Expected Results** | **Actual Result** | **Pass or Fail** |
| TestTriangle.py:32 (TestTriangles.testEquilateralTriangles) | 1,1,1 | Equilateral | InvalidInput | Fail |
| TestTriangle.py:35 (TestTriangles.testIsocelesTriangles) | 3,4,4 | Isoceles | InvalidInput | Fail |
| TestTriangle.py:29 (TestTriangles.testNotATriangles) | 1,10,12 | NotATriangle | InvalidInput | Fail |
| TestTriangle.py:20 (TestTriangles.testRightTriangleA) | 3,4,5 | Right | InvalidInput | Fail |
| TestTriangle.py:23 (TestTriangles.testRightTriangleB) | 5,3,4 | Right | InvalidInput | Fail |
| TestTriangle.py:26 (TestTriangles.testScaleneTriangles) | 5,4,6 | Scalene | InvalidInput | Fail |

Following screenshot shows test success when changes made to both files TestTriangle.py and Traingle.py





**Test Report of Successful execution:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Input** | **Expected Results** | **Actual Result** | **Pass or Fail** |
| (TestTriangles.testEquilateralTriangles) | 1,1,1 | Equilateral | Equilateral | Pass |
| (TestTriangles.testIsocelesTriangles) | 3,4,4 | Isoceles | Isoceles | Pass |
| (TestTriangles.testNotATriangles) | 1,10,12 | NotATriangle | NotATriangle | Pass |
| (TestTriangles.testRightTriangleA) | 3,4,5 | Right | Right | Pass |
| (TestTriangles.testRightTriangleB) | 5,3,4 | Right | Right | Pass |
| (TestTriangles.testScaleneTriangles) | 5,4,6 | Scalene | Scalene | Pass |

**Test Summary:**

As per the code provided in testtriangle.py, only one set of input is tested for each triangle type. This strategy is not sufficient to perform all scenarios including robustness testing.

First assignment file FindTriangleTypes.py has been tested with different types of parameters to show pass or fail for each type of triangle.

For example:

# Test with different parameters. Input different values

class TestEquilateralTriangle(unittest.TestCase):

def test\_all\_sides\_are\_equal(self):

self.assertIs(equilateral([2, 2, 2]), True)

def test\_all\_zero\_sides\_is\_not\_a\_triangle(self):

self.assertIs(equilateral([0, 0, 0]), False)

def test\_third\_triangle\_inequality\_violation(self):

self.assertIs(isosceles([3, 1, 1]), False)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test Run 1 | Test Run2 | Test Run 3 |
|
|
| Tests Planned | 6 | 6 | 6 |
|
|
| Tests Executed | 6 | 6 | 6 |
|
|
| Tests Passed | 0 | 6 | 6 |
|
|
| Defects Found | 6 | 0 | 0 |
|
|
| Defects Fixed | 0 | 0 | 0 |
|
|