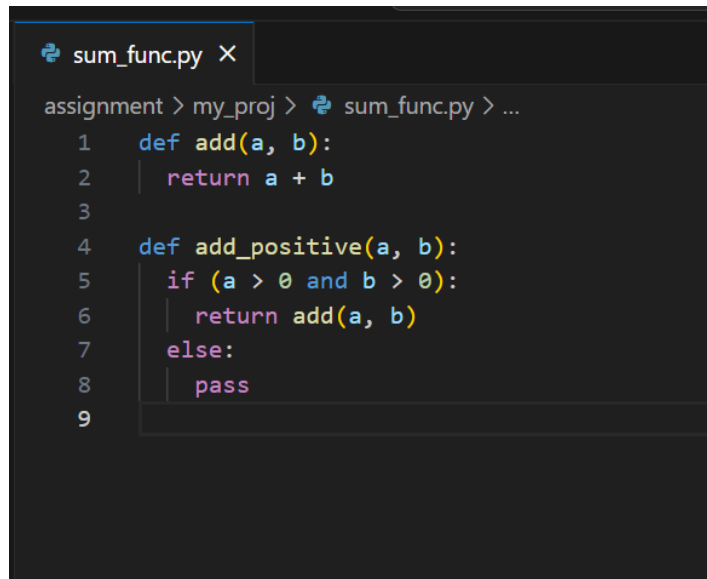


ASSIGNMENT 6.2
on
Testing

Submitted by:
Haseebullah Shaikh (2303.KHI.DEG.015)
Faiza Gulzar Ahmed (2303.KHI.DEG.001)
and
Aqsa Tauheed (2303.KHI.DEG.001)

Dated: 23rd May 2023

Given add functions

A screenshot of a code editor with a dark background. At the top, a tab is labeled 'sum_func.py' with a close button. Below the tab, the file path 'assignment > my_proj > sum_func.py > ...' is visible. The code contains two functions. The first function, 'def add(a, b):', is on line 1 and returns 'a + b' on line 2. The second function, 'def add_positive(a, b):', starts on line 4. It has an 'if' statement on line 5: 'if (a > 0 and b > 0):'. Inside the 'if' block, line 6 returns 'add(a, b)'. Line 7 is 'else:', and line 8 is 'pass'. Line 9 is empty.

```
1 def add(a, b):
2     return a + b
3
4 def add_positive(a, b):
5     if (a > 0 and b > 0):
6         return add(a, b)
7     else:
8         pass
9
```

Tests implementation

Tests are implemented in way to achieve 100% code coverage, given tests ensures

- ✓ Each statement in code is executed
- ✓ Conditions works properly which is tested by trying all the condition combinations.
- ✓ 100% code coverage by implementing branch testing, each function is evaluated as true as well as false.

Implementation code

```
sum_func.py  test_sum_func.py X
assignment > my_proj > tests > test_sum_func.py > ...
1  from my_proj.sum_func import add, add_positive
2
3  # Testing add function on positive integers
4  def test_add_2_2_returns_4():
5      assert add(2, 2) == 4
6
7  # Testing add function on zeros
8  def test_add_0_0_returns_0():
9      assert add(0, 0) == 0
10
11 # Testing add function on negative integers
12 def test_add_negative_nums_4_2_returns_0():
13     assert add(-4, -2) == -6
14
15 # Testing add function on one positive and one negative integer
16 def test_add_negative_and_positive_nums_4_2_returns_0():
17     assert add(-4, 2) == -2
18
19
20 # Testing add positive function on positive integers
21 def test_add_positive_2_2_returns_4():
22     assert add_positive(2, 2) == 4
23
24 # Testing add function on one zeros (>0)
25 def test_add_positive_0_0_returns_none():
26     assert add_positive(0, 0) == None
27
28
29 # Testing add positive function on negative integers
30 def test_add_positive_negative_nums_1_2_returns_None():
31     assert add_positive(-1, -2) == None
32
33 # Testing add positive function on positive (a) negative (b > 0) integers
34 def test_add_positive_postivie_negative_nums_1_2_returns_None():
35     assert add_positive(1, -2) == None
36
37 # Testing add positive function on negative (a > 0) positive (b) integers
38 def test_add_positive_negative_positive_nums_1_2_returns_None():
39     assert add_positive(-1, 2) == None
40
```

Testing coverage results

```
my_proj> cd .\tests
● PS E:\My_Career\Xloop\TRAINING\Technical\Git-Assignment-Repo\Haseebullah-Shaikh-2303.KHI.DEG.015-\Module-6\Assignment 6.2\assignment\
my_proj\tests> pytest --cov=my_proj .\test_sum_func.py
===== test session starts =====
platform win32 -- Python 3.9.6, pytest-7.3.1, pluggy-1.0.0
rootdir: E:\My_Career\Xloop\TRAINING\Technical\Git-Assignment-Repo\Haseebullah-Shaikh-2303.KHI.DEG.015-\Module-6\Assignment 6.2\assignment\my_proj\tests
plugins: dash-2.9.3, cov-4.0.0
collected 9 items

test_sum_func.py ..... [100%]

----- coverage: platform win32, python 3.9.6-final-0 -----
Name
      Stmts   Miss  Cover
-----
E:\My_Career\Xloop\TRAINING\Technical\Git-Assignment-Repo\Haseebullah-Shaikh-2303.KHI.DEG.015-\Module-6\Assignment 6.2\assignment\my_
proj\__init__.py      0      0  100%
E:\My_Career\Xloop\TRAINING\Technical\Git-Assignment-Repo\Haseebullah-Shaikh-2303.KHI.DEG.015-\Module-6\Assignment 6.2\assignment\my_
proj\sum_func.py      6      0  100%
__init__.py           0      0  100%
test_sum_func.py     19      0  100%
-----
TOTAL
      25      0  100%

===== 9 passed in 0.16s =====
○ PS E:\My_Career\Xloop\TRAINING\Technical\Git-Assignment-Repo\Haseebullah-Shaikh-2303.KHI.DEG.015-\Module-6\Assignment 6.2\assignment\
my_proj\tests>
```

The End 😊