**Software Quality Assurance CSSE 351**

**Spring 2022**

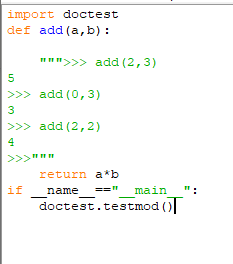
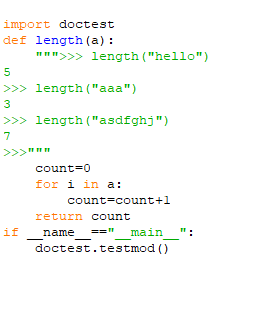
**Assignment 1**

**Haris Lodhi**

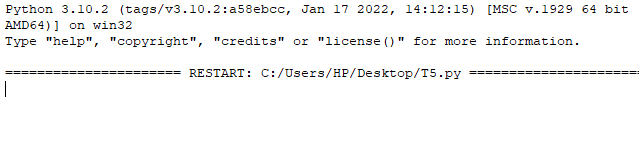
**231487743**

**Test Case1 and 5:**

In test case 1 and 5, I’ve used the technique of doctest testing. Test case 1 is a program which finds the length of a string and task 5 sums two numbers. If we run these tasks and there are no errors, there will be no output in this method. However we can refer to cmd.exe for further details.

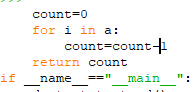


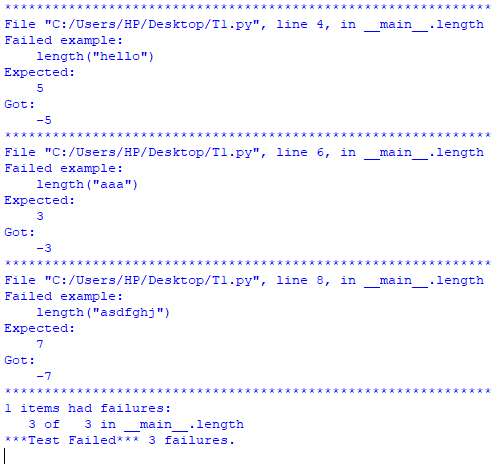
**Output with no errors in code:**



**Output with errors in task 1:**

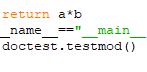
Suppose there is an error in our code, here it is minus instead of +,

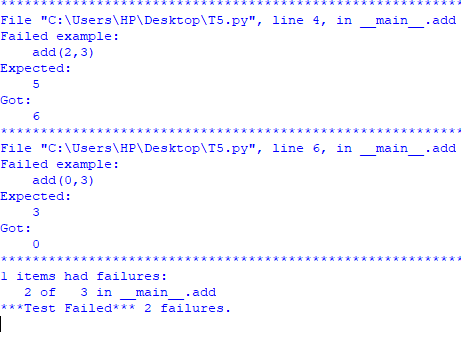
now the output will be



**Output with errors for task 5:**

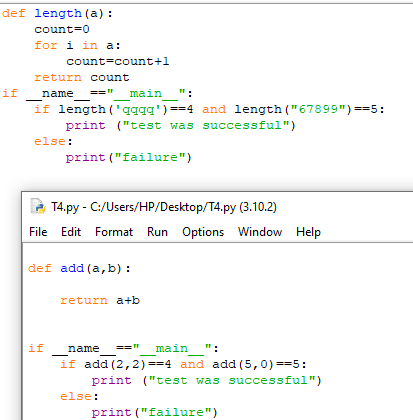
Suppose there is \* instead of + in our code

now the output will be

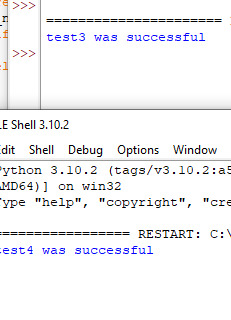


**Test case 3 and 4:**

In test case 3 and 4 name attribute testing, here it is quite simple to detect errors, you yourself make some conditions if the program falls over them it is a PASS else it is a fail.



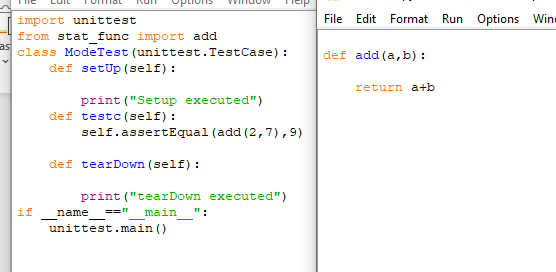
**Output for task 3 and 4:**

both were successful.

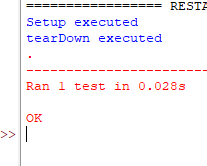
**TEST Case 2:**

in test case 2 we have used unit testing technique, this is also an automated technique for software testing. Here if the software is successful ok is printed else there is a F for failure.

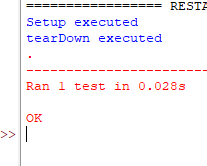
NO error case:



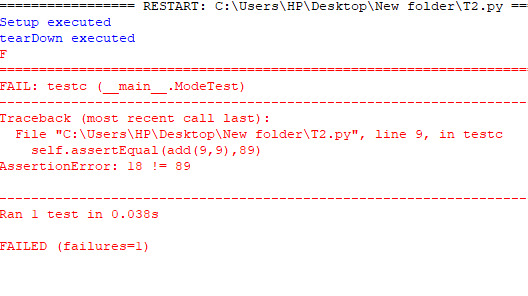
**Output in this case:**



**In case of an error:**

****

**Output in this case:**



**TEST SUITE:**

import doctest

def length(a):

""">>> length("hello")

5

>>> length("aaa")

3

>>> length("asdfghj")

7

>>>"""

count=0

for i in a:

count=count+1

return count

if \_\_name\_\_=="\_\_main\_\_":

doctest.testmod()

def add(a,b):

return a+b

import unittest

from stat\_func import add

class ModeTest(unittest.TestCase):

def setUp(self):

print("Setup executed")

def testc(self):

self.assertEqual(add(9,9),18)

def tearDown(self):

print("tearDown executed")

if \_\_name\_\_=="\_\_main\_\_":

unittest.main()

def length(a):

count=0

for i in a:

count=count+1

return count

if \_\_name\_\_=="\_\_main\_\_":

if length('qqqq')==4 and length("67899")==5:

print ("test3 was successful")

else:

print("failure")

def add(a,b):

return a+b

if \_\_name\_\_=="\_\_main\_\_":

if add(2,2)==4 and add(5,0)==5:

print ("test4 was successful")

else:

print("tesst4 was failure")

import doctest

def add(a,b):

""">>> add(2,3)

5

>>> add(0,3)

3

>>> add(2,2)

4

>>>"""

return a+b

if \_\_name\_\_=="\_\_main\_\_":

doctest.testmod()