*CSCS 351: Software Quality Assurance*

*Assignment 1*

Name: Haris Amin

Roll Number: 231494566

Section: A

# ***Overview:***

* I have made 5 different test cases.
* Doc Testing, Attribute Testing and Unit Testing Techniques have been used to test these codes.

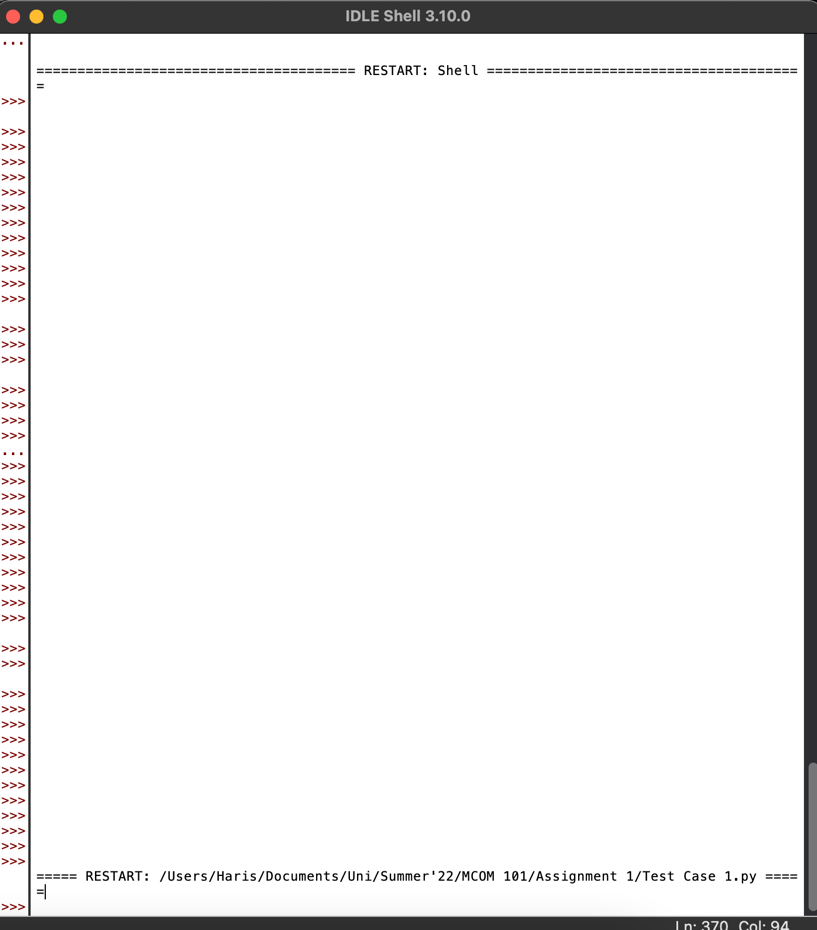
# ***Test Case 1:***

* Below are the screenshots for Test Case 1.
* **Doc Testing Technique** has been used to test this code.
* If the code is written correctly and has no errors, it will run normally and there will be no error just like in the screenshot below. Notice how in the output screenshot, it is shown that the code executed without errors as all the test cases were passed.

Correct Code Screenshot



Correct Code Output Screenshot

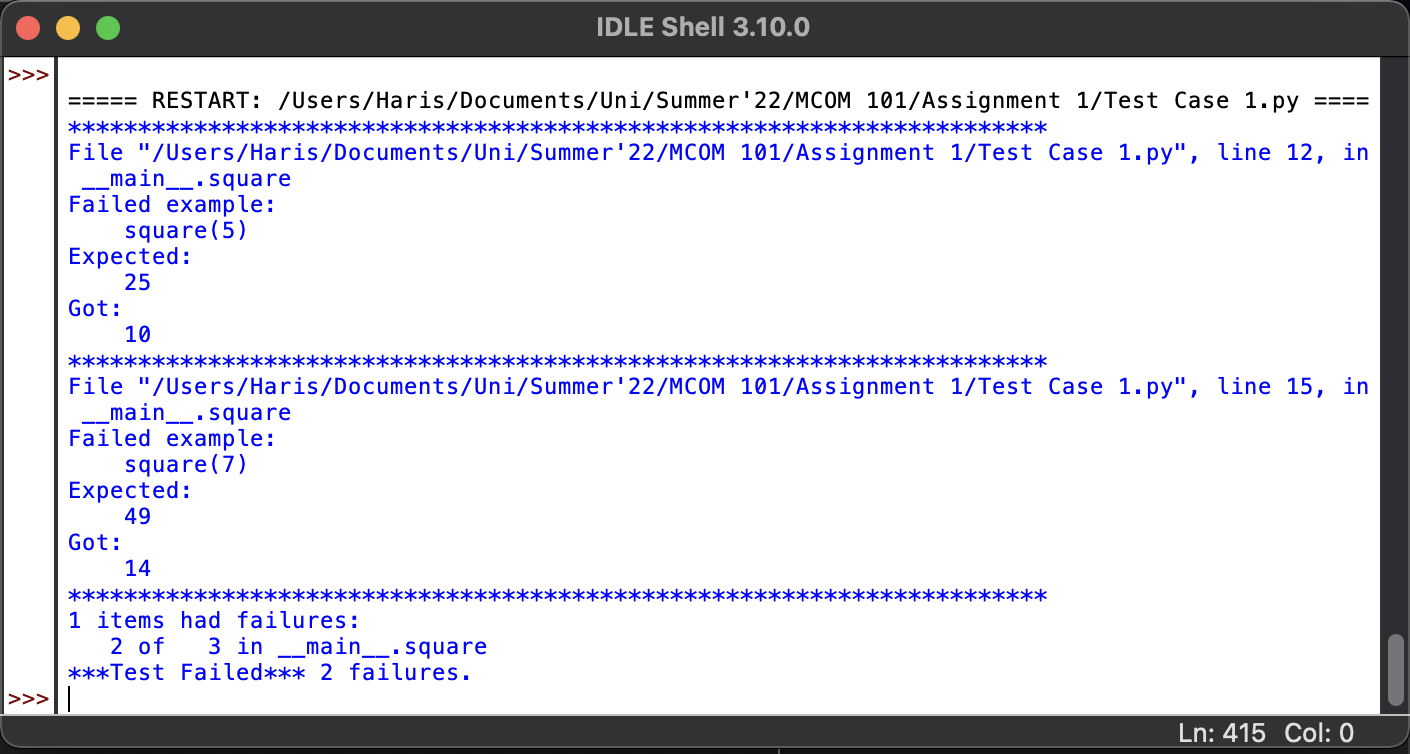


* If there are errors in the code, the test cases would fail. The first test case passes, but the remaining two fail.

Incorrect Code Screenshot



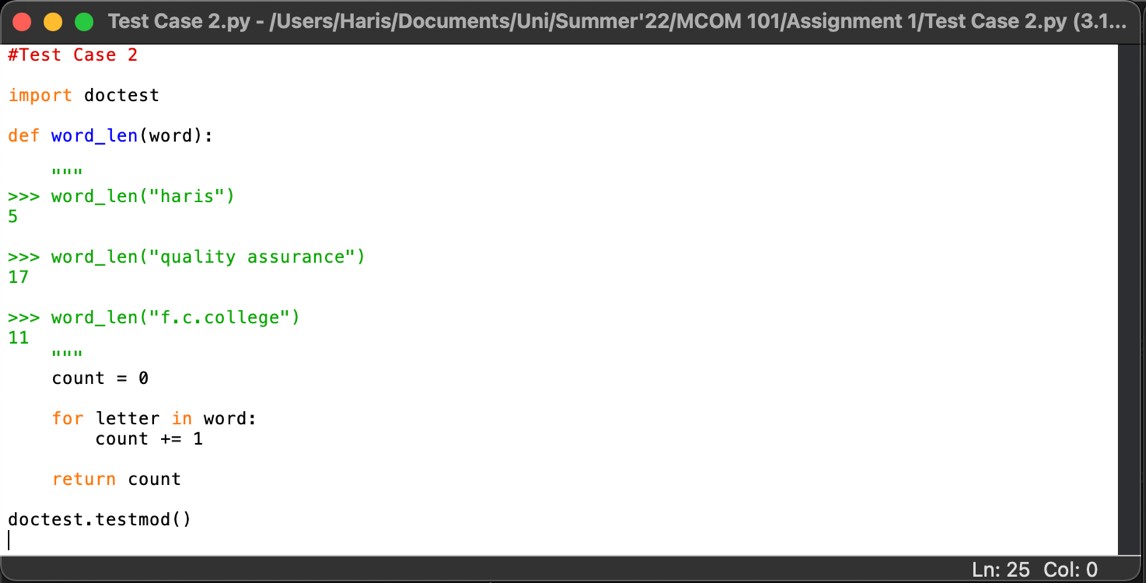
Incorrect Code Output Screenshot



# ***Test Case 2:***

* Below are the screenshots for Test Case 2.
* **Doc Testing Technique** has been used to test this code.
* If the code is written correctly and has no errors, it will run normally and there will be no error just like in the screenshot below. Notice how in the output screenshot, it is shown that the code executed without errors as all the test cases were passed.

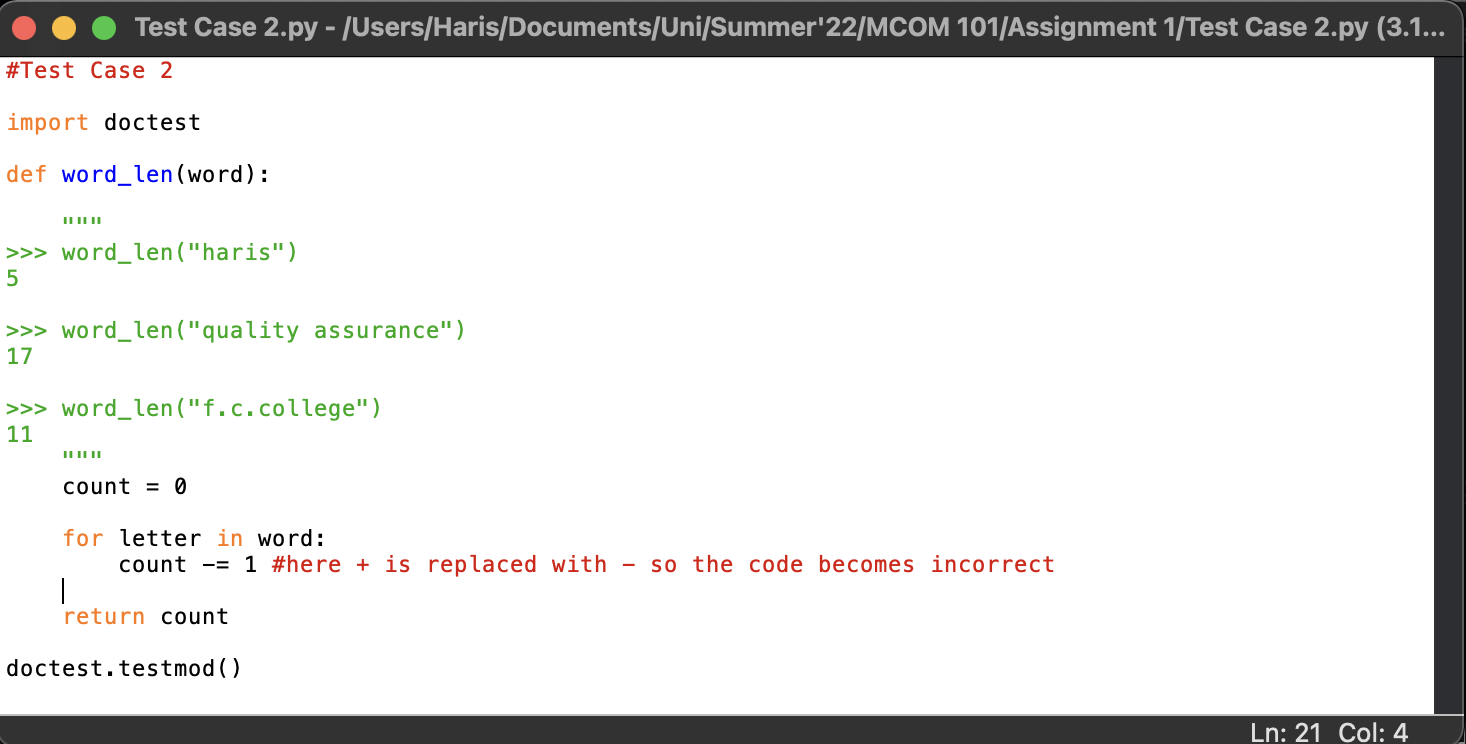
Correct Code Screenshot



Correct Code Output Screenshot



* If there are errors in the code, the test cases would fail.

Incorrect Code Screenshot

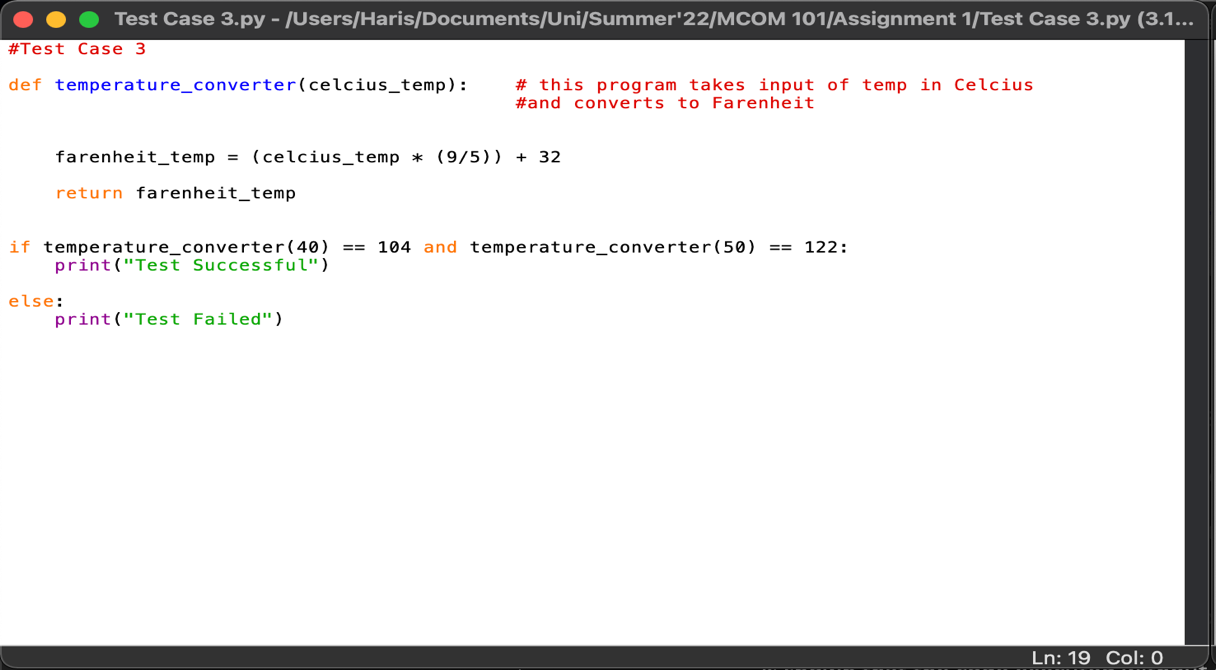
Incorrect Code Output Screenshot



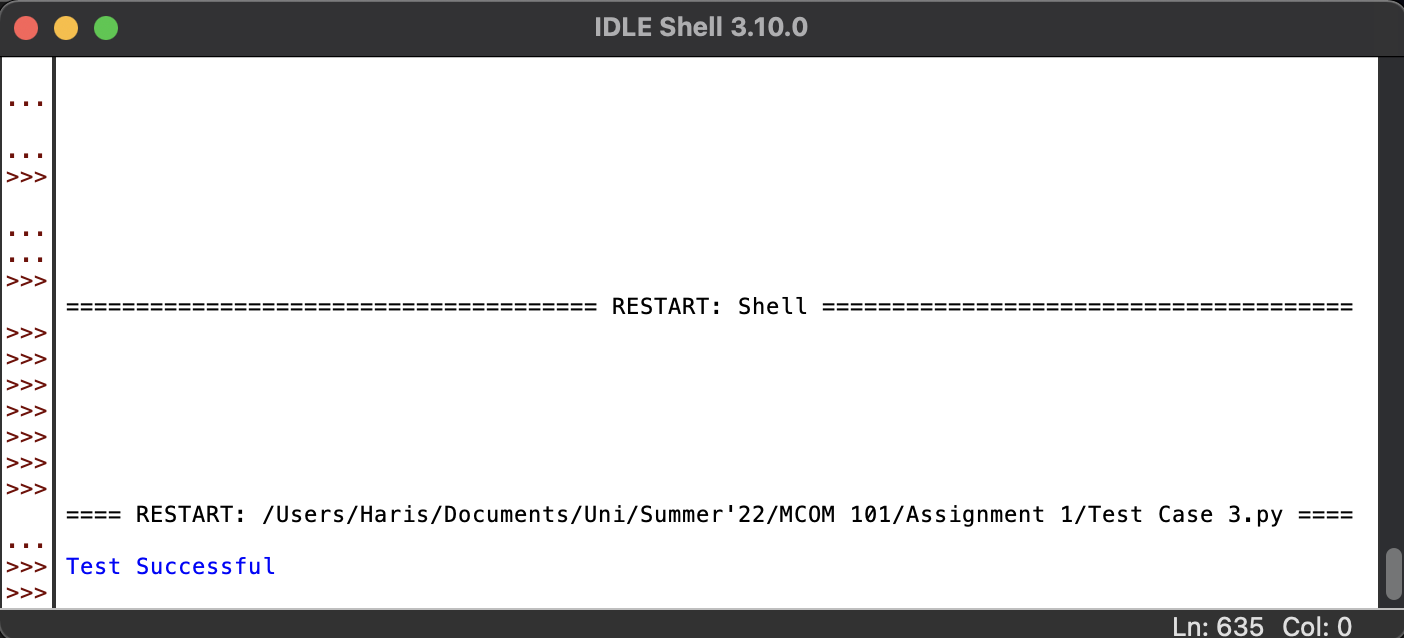
# ***Test Case 3:***

* Below are the screenshots for Test Case 3.
* **Attribute Testing Technique** has been used to test this code.
* In this testing technique, you don’t have to import anything like in Doc Testing. You just have to use if conditions upon which Pass and Fail of your code will be decided.
* If the code is written correctly and has no errors, it will run normally and there will be no error just like in the screenshot below. Notice how in the output screenshot, it is displayed that Test Successful if code is right but if it’s wrong, it says Test Failed.

Correct Code Screenshot

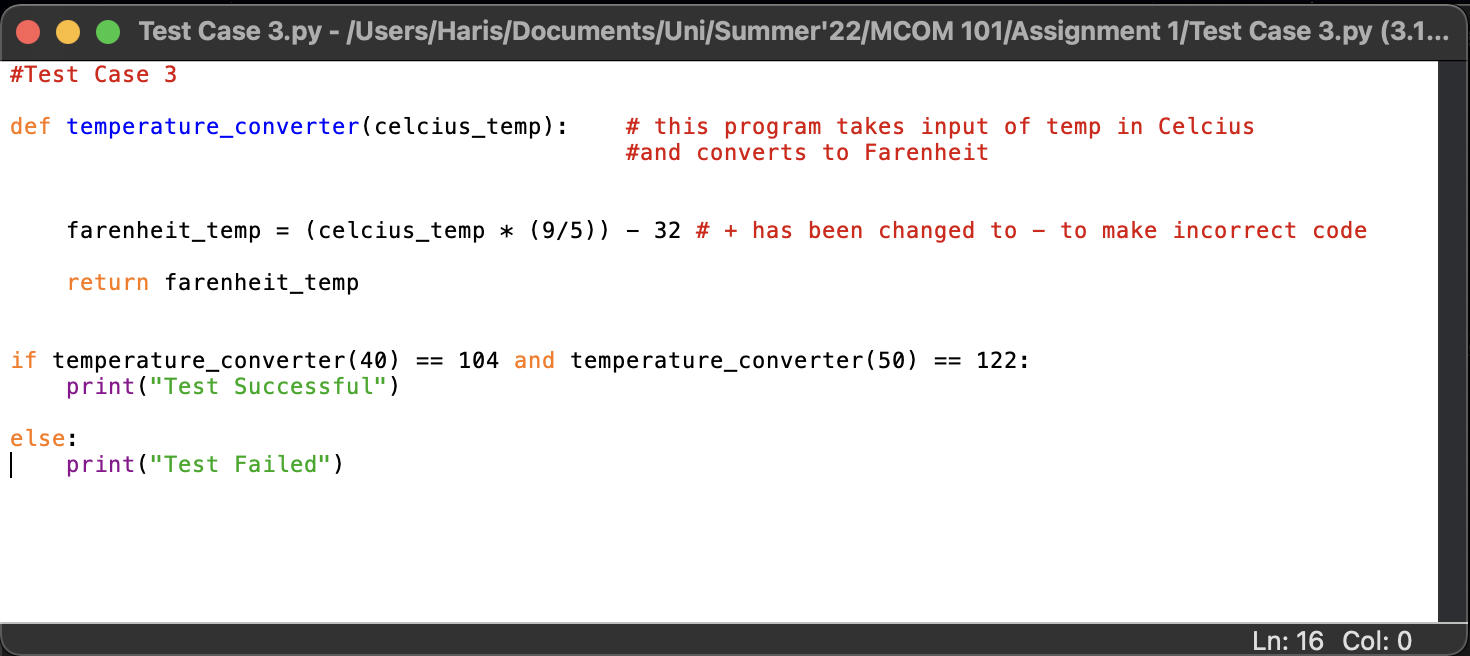


Correct Code Output Screenshot

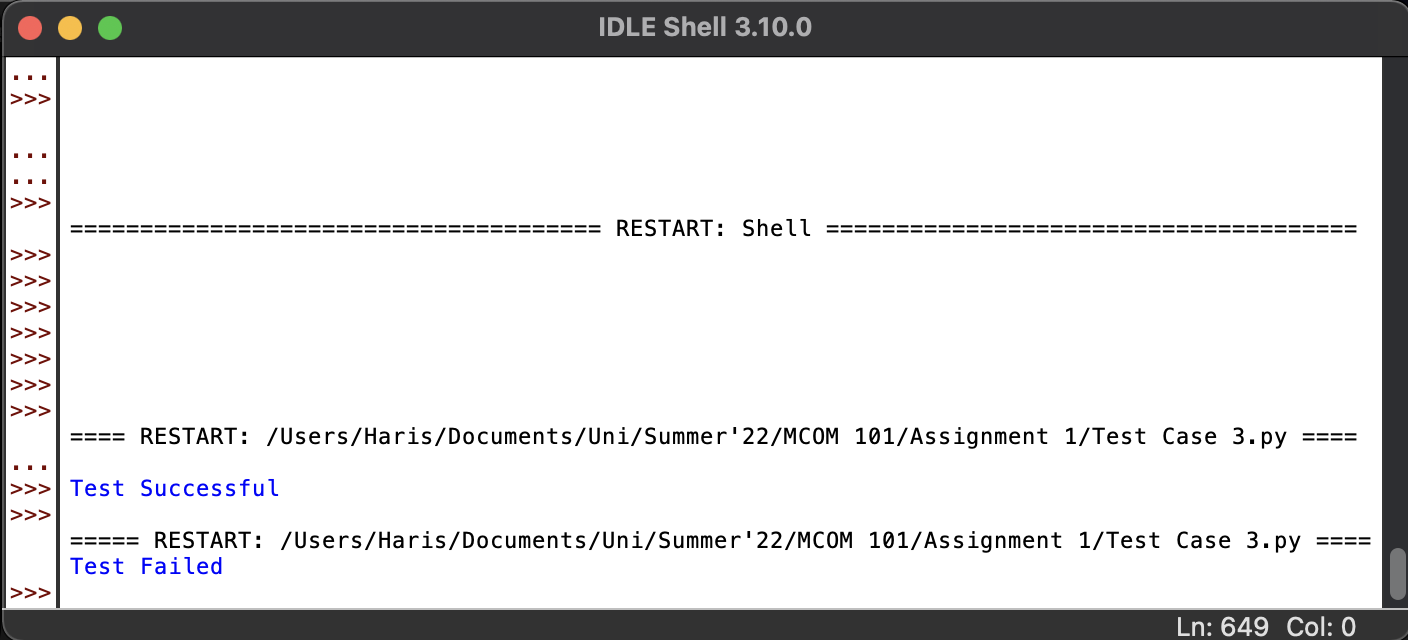


* If there are errors in the code, the test cases would fail, and it will be displayed that the test was failed.

Incorrect Code Screenshot



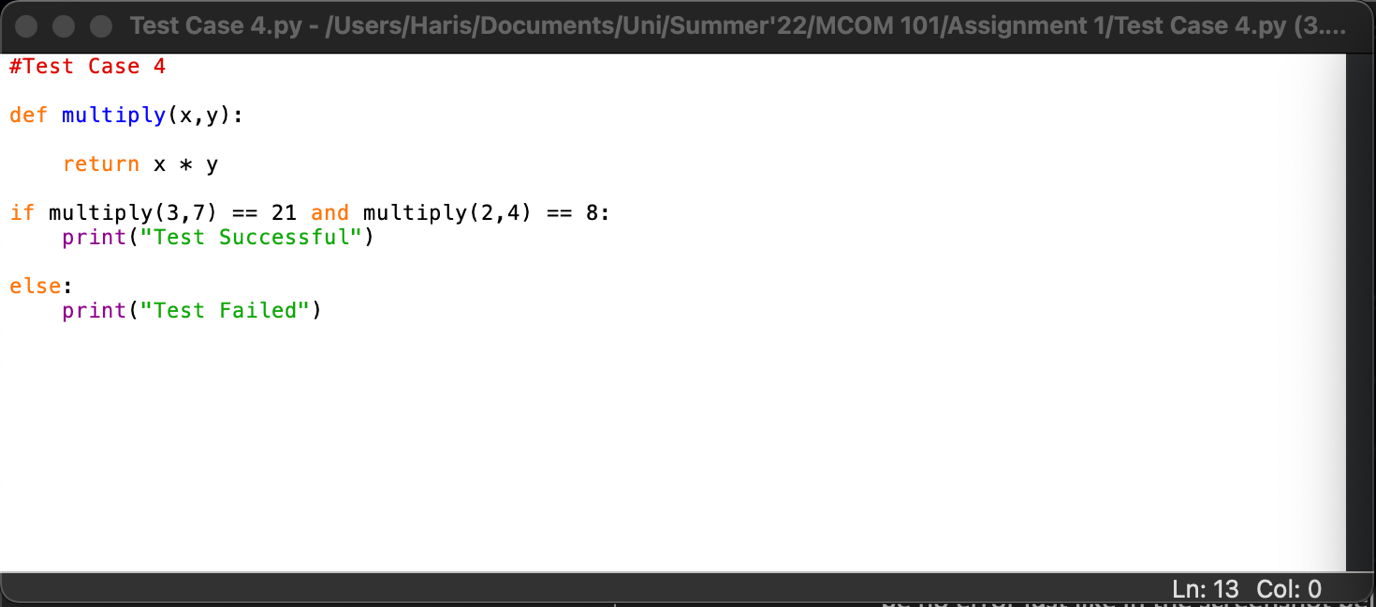
Incorrect Code Output Screenshot



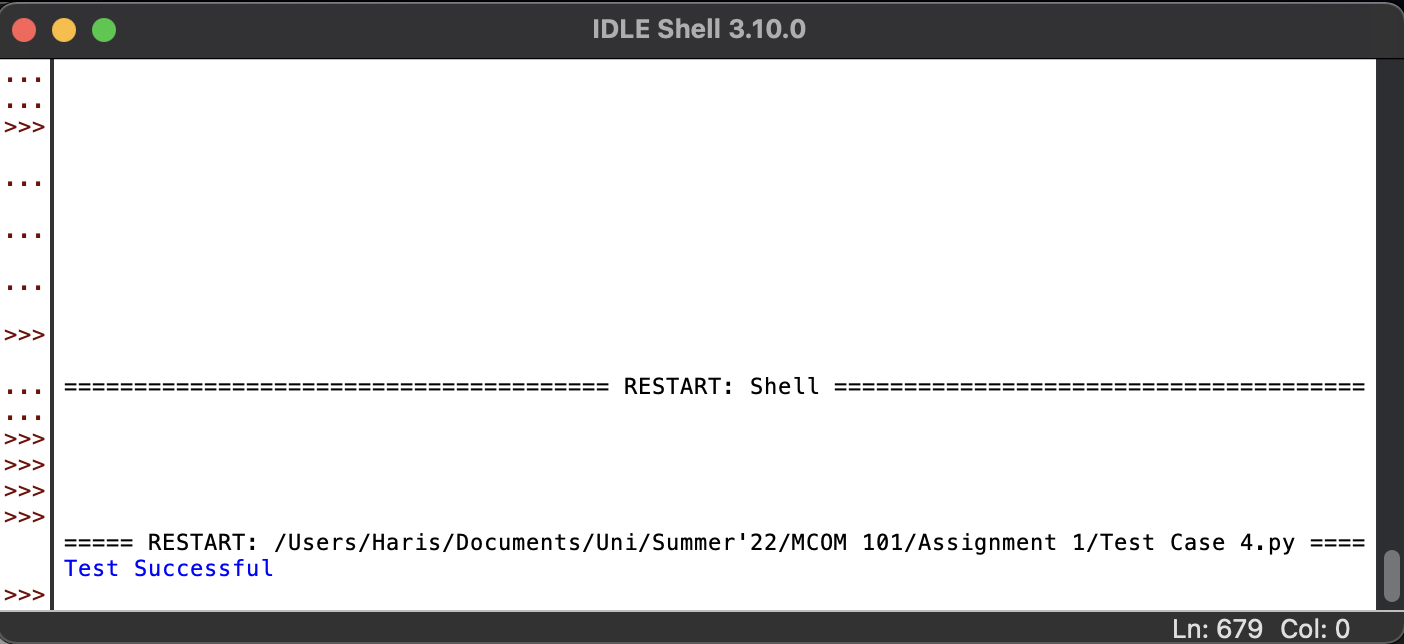
# ***Test Case 4:***

* Below are the screenshots for Test Case 4.
* **Attribute Testing Technique** has been used to test this code.
* In this testing technique, you don’t have to import anything like in Doc Testing. You just have to use if conditions upon which Pass and Fail of your code will be decided.
* If the code is written correctly and has no errors, it will run normally and there will be no error just like in the screenshot below. Notice how in the output screenshot, it is displayed that Test Successful if code is right but if it’s wrong, it says Test Failed.

Correct Code Screenshot

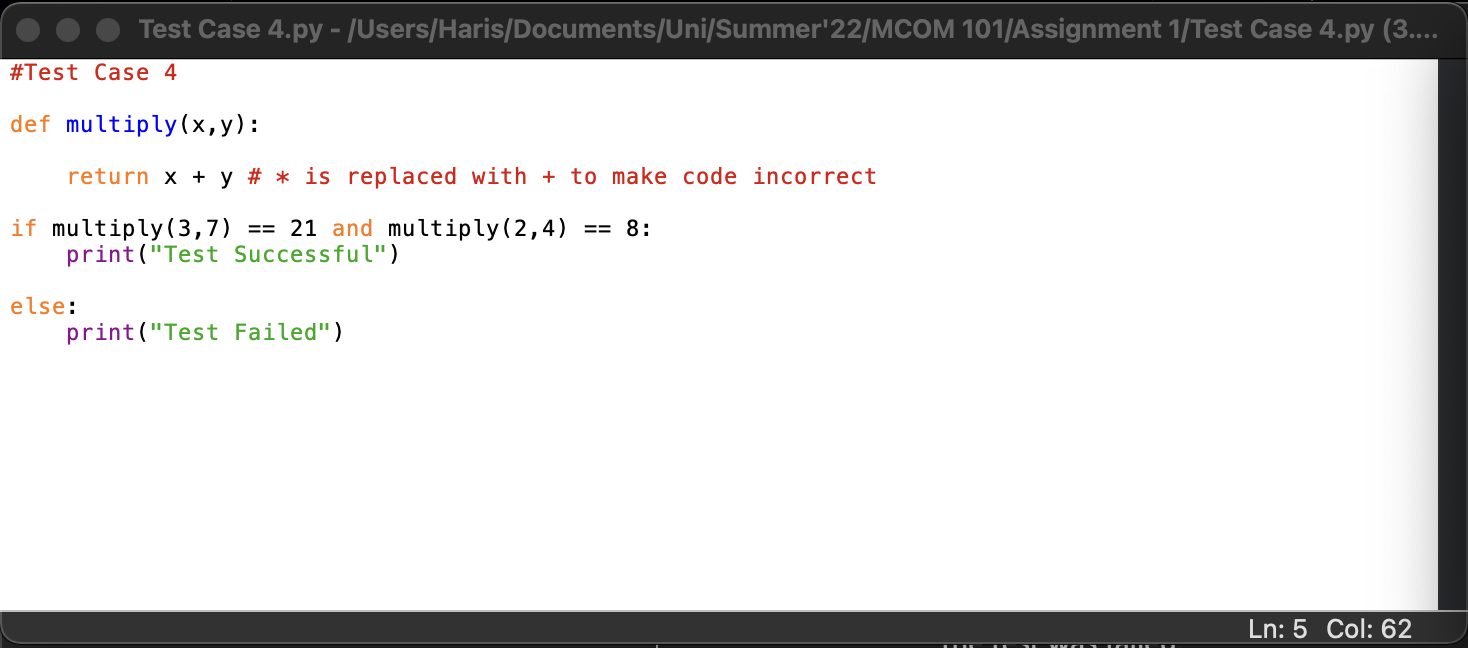


Correct Code Output Screenshot

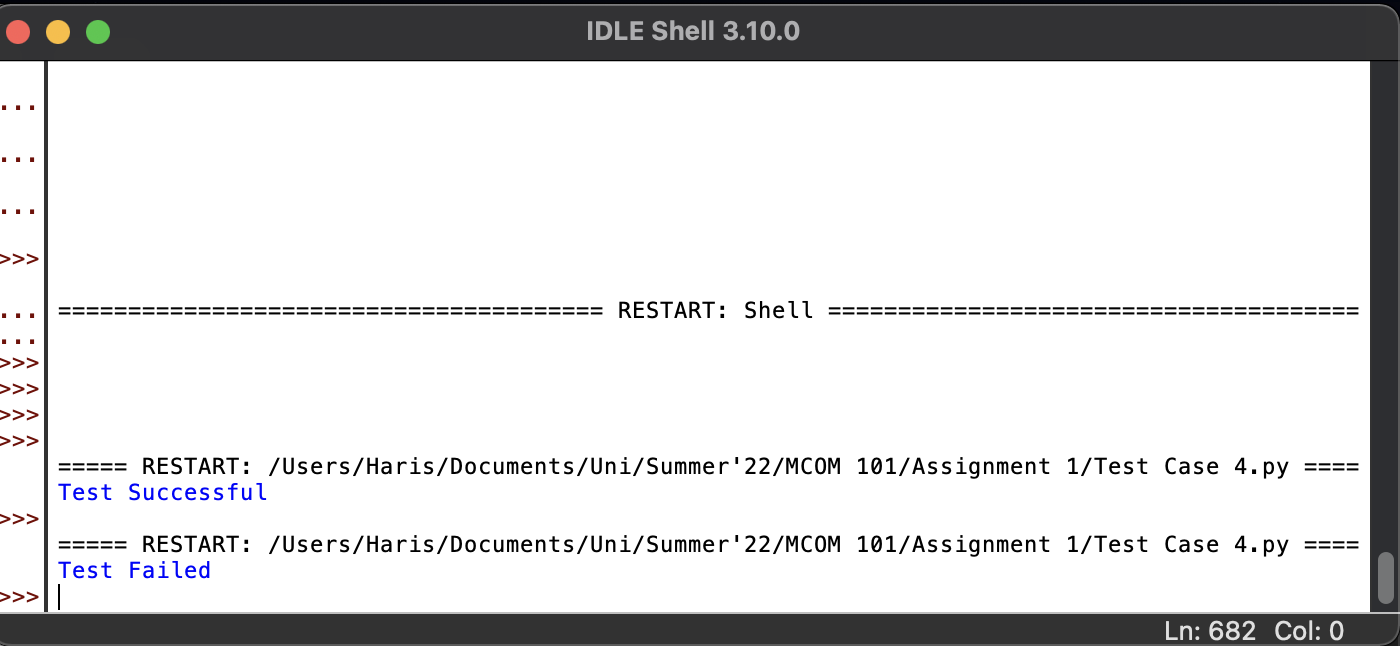


* If there are errors in the code, the test cases would fail, and it will be displayed that the test was failed.

Incorrect Code Screenshot



Incorrect Code Output Screenshot

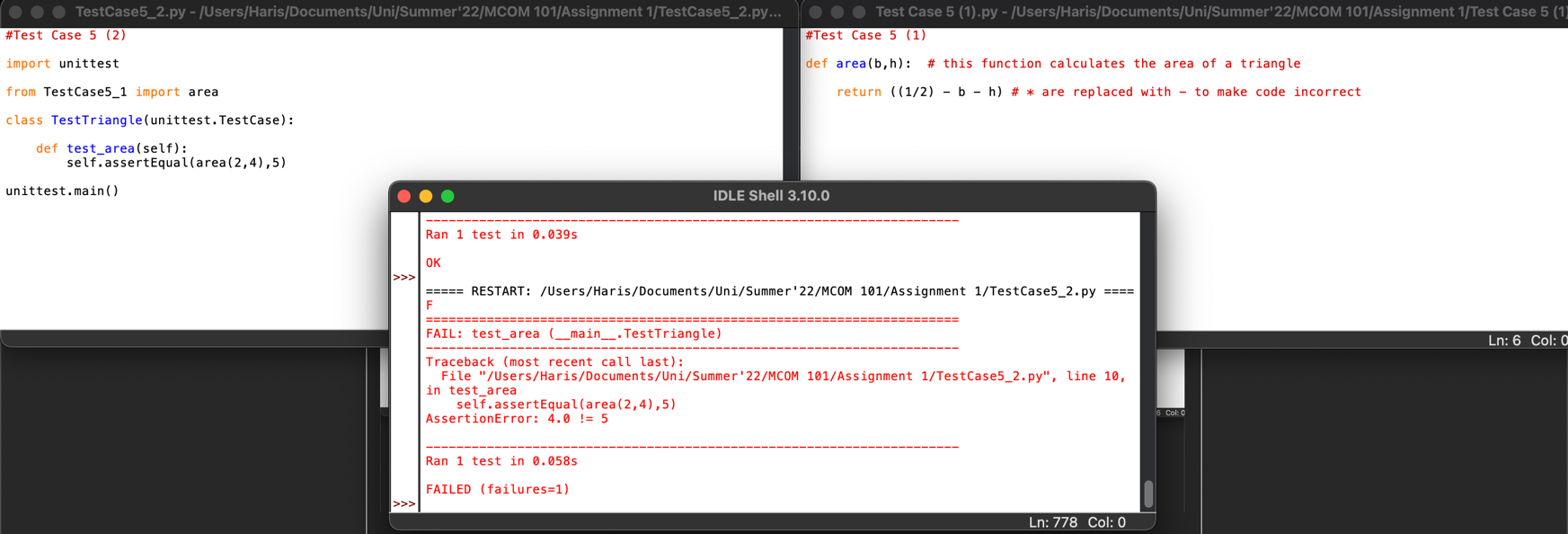


# ***Test Case 5:***

* Below are the screenshots for Test Case 5.
* **Unit Testing Technique** has been used to test this code.
* This is an automated testing technique for software testing.
* If the code is written correctly and has no errors, it will run normally and OK will be printed.
* If the code isn’t written correctly, F will be printed which means the tests Failed.

Correct Code Screenshot



Incorrect Code Screenshot

# ***Test Suite:***

* Below are all the test codes:

#Test Case 1

import doctest

def square(x):

"""

>>> square(2)

4

>>> square(5)

25

>>> square(7)

49

"""

return x+x #here \* is replaced with +, hence code becomes incorrect

doctest.testmod()

#Test Case 2

import doctest

def word\_len(word):

"""

>>> word\_len("haris")

5

>>> word\_len("quality assurance")

17

>>> word\_len("f.c.college")

11

"""

count = 0

for letter in word:

count -= 1 #here + is replaced with - so the code becomes incorrect

return count

doctest.testmod()

#Test Case 3

def temperature\_converter(celcius\_temp): # this program takes input of temp in Celcius

#and converts to Farenheit

farenheit\_temp = (celcius\_temp \* (9/5)) - 32 # + has been changed to - to make incorrect code

return farenheit\_temp

if temperature\_converter(40) == 104 and temperature\_converter(50) == 122:

print("Test Successful")

else:

print("Test Failed")

#Test Case 4

def multiply(x,y):

return x + y # \* is replaced with + to make code incorrect

if multiply(3,7) == 21 and multiply(2,4) == 8:

print("Test Successful")

else:

print("Test Failed")

#Test Case 5 (1)

def area(b,h): # this function calculates the area of a triangle

return ((1/2) \* b \* h)

#Test Case 5 (2)

import unittest

from TestCase5\_1 import area

class TestTriangle(unittest.TestCase):

def test\_area(self):

self.assertEqual(area(2,4),5)

unittest.main()