Course Title:

CSCS 351- Software Quality Assurance

Section: A

Instructor:

Dr Saad Bin Saleem

Instrument:

Assignment 1

Student name with roll number:

Sundas Javaid 19-10685

Semester:

SP 22’

Submission Date:

May 14th, 2022

Introduction

The task mainly focuses on demonstrating the practical usage of unit test framework in Python. The version used for Python code is 3.10.0. This particular programming language has been used due to its simplicity and user-friendliness. There are five testing techniques used which depend upon the nature and functions of the code.

Code

Input:

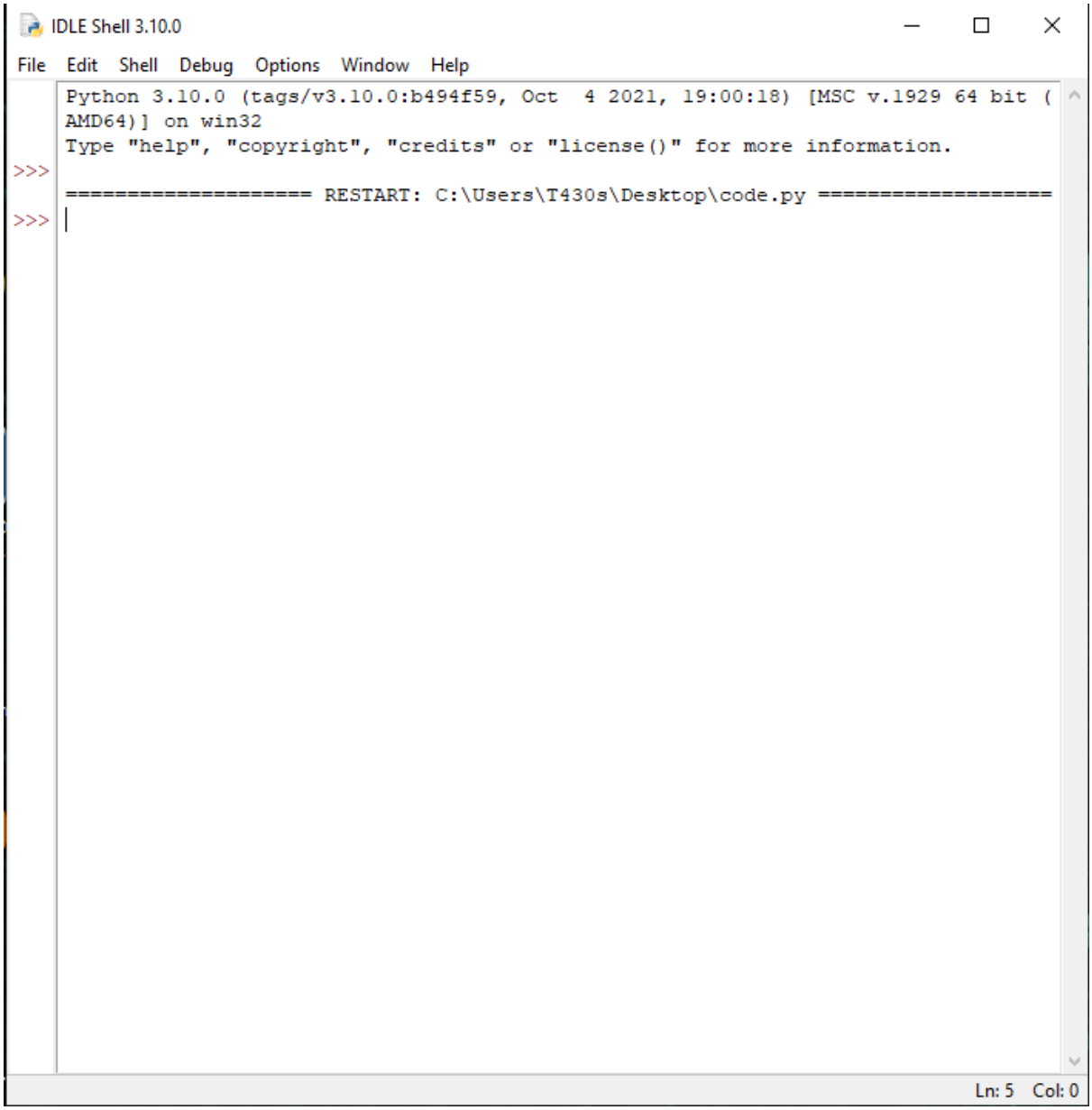
code.py - C:\Users\T430s\Desktop\code.py (3.10.0)

File Edit Format Run Options Window Help

```python
def maximum(a, b, c):
    return max(a,b,c)


def minimum(a, b, c):
    return min(a,b,c)


def equal(a, b):
    if a == b:
        return True

    else:
        False

def positive(a):
    if a > 0:
        return True

    else:
        return False


def zero(b):
    if b == 0:
        return True

    else:
        return False
```

Ln: 1 Col: 0

Output:


IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\T430s\Desktop\code.py ==========
>>>
Ln: 5 Col: 0

Testing code

Input:

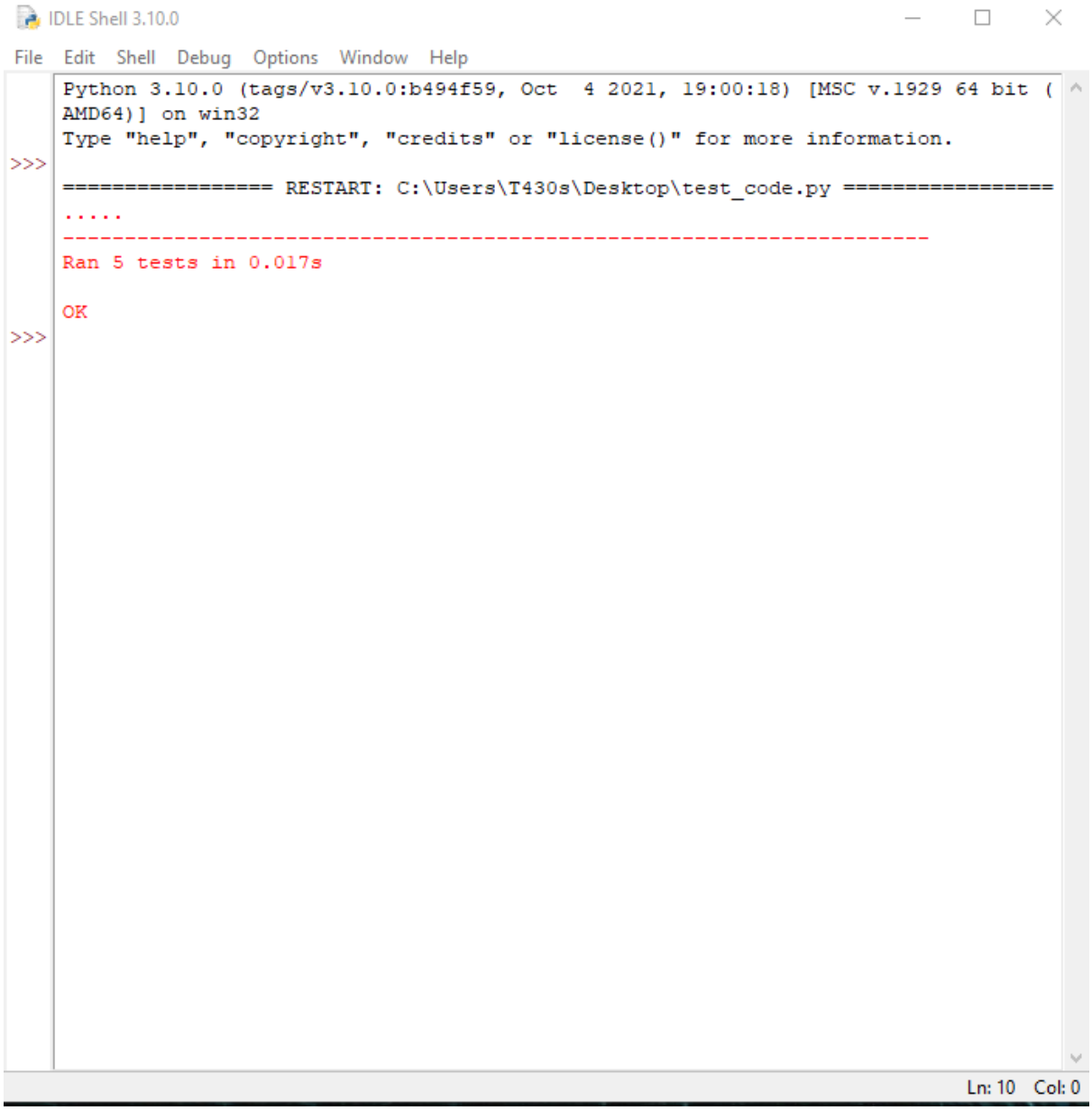
test_code.py - C:\Users\T430s\Desktop\test_code.py (3.10.0)

File Edit Format Run Options Window Help

```python
# Test code in order to test the implemented code

import unittest
import code

class TestCalc(unittest.TestCase):

    def test_max(self):
        self.assertEqual(code.maximum(2, 3, 5), 5)

    def test_min(self):
        self.assertEqual(code.minimum(1, 0, 4), 0)

    def test_equal(self):
        self.assertTrue(code.equal(1, 1))

    def test_postive(self):
        self.assertTrue(code.positive(4))
        self.assertTrue(code.positive(3))

    def test_zero(self):
        self.assertTrue(code.zero(0))

if __name__ == '__main__':
    unittest.main()
```

Ln: 26 Col: 0

Output:



Conclusion:

When we execute a module as a standalone program, the attribute *__name__* will be assigned the string '*__main__*'. 'OK' indicates successful execution of test cases. If anyone of the test cases fail, then Python indicates the failure by showing *AssertionError*.

***