Software Quality Assurance

Name: Tabish Amjad (22-11262)

Section: A

Course: Software Quality Assurance (CSCS 351)

Instructor: Dr. Saad Bin Saleem

What is unit test?

JUnit inspired the unit testing framework, which has a similar taste to prominent unit testing frameworks in other languages. It allows for test automation, the sharing of test setup and shutdown code, the grouping of tests into collections, and test independence from the reporting system.

To accomplish this, unit test uses an object-oriented approach to support a few key concepts:

Test fixture:

A test fixture is a representation of the steps required to conduct one or more tests, as well as any cleanup procedures. This could entail building temporary or proxy databases, directories, or initiating a server process and vice versa.

Test Case:

The specific unit of testing is known as a test case. It looks for a particular response to a specific set of inputs. Test Case is a foundation class provided by unit test that may be used to generate new test cases.

Test Suite:

A test suite is either a collection of test cases or a group of test suites. It's used to group tests that should all be run at the same time.

Test Runner:

A test runner is a component that orchestrates test execution and presents the results to the user. To indicate the results of running the tests, the runner can use a graphical interface, a textual interface, or return a particular value.

- **For This assignment I will be performing testing using PyUnit testing framework and wrote 5 test cases.**

Methods:

Several assert methods are provided by the TestCase class to check for and report failures. Following are the methods that I had used in this assignment:

Method	**Checks that**
assertEqual(a, b)	a == b
assertNotEqual(a, b)	a != b
assertTrue(x)	bool(x) is True
assertGreater(a,b)	a > b
assertLess(a,b)	a < b

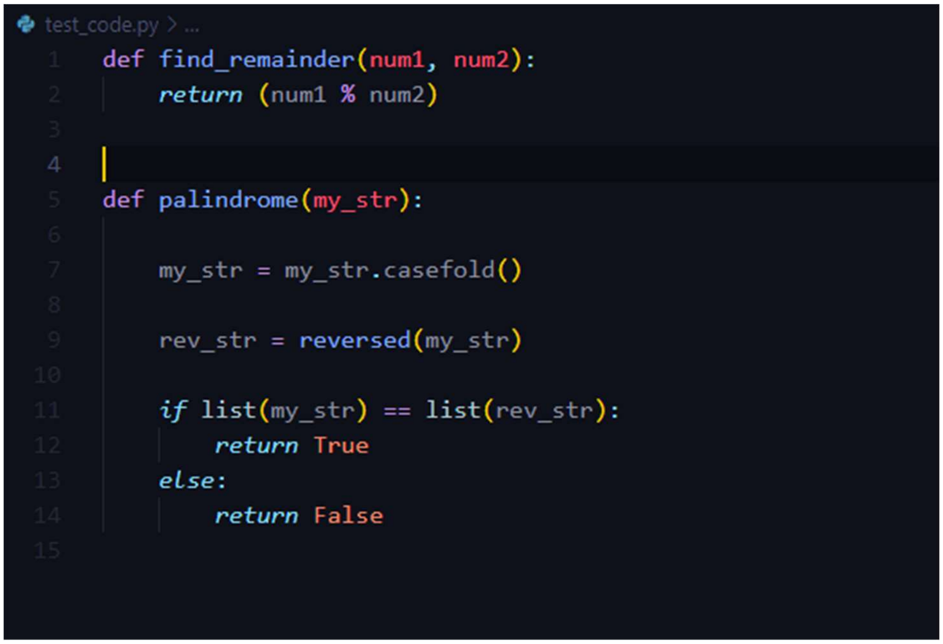
Code Snippet of Functions on which testing will be applied:

I have written down two functions:

- Function to find_remainder (assertEqual(a, b), assertNotEqual(a, b), assertGreater(a,b), assertLess(a,b) will be applied on this function).
- Function to determine whether a string is palindrome or not (assertTrue(x) will be applied on this function).
- For the function of find_remainder, I am taking the example of remainder of i.e., 10 % 2 = 0 and 10 % 3 = 1.
- For the function of palindrome, I am taking string **"acca"** as palindrome.

In these two functions I will perform PyUnit testing.

Test Code

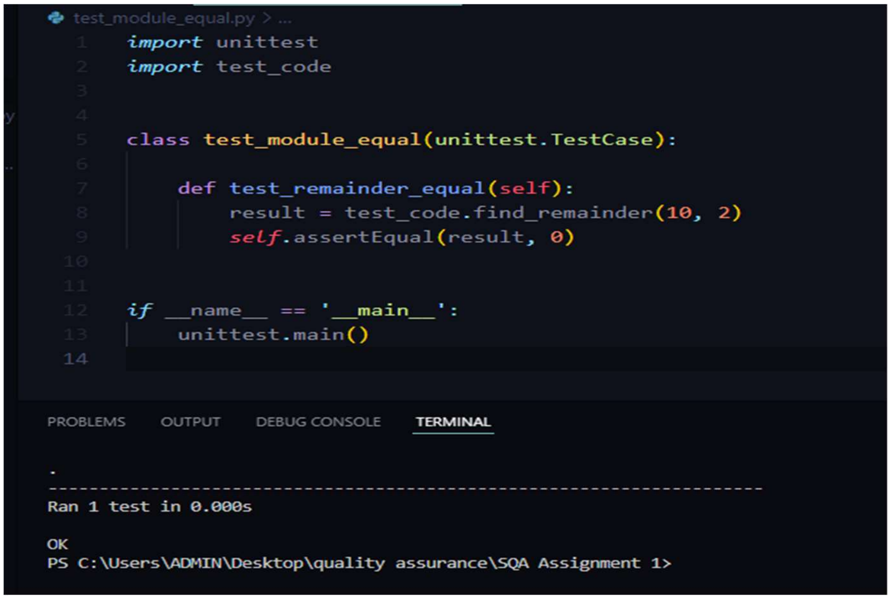


```python
def find_remainder(num1, num2):
    return (num1 % num2)


def palindrome(my_str):

    my_str = my_str.casefold()

    rev_str = reversed(my_str)

    if list(my_str) == list(rev_str):
        return True
    else:
        return False

```

Test Cases:

Following are the Test Cases that I had written for PyUnit testing:

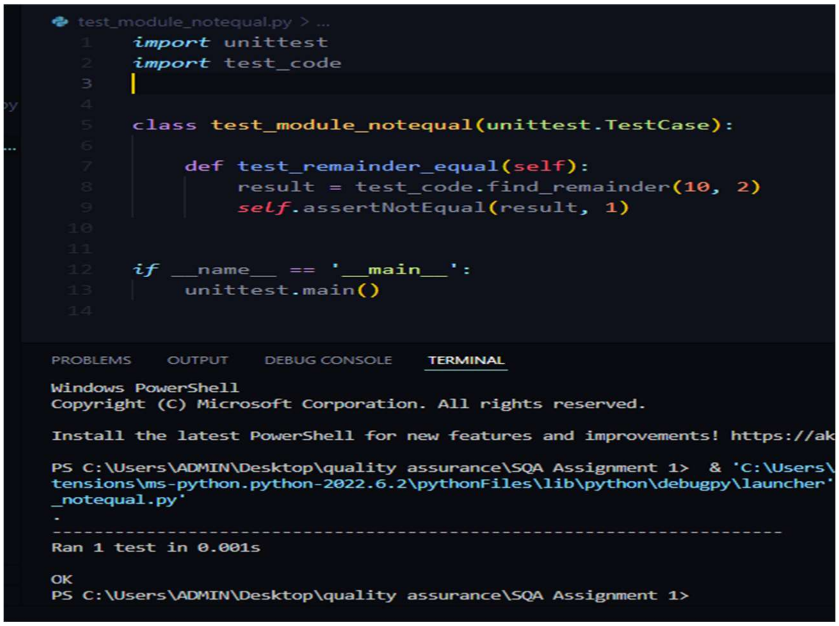
test_module_equal



```python
import unittest
import test_code


class test_module_equal(unittest.TestCase):

    def test_remainder_equal(self):
        result = test_code.find_remainder(10, 2)
        self.assertEqual(result, 0)


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

```

```
.
----------------------------------------------------------------------
Ran 1 test in 0.000s

OK
PS C:\Users\ADMIN\Desktop\quality assurance\SQA Assignment 1>
```

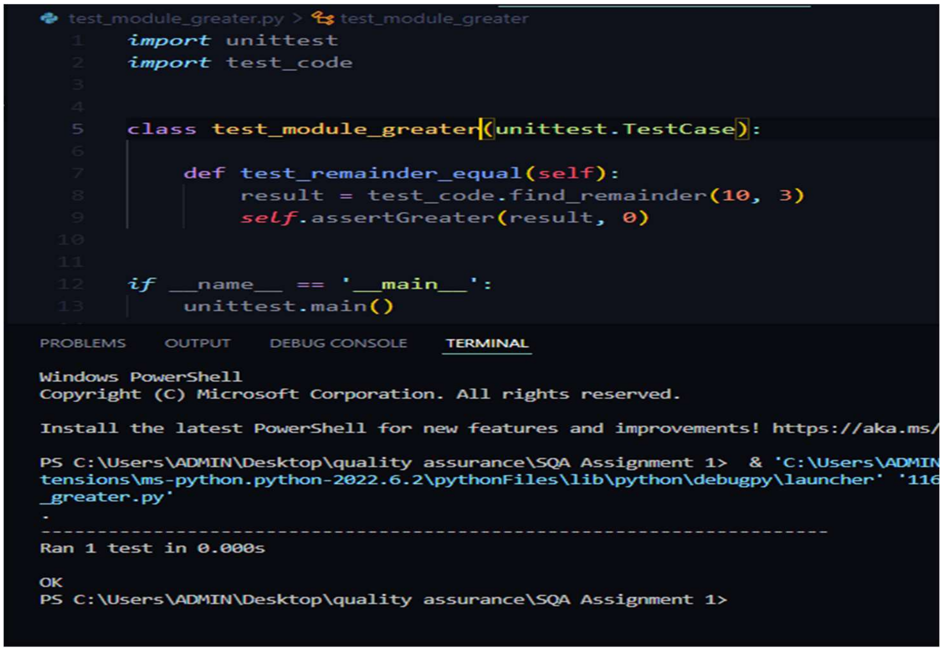
test_module_notequal



```python
import unittest
import test_code


class test_module_notequal(unittest.TestCase):

    def test_remainder_equal(self):
        result = test_code.find_remainder(10, 2)
        self.assertNotEqual(result, 1)


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

```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://ak

PS C:\Users\ADMIN\Desktop\quality assurance\SQA Assignment 1>  & 'C:\Users\
tensions\ms-python.python-2022.6.2\pythonFiles\lib\python\debugpy\launcher'
_notequal.py'
.
----------------------------------------------------------------------
Ran 1 test in 0.001s

OK
PS C:\Users\ADMIN\Desktop\quality assurance\SQA Assignment 1>
```

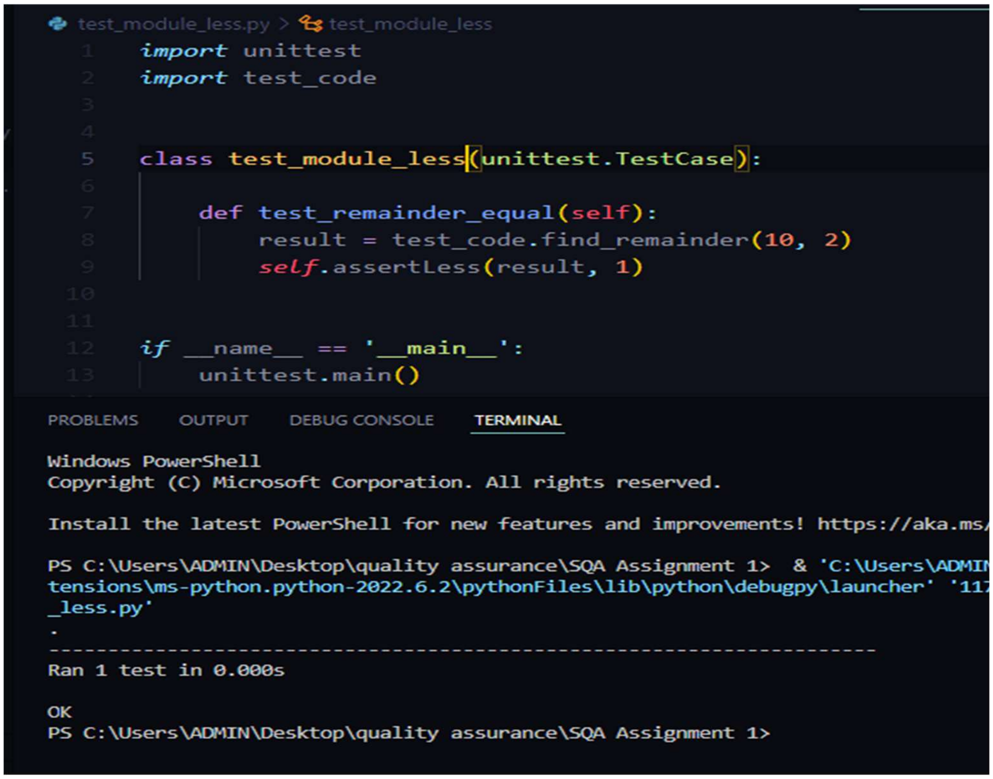
test_module_greater



```python
import unittest
import test_code


class test_module_greater(unittest.TestCase):

    def test_remainder_equal(self):
        result = test_code.find_remainder(10, 3)
        self.assertGreater(result, 0)


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

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/

PS C:\Users\ADMIN\Desktop\quality assurance\SQA Assignment 1>  & 'C:\Users\ADMIN
tensions\ms-python.python-2022.6.2\pythonFiles\lib\python\debugpy\launcher' '116
_greater.py'
.
----------------------------------------------------------------------
Ran 1 test in 0.000s

OK
PS C:\Users\ADMIN\Desktop\quality assurance\SQA Assignment 1>
```

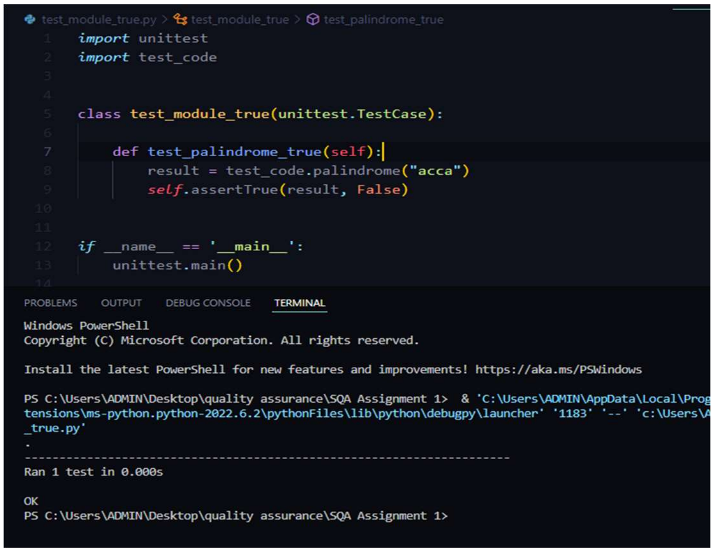
test_module_less



```python
import unittest
import test_code


class test_module_less(unittest.TestCase):

    def test_remainder_equal(self):
        result = test_code.find_remainder(10, 2)
        self.assertLess(result, 1)


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

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/

PS C:\Users\ADMIN\Desktop\quality assurance\SQA Assignment 1>  & 'C:\Users\ADMIN
tensions\ms-python.python-2022.6.2\pythonFiles\lib\python\debugpy\launcher' '117
_less.py'
.
----------------------------------------------------------------------
Ran 1 test in 0.000s

OK
PS C:\Users\ADMIN\Desktop\quality assurance\SQA Assignment 1>
```

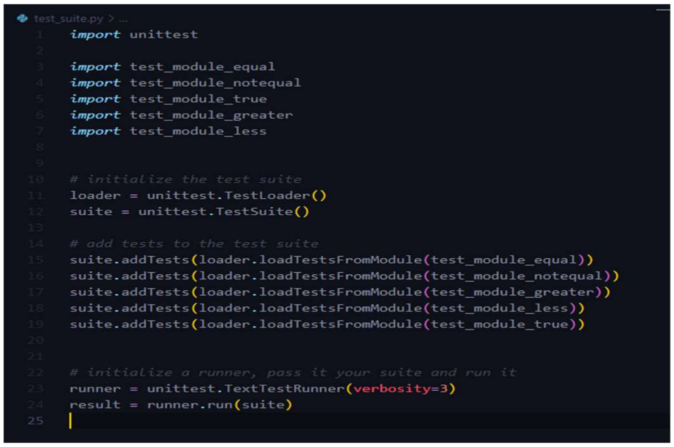
test_module_true


test_module_true.py > test_module_true > test_palindrome_true
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL

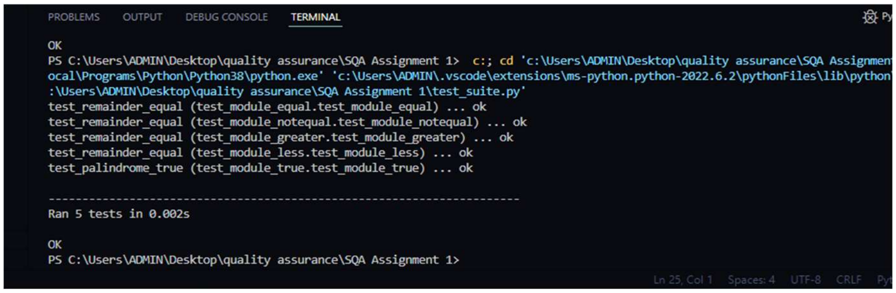
```python
import unittest
import test_code


class test_module_true(unittest.TestCase):

    def test_palindrome_true(self):
        result = test_code.palindrome("acca")
        self.assertTrue(result, False)


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

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\ADMIN\Desktop\quality assurance\SQA Assignment 1>  & 'C:\Users\ADMIN\AppData\Local\Prog
tensions\ms-python.python-2022.6.2\pythonFiles\lib\python\debugpy\launcher' '1183' '--' 'c:\Users\A
_true.py'
.
----------------------------------------------------------------------
Ran 1 test in 0.000s

OK
PS C:\Users\ADMIN\Desktop\quality assurance\SQA Assignment 1>
```

Test Suite for above test cases is given below:



```python
import unittest

import test_module_equal
import test_module_notequal
import test_module_true
import test_module_greater
import test_module_less


# initialize the test suite
loader = unittest.TestLoader()
suite = unittest.TestSuite()

# add tests to the test suite
suite.addTests(loader.loadTestsFromModule(test_module_equal))
suite.addTests(loader.loadTestsFromModule(test_module_notequal))
suite.addTests(loader.loadTestsFromModule(test_module_greater))
suite.addTests(loader.loadTestsFromModule(test_module_less))
suite.addTests(loader.loadTestsFromModule(test_module_true))


# initialize a runner, pass it your suite and run it
runner = unittest.TextTestRunner(verbosity=3)
result = runner.run(suite)

```



```
OK
PS C:\Users\ADMIN\Desktop\quality assurance\SQA Assignment 1>  c:; cd 'c:\Users\ADMIN\Desktop\quality assurance\SQA Assignmen
ocal\Programs\Python\Python38\python.exe' 'c:\Users\ADMIN\.vscode\extensions\ms-python.python-2022.6.2\pythonFiles\lib\python
:\Users\ADMIN\Desktop\quality assurance\SQA Assignment 1\test_suite.py'
test_remainder_equal (test_module_equal.test_module_equal) ... ok
test_remainder_equal (test_module_notequal.test_module_notequal) ... ok
test_remainder_equal (test_module_greater.test_module_greater) ... ok
test_remainder_equal (test_module_less.test_module_less) ... ok
test_palindrome_true (test_module_true.test_module_true) ... ok

----------------------------------------------------------------------
Ran 5 tests in 0.002s

OK
PS C:\Users\ADMIN\Desktop\quality assurance\SQA Assignment 1>
```

Outcome

- All the 5 test cases have sucessfully been run.
- The test suite has successfully been run.
- Python unit testing has successfully been performed.