Misha Akram CS-18118
Iqra Irfan CS-18123

Mujtuba CS-18136
Firdous Riaz CS-18141

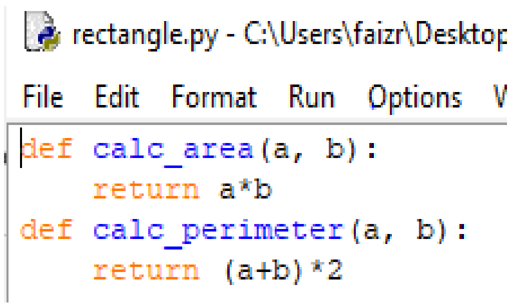
Lab Session 09

Practice framework that facilitates writing test cases for different applications

Exercises:

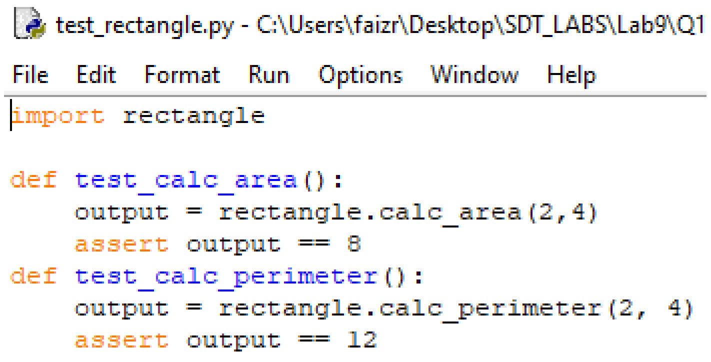
1. Write a simple program to calculate the area and the perimeter of a rectangle in Python and perform testing using Pytest.

rectangle.py:


rectangle.py - C:\Users\faizr\Desktop
File Edit Format Run Options V

```
def calc_area(a, b):
    return a*b
def calc_perimeter(a, b):
    return (a+b)*2
```

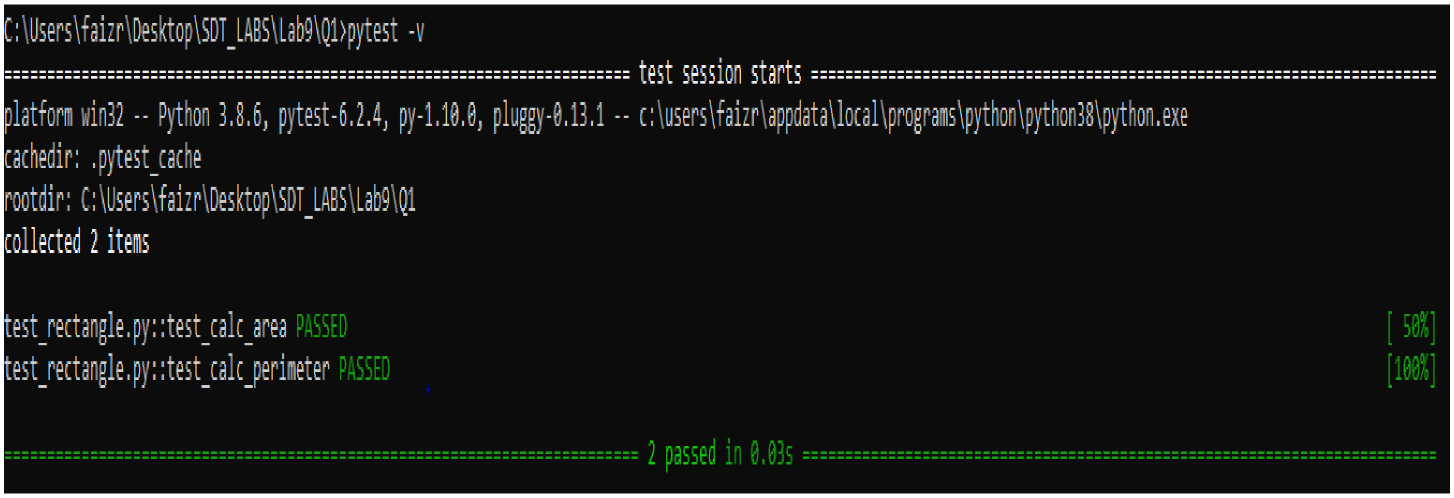
test_rectangle.py:


test_rectangle.py - C:\Users\faizr\Desktop\SDT_LABS\Lab9\Q1
File Edit Format Run Options Window Help

```
import rectangle

def test_calc_area():
    output = rectangle.calc_area(2,4)
    assert output == 8
def test_calc_perimeter():
    output = rectangle.calc_perimeter(2, 4)
    assert output == 12
```

Output:



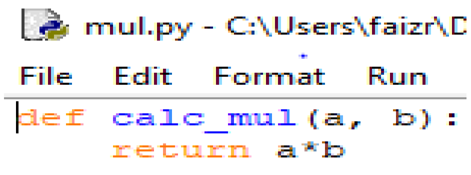
```
C:\Users\faizr\Desktop\SDT_LABS\Lab9\Q1>pytest -v
====================================================================== test session starts ======================================================================
platform win32 -- Python 3.8.6, pytest-6.2.4, py-1.10.0, pluggy-0.13.1 -- c:\users\faizr\appdata\local\programs\python\python38\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\faizr\Desktop\SDT_LABS\Lab9\Q1
collected 2 items

test_rectangle.py::test_calc_area PASSED                                                                                                                   [ 50%]
test_rectangle.py::test_calc_perimeter PASSED                                                                                                              [100%]

======================================================================= 2 passed in 0.03s =======================================================================
```

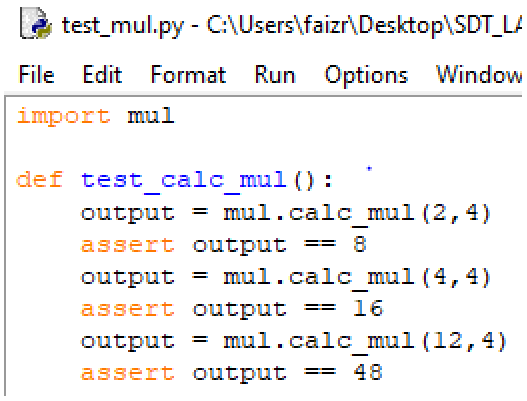
2. Write a program to perform multiplication of numbers and comparing the output(`result`). If the calculation is equal to the result, then, the test case will be passed otherwise not.

mul.py:



```
def calc_mul(a, b):
    return a*b
```

test_mul.py:


test_mul.py - C:\Users\faizr\Desktop\SDT_L
File Edit Format Run Options Window

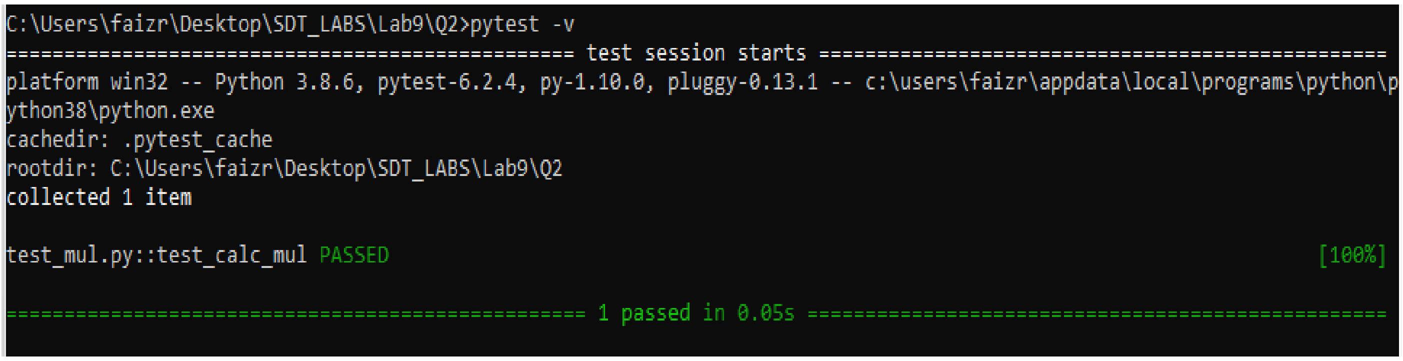
```
import mul

def test_calc_mul():
    output = mul.calc_mul(2,4)
    assert output == 8
    output = mul.calc_mul(4,4)
    assert output == 16
    output = mul.calc_mul(12,4)
    assert output == 48
```

Misha Akram CS-18118
Iqra Irfan CS-18123

Mujtuba CS-18136
Firdous Riaz CS-18141

Output:

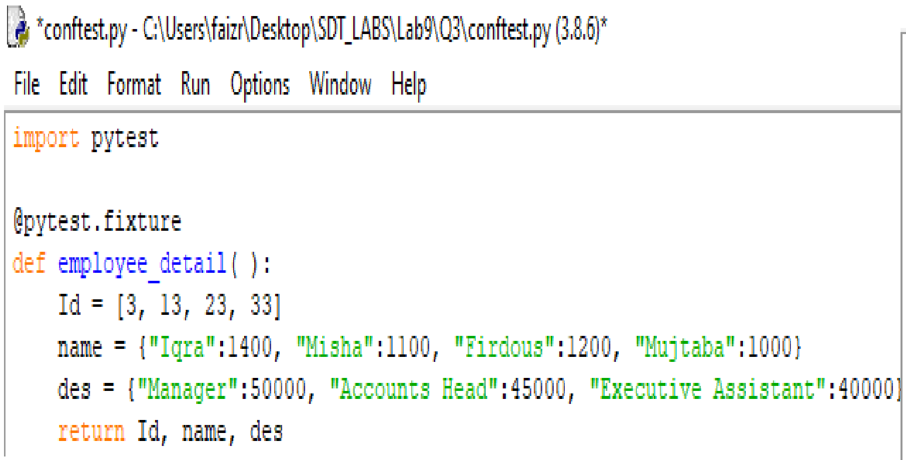


```
C:\Users\faizr\Desktop\SDT_LABS\Lab9\Q2>pytest -v
================================================= test session starts =================================================
platform win32 -- Python 3.8.6, pytest-6.2.4, py-1.10.0, pluggy-0.13.1 -- c:\users\faizr\appdata\local\programs\python\p
ython38\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\faizr\Desktop\SDT_LABS\Lab9\Q2
collected 1 item

test_mul.py::test_calc_mul PASSED                                                                                [100%]

================================================== 1 passed in 0.05s ==================================================
```

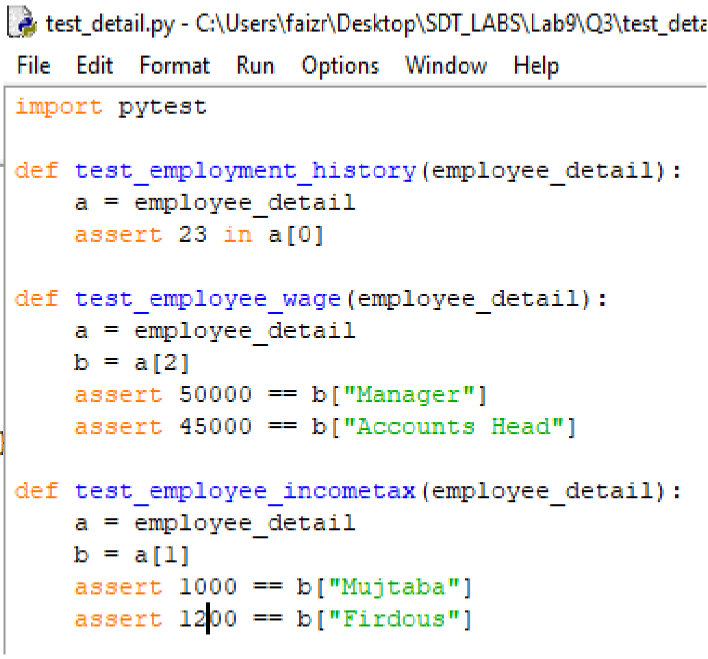
3. Construct a fixture that takes input of name, employee id and designation of the employee. Now use the fixture in the test functions of the following functions:
   - Employment_history
   - Employee_wage
   - Employee_incomeTax



conftest.py - C:\Users\faizr\Desktop\SDT_LABS\Lab9\Q3\conftest.py (3.8.6)

```
import pytest

@pytest.fixture
def employee_detail( ):
    Id = [3, 13, 23, 33]
    name = {"Iqra":1400, "Misha":1100, "Firdous":1200, "Mujtaba":1000}
    des = {"Manager":50000, "Accounts Head":45000, "Executive Assistant":40000]
    return Id, name, des
```



test_detail.py - C:\Users\faizr\Desktop\SDT_LABS\Lab9\Q3\test_deta

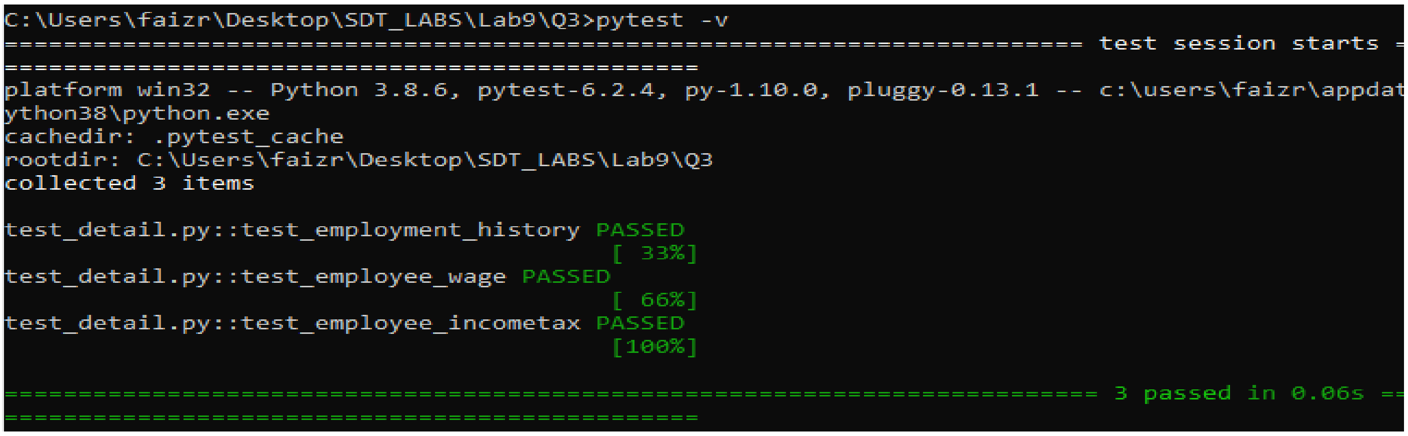
```
import pytest

def test_employment_history(employee_detail):
    a = employee_detail
    assert 23 in a[0]

def test_employee_wage(employee_detail):
    a = employee_detail
    b = a[2]
    assert 50000 == b["Manager"]
    assert 45000 == b["Accounts Head"]

def test_employee_incometax(employee_detail):
    a = employee_detail
    b = a[1]
    assert 1000 == b["Mujtaba"]
    assert 1200 == b["Firdous"]
```

Output:



```
C:\Users\faizr\Desktop\SDT_LABS\Lab9\Q3>pytest -v
========================================================================== test session starts =
===============================================
platform win32 -- Python 3.8.6, pytest-6.2.4, py-1.10.0, pluggy-0.13.1 -- c:\users\faizr\appdat
ython38\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\faizr\Desktop\SDT_LABS\Lab9\Q3
collected 3 items

test_detail.py::test_employment_history PASSED
                                         [ 33%]
test_detail.py::test_employee_wage PASSED
                                         [ 66%]
test_detail.py::test_employee_incometax PASSED
                                         [100%]

=========================================================================== 3 passed in 0.06s ==
===============================================
```