**Test Report**

**TEST CODE:**

# Program make a simple calculator

# This function adds two numbers

def add(x, y):

return x + y

# This function subtracts two numbers

def subtract(x, y):

return x - y

# This function multiplies two numbers

def multiply(x, y):

return x \* y

# This function divides two numbers

def divide(x, y):

if y == 0:

raise ValueError('Can not divide by zero!')

return x / y

def power(x, y):

if x <=0:

raise ValueError('Can not take power')

return x \*\* y

**TEST CASE:**

import unittest

import calc

class Testcalc(unittest.TestCase):

def test\_add(self):

result=calc.add(10,5)

self.assertEqual(calc.add(10, 5), 15)

self.assertEqual(calc.add(-5, 5), 0)

self.assertEqual(calc.add(1.2, 1.3), 2.50)

self.assertEqual(calc.add(-3, -6), -9)

def test\_subtract(self):

self.assertEqual(calc.subtract(10, 5), 5)

self.assertEqual(calc.subtract(-5, 7), -12)

self.assertEqual(calc.subtract(-4,-4), 0)

self.assertEqual(calc.subtract(1.2,0.2), 1)

def test\_multiply(self):

self.assertEqual(calc.multiply(1.2, 1.2), 1.44)

self.assertEqual(calc.multiply(5, 0), 0)

self.assertEqual(calc.multiply(55,2), 110)

self.assertEqual(calc.multiply(-55,-2), 110)

self.assertEqual(calc.multiply(-55,2), -110)

def test\_divide(self):

self.assertEqual(calc.divide(10, 2), 5)

self.assertEqual(calc.divide(-2, 1), -2)

self.assertEqual(calc.divide(-5, -1), 5)

self.assertEqual(calc.divide(5, 2), 2.5)

with self.assertRaises(ValueError):

calc.divide(10, 0)

def test\_power(self):

self.assertEqual(calc.power(10, 2), 100)

self.assertEqual(calc.power(2, -1), 0.5)

self.assertEqual(calc.power(2, 0), 1)

with self.assertRaises(ValueError):

calc.power(-2, 0)

if \_\_name\_\_=='\_\_main\_\_':

unittest.main()

Screen Shot of the Output:



All 5 test passed

we gave 4 different sets of parameters in each test to check the different outputs

1) positive whole numbers

2) one negative number

3) decimal numbers

4) two negative numbers