FileName: Fibonacci.py

The filename extension should be py

def FibRecursion(n):

if n <= 1:

return n

else:

return(FibRecursion(n-1) + FibRecursion(n-2))

nterms = int(input("Enter the terms? ")) # take input from the user

if nterms <= 0: # check if the number is valid

print("Please enter a positive integer")

else:

print("Fibonacci sequence:")

for i in range(nterms):

print(FibRecursion(i))

Steps to run the python script file

1. Go to physical path of the whether the file is saved.

Graphical user interface, application, table

Description automatically generated

1. Type the cmd in the explore of the path

Shape, rectangle

Description automatically generated

Now In the command prompt will have the path of the folder

1. Text

   Description automatically generated

Syntax to run the python file python Fibonacci.py

Test Case Explanation

1. First I created the file with fibonacci\_testcase.py
2. Naming convention should be with testcase along with the original Fibonacci.py
3. First I have included the namespace for the test case import unittest
4. Second I have imported the function name by using the filename of the Fibonacci.py

from fibonacci import FibRecursion

where Fibonacci is the Fibonacci.py filename

where FibRecursion is the function name

-------------------------------------code block Start--------------------------------------------------------

import unittest

from fibonacci import FibRecursion

class MyTest(unittest.TestCase):

def test\_fibonacci(self):

self.assertEqual(FibRecursion(5),7)

def test\_fibonacci\_1(self):

self.assertEqual(FibRecursion(1),1)

def test\_fibonacci\_1(self):

self.assertEqual(FibRecursion("stringvalue"),1)

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

unittest.main()

-------------------------------------code block end--------------------------------------------------------

Text

Description automatically generated

where class MyTest is the testcase classname

def test\_fibonacci(self):

self.assertEqual(FibRecursion(5),7)

By using the assertequal we can check whether the testcase is passed or failed.

1. Positive testcase to be created
2. Negative testcase to be created ( like string value, 0 value, -1)