Function Recursion

Function recursion is calling function itself.
In recursion, calling function and called function both are same. Python also accepts function recursion, which means a defined function can call itself. Recursion is a common mathematical and programming concept. It means that a function calls itself. This has the benefit of meaning that you can loop through data to reach a result.

Syntax:

```
def <function-name>([parameters]):
        statement-1
        statement-2
        function-name() # Recursive call
```

Function recursion required 3 statements

1. Initialization statement
2. Condition
3. Update statement

Initialization statement, which define initial value of condition
Condition defines how many time recursion has to repeated
Update statement, which updates condition

Example:

```
def fun1():
    print("inside fun1")
    fun1() # Recursion call


fun1()
```

Output

In the above program inside fun1 is repeated until reach to maximum recursion depth. The maximum recursion depth is 1000

How to find recursion depth?

“sys” module provides a function called getrecursionlimit()

```
>>> import sys
>>> sys.getrecursionlimit()
1000
```

How to modify recursion limit?
The default recursion limit can be changed using a function provided by “sys” module.

```
>>> sys.setrecursionlimit(5)
>>> sys.getrecursionlimit()
5
```

Example:

```
import sys
sys.setrecursionlimit(2)
def fun1():
    print("inside fun1")
    fun1() # Recursion call


fun1()
```

Example:	**Output**

def print_num(num):
  if num<=10: # condition
  print(num)
  print_num(num+1) # Recursion call
 	
print_num(1)
	1
	2
	3
	4
	5
	6
	7
	8
	9
	10

Example
# Finding factorial of a number 	
 	
def factorial(num):
  if num==0:
	Output
	Enter any number 4
	factorial of number is 24

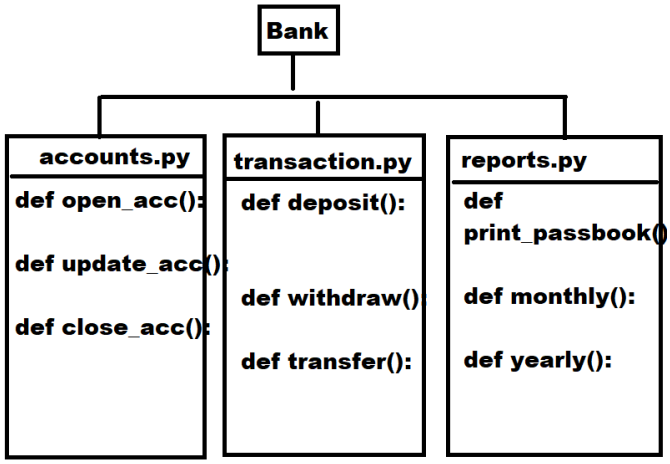
return 1 else: return num*factorial(num-1) number=int(input("Enter any number ")) fact=factorial(number) print(f'factorial of number is {fact}')	

Example
# Write a program to find sum of digits of input number 	
 	
s=0
def sum_of_digits(num):
global s
if num!=0:
r=num%10
s=s+r
sum_of_digits(num//10)
 	
number=int(input("Enter any number "))
sum_of_digits(number)
print(f'Sum of digits {s}')
	Output
	Enter any number 123
	Sum of digits 6
	
	Enter any number 456
	Sum of digits 15

Modules and Packages

What is modular programming?
Modular programming allows dividing application functionality into number of programs (modules).

Advantage:
1. Easy to understand and maintain code
2. Reusability between programs
3. Efficient way of developing projects



What is module?
Python program is called module (OR)
Module is nothing but a python program (OR) .py file

Python modules are 2 types
1. Predefined modules
2. User defined modules

Predefined modules
Existing modules/programs are called predefined modules.
These are libraries.
Example: sys, datetime, calendar, os,..

User defined modules
Programmer developed modules/programs are called user defined modules. These are application specific modules.

Creating module is nothing but writing python program.

PyCharm

PyCharm is python editor or IDE

How to download pycharm

https://www.jetbrains.com/pycharm/download/?section=windows