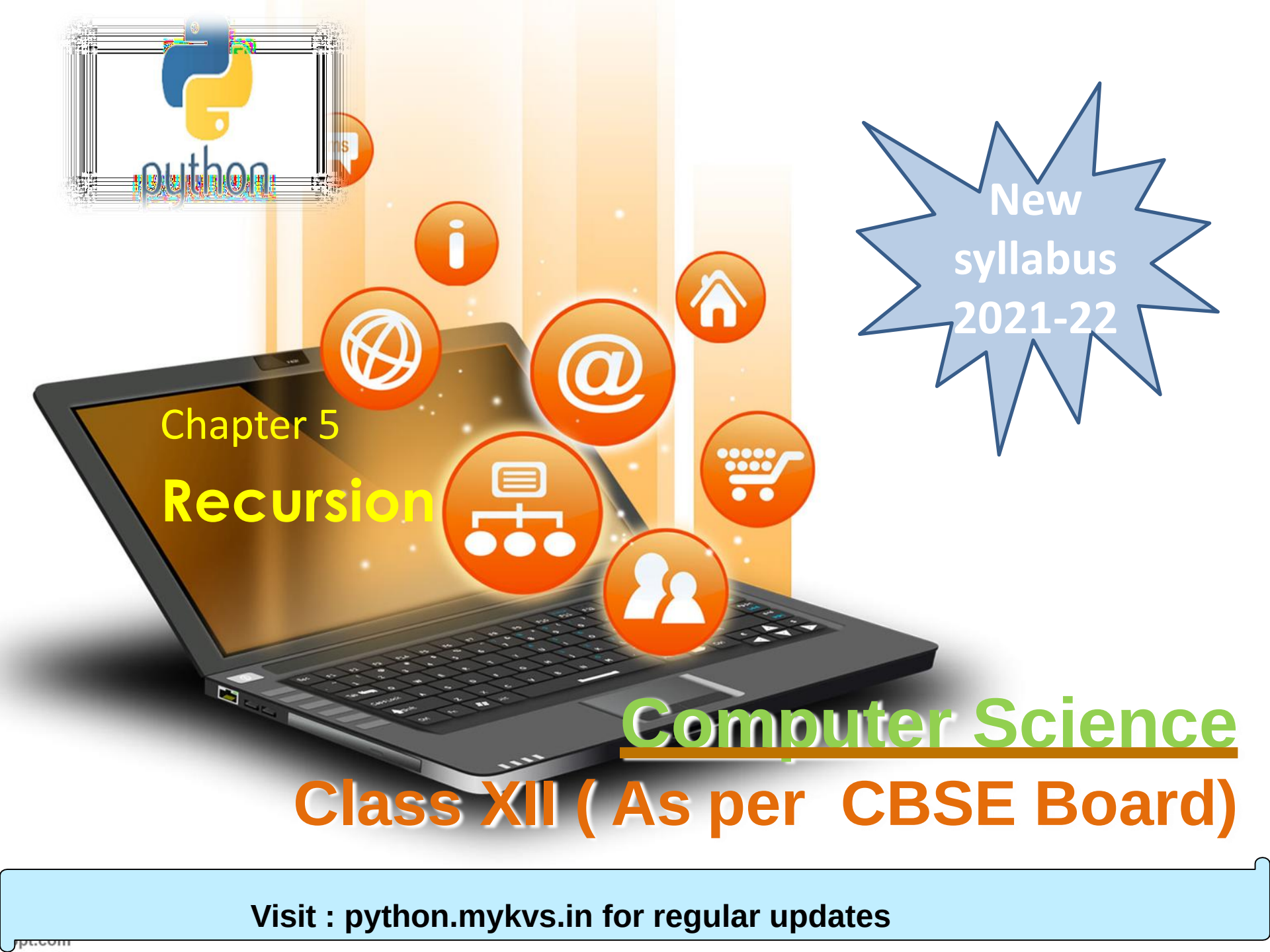


New syllabus 2021-22

Chapter 5

Recursion

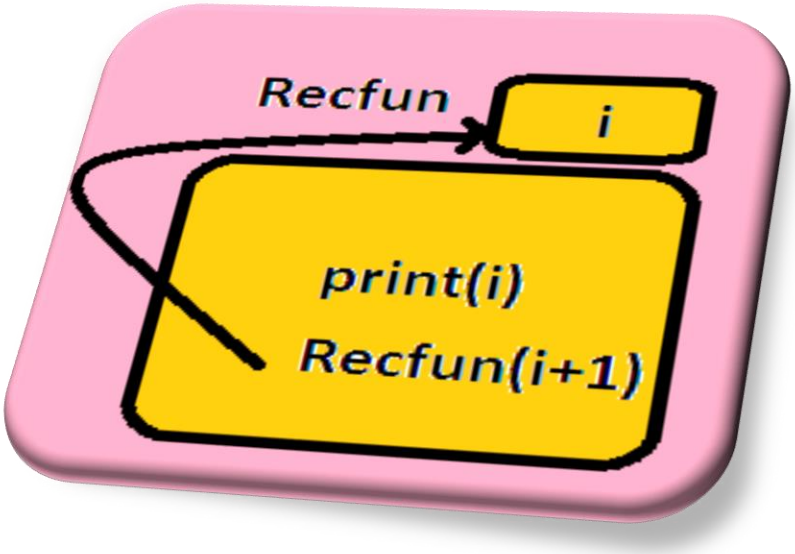
Computer Science

Class XII (As per CBSE Board)

Visit : python.mykvs.in for regular updates

Recursion

It is a way of programming or coding technique, in which a function calls itself for one or more times in its body. Usually, it is returning the return value of this function call procedure. If a function definition fulfils such conditions, we can call this function a recursive function.



Recursion

The Two Laws of Recursion

- Must have a base case - There must be at least one base criteria/condition, when such condition is met the function stops calling itself.
- Must move toward the base case - The recursive calls should moves in such a way that each time it comes closer to the base criteria.

Recursion

Simple algorithms with recursion

ALGORITHM

1.Test if n equal to base case return 1.

2.If not, then call the algorithm with n – 1(so as to move towards base case)

Recursion

Print a message forever

```
def hellomessage():
    print("hello")
    hellomessage()
hellomessage()
```

Program explanation

Run this program.it will display ‘hello’ message forever ,because there is no base case to exit.

It call hellomessage() function & Display ‘hello’

And again call hellomessage() function.

This will call hellomessage() function forever and will display ‘hello’ continuously

To exit press ctrl+c

Recursion

Sum of Natural Numbers Using Recursive Function

```
def sum(n):
    if n <= 1:
        return n
    else:
        return n + sum(n-1)
num = int(input("Enter a number: "))
print("The sum is: ", sum(num))
```

Program explanation

The input() function takes input from the user and int() function converts its type to an integer as input() return string. Here we call sum() function and pass the entered number ,which is assigned to n. The base condition for recursion is defined and if the input number is less than or equals to 1, the number is returned, else we return the same function call with number decremented by 1. In this way, the recursive function works in Python that can calculate the sum of natural numbers.It works like(suppose we pass 5 in input

5+sum(4)+sum(3)+sum(3)+1

Recursion

Factorial of a Number Using Recursion

ALGORITHM

1.Test if n <= 0. If so, return 1.

2.If not, then call the factorial algorithm with n – 1 and multiply the result by n and return that value.

(factorial 5)
(* 5 (factorial 4))
(* 5 (* 4 (factorial 3)))
(* 5 (* 4 (* 3 (factorial 2))))
(* 5 (* 4 (* 3 (* 2 (factorial 1)))))
(* 5 (* 4 (* 3 (* 2 1))))
(* 5 (* 4 (* 3 2)))
(* 5 (* 4 6))
(* 5 24))
120

Recursion

Factorial of a Number Using Recursion

PYTHON PROGRAM

```
def factorial(x):
    if x==1:
        return 1
    else:
        return x*factorial(x-1)

f=factorial(5)
print ("factorial of 5 is ",f)
```

```
(factorial 5)
(* 5 (factorial 4))
(* 5 (* 4 (factorial 3)))
(* 5 (* 4 (* 3 (factorial 2))))
(* 5 (* 4 (* 3 (* 2 (factorial 1)))))
(* 5 (* 4 (* 3 (* 2 1))))
(* 5 (* 4 (* 3  2 )))
(* 5 (* 4 6 ))
(* 5  24 ))
120
```

Recursion

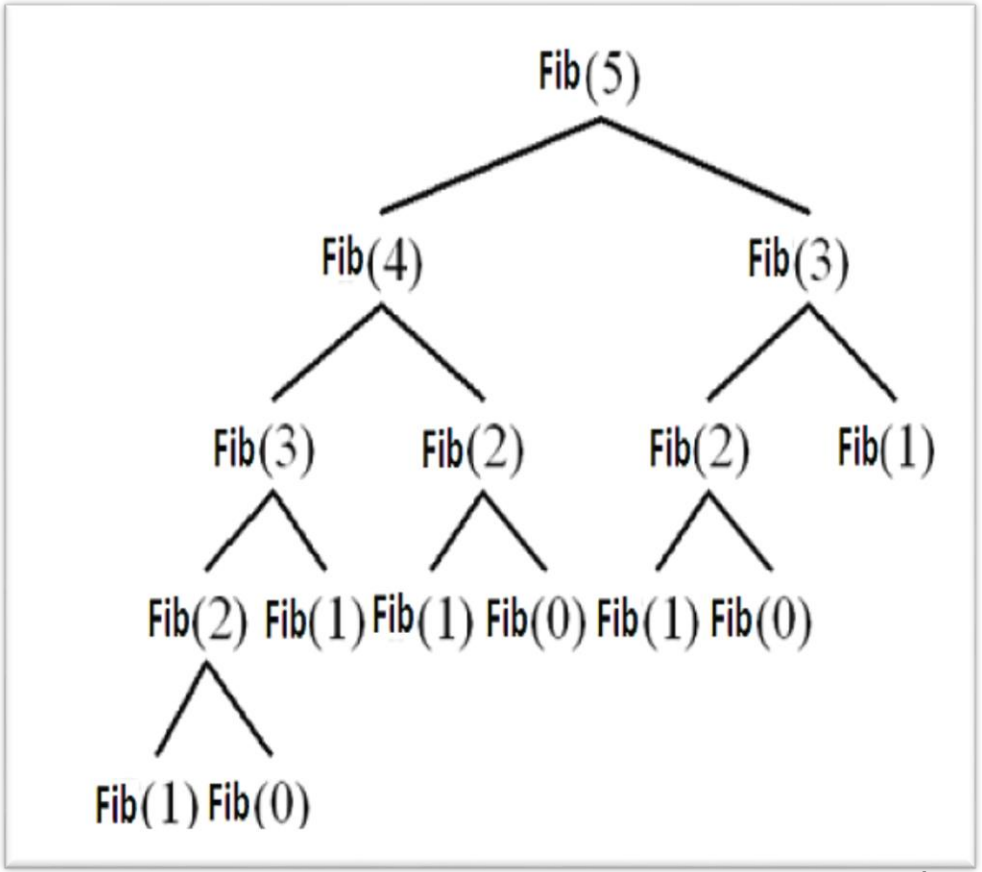
Fibonacci numbers Using Recursion

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, …

Algorithm

Fib(n)

1. If n =1 or n=2, then
2. return 1
3. Else
4. a = Fib(n-1)
5. b = Fib(n-2)
6. return a+b



Recursion

Fibonacci numbers Using Recursion

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, …

Program

```
def fib(n):
   if n <= 1:
      return n
   else:
      return(fib(n-1) + fib(n-2))

nterms = int(input("enter a number"))

if nterms <= 0:
   print("Plese enter a positive integer")
else:
   print("Fibonacci sequence:")
   for i in range(nterms):
      print(fib(i))
```