SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 919

Accredited by NAAC-UGC with 'A' Grade

Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

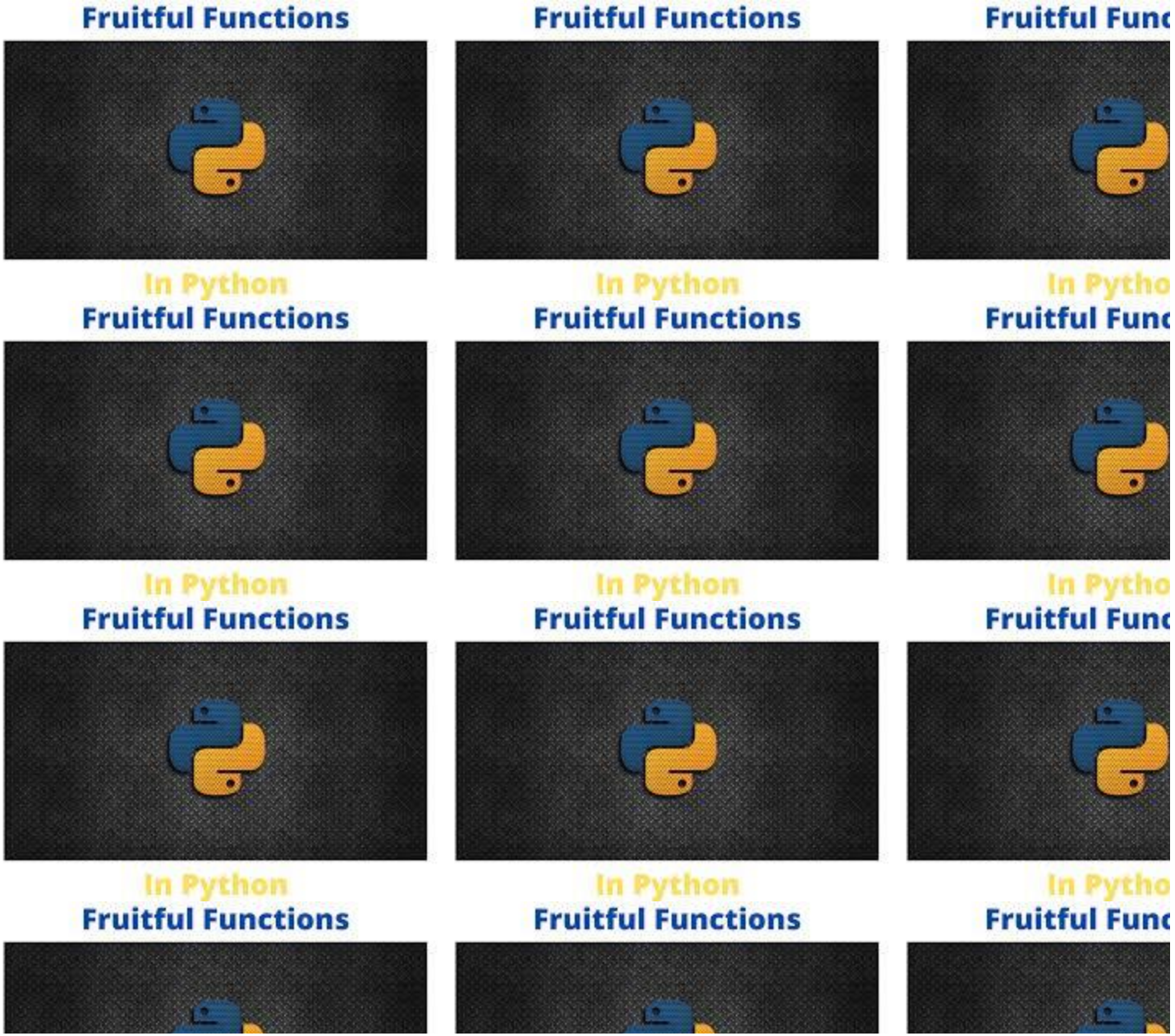
Department of Information Technology

COURSE NAME: 23ITB202-PYTHON PROGRAMMING

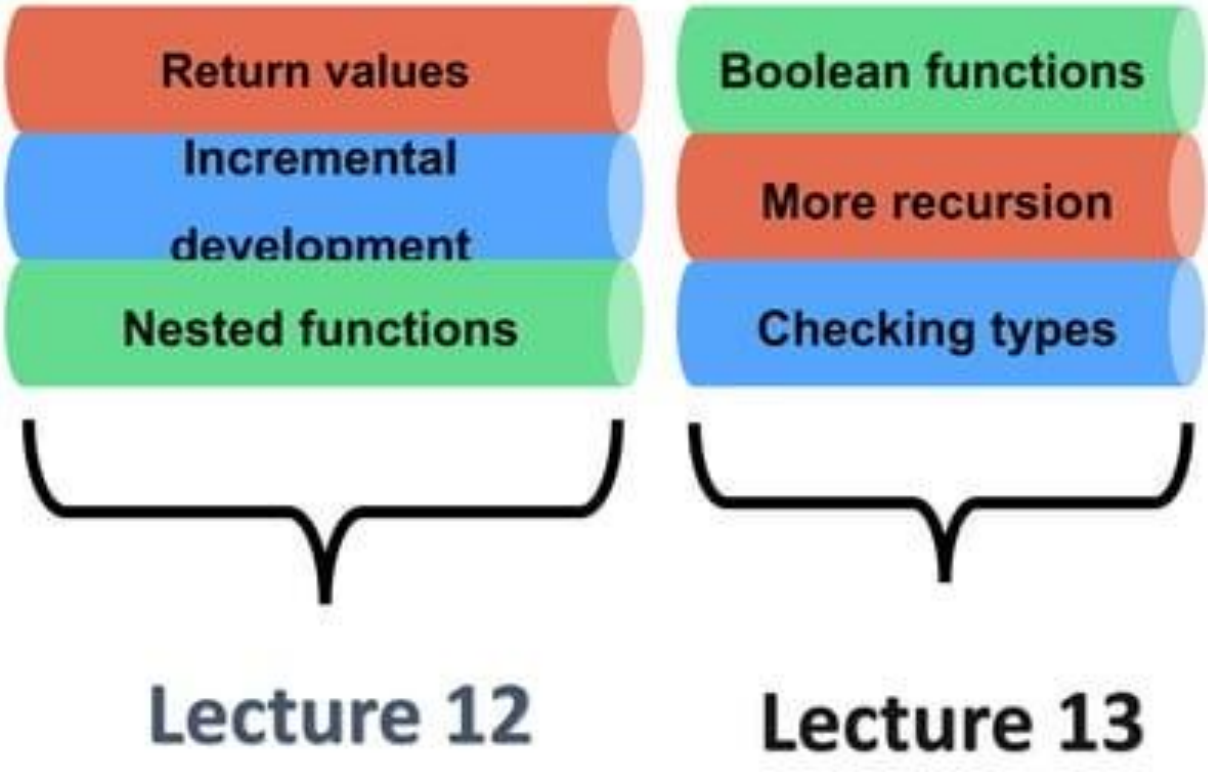
II YEAR/ III SEM

Unit : FUNCTION AND STRING

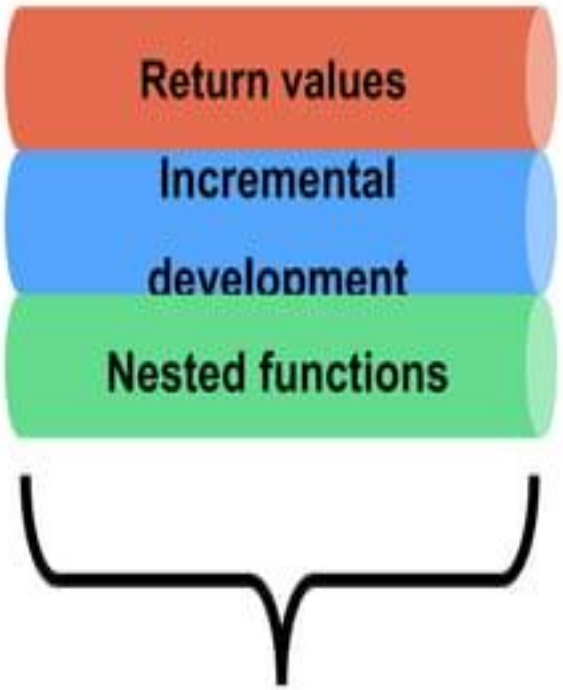
Topic : FRUITFUL FUNCTIONS


Fruitful Functions
In Python

Fruitful functions



Fruitful functions

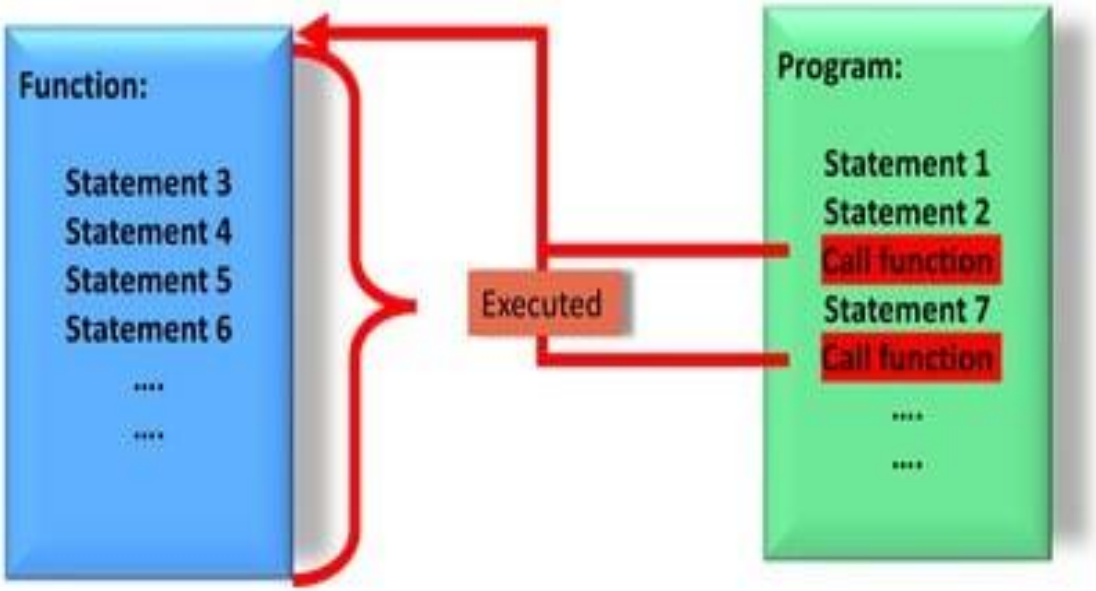


Lecture 12

Return values

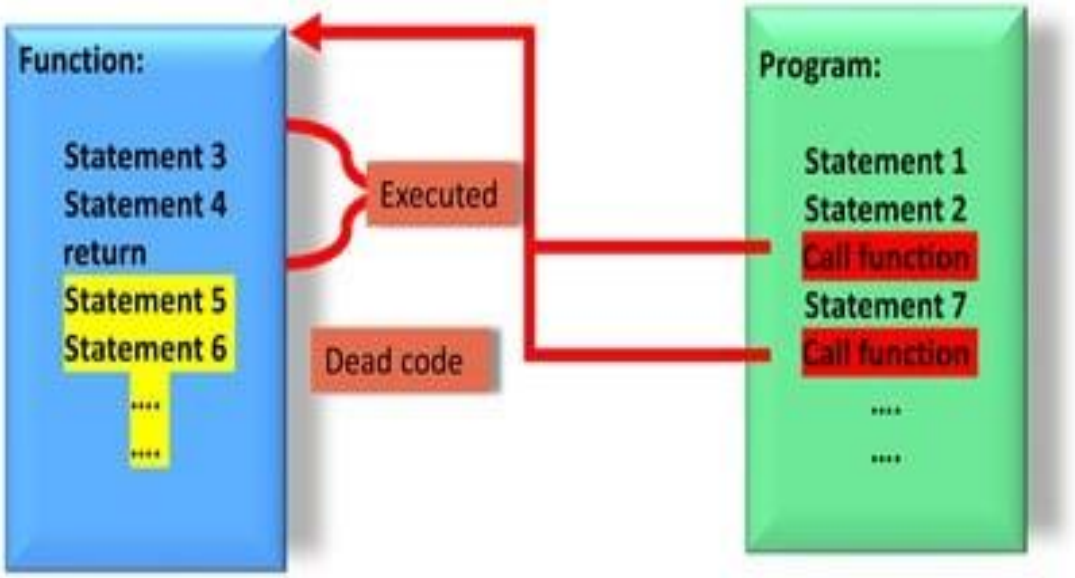
- As soon as a return statement executes, the function terminates without executing any subsequent statements.
- Code that appears after a return statement is called dead code.

Return values



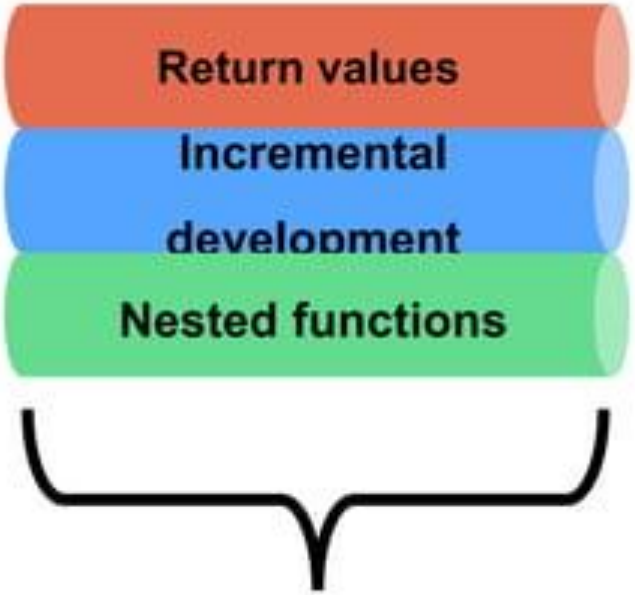
Function without return

Return values



Function with return

Fruitful functions



Lecture 12

Incremental development

- Larger functions are difficult to debug.
- To deal with complex programs, use incremental development.
- The goal of incremental development is to avoid long debugging sessions by adding and testing only a small amount of code at a time.

Incremental development

- Suppose you want to find the distance between two points, given by the coordinates (x1, y1) and (x2, y2).
- By the Pythagorean theorem, the distance is:

$$\text{distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Incremental development

- Suppose you want to find the distance between two points, given by the coordinates (x1, y1) and (x2, y2).
- By the Pythagorean theorem, the distance is:

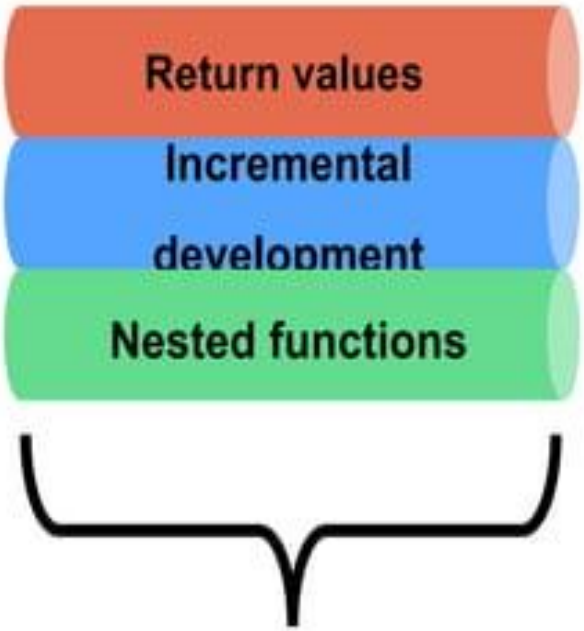
$$\text{distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Incremental development

- Suppose you want to find the distance between two points, given by the coordinates (x1, y1) and (x2, y2).
- By the Pythagorean theorem, the distance is:

$$\text{distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Fruitful functions



Lecture 12

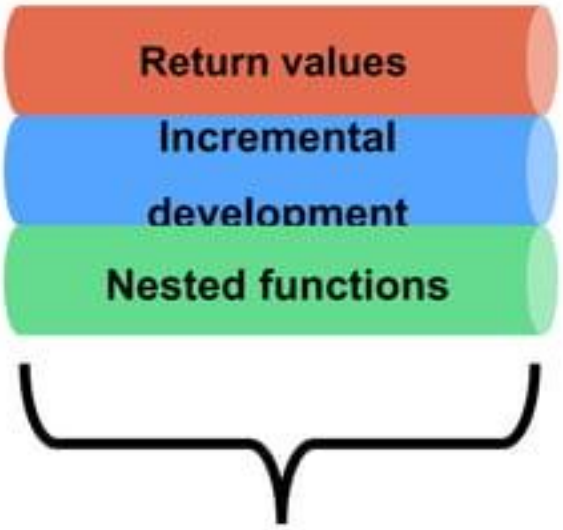
Nested functions

- We can call one function from within another. This ability is called composition.

```
def circle_area(xc, yc, xp, yp):
    radius = distance(xc, yc, xp, yp)
    result = area(radius)
    return result
```

```
def circle_area(xc, yc, xp, yp):
    return area(distance(xc, yc, xp, yp))
```

Fruitful functions



Lecture 12

Python Programming

(Lecture 13)

Unit – II (Part III)

Fruitful functions

Boolean functions

More recursion

Checking types

Boolean functions

- Functions can return Booleans.
- For hiding complicated tests inside functions.

```
def is_divisible(x, y):
    if x % y == 0:
        return True
    else:
        return False
```

```
def is_divisible(x, y):
    return x % y == 0
```

```
if is_divisible(x, y) == True:
    print 'x is divisible by y'
```

More recursion

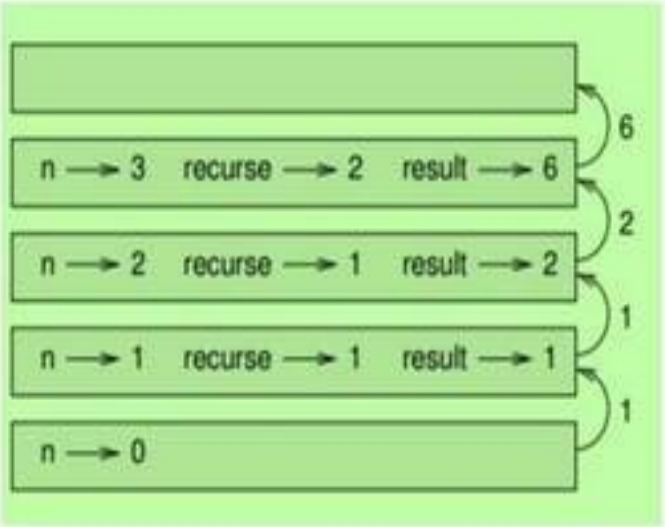
```
def factorial(n):
```

If the argument happens to be 0, all we have to do is return 1:

```
def factorial(n):
    if n == 0:
        return 1
```

More recursion

```
def factorial(n):
    if n == 0:
        return 1
    else:
        recurse = factorial(n-1)
        result = n * recurse
        return result
```



Checking types

```
def fibonacci (n):
    if n == 0:
        return 0
    elif  n == 1:
        return 1
    else:
        return fibonacci(n-1) + fibonacci(n-2)
```

What happens if we call `factorial` and give it 1.5 as an argument?

```
>>> factorial(1.5)
RuntimeError: Maximum recursion depth exceeded
```

Checking types

- We can use the built-in function isinstance to verify the type of the argument.

```
def factorial (n):
    if not isinstance(n, int):
        print 'Factorial is only defined for integers.'
        return None
    elif n < 0:
        print 'Factorial is not defined for negative integers.'
        return None
    elif n == 0:
        return 1
    else:
        return n * factorial(n-1)
```