1) A function is a useful method by which we can define a function and execute it whenever we want by calling it. The function consists of a group of statements and output of the function will be returned after executing these statements when we call the function.

Syntax: def function_name(parameters):
‘ ‘ ‘ Docstring ‘ ‘ ‘
Statement 1
Statement w
function_name(arguments)

Example: function to add two numbers

```
def addition(a,b):
    return a+b

print(addition(5,7)
```

2) There are three types of function.
- Normal function
- Lambda function (inline/anonymous function)
- Recursive function

- Normal function: In this function, we can define a function and execute it by calling it

- Lambda function: Its an anonymous function. This function can have any number of arguments but only one expression.

  Syntax: lambda arguments: expression

  Example: x=lambda a,b: a+b
  x(2,3)

- Recursive function: Recursive function means the function can call itself.

  Syntax: def function_name()
  Return function_name()

3) <u>Local variable</u>

If a variable in a function with a name which is exactly same as the name in the main block, the variable inside the function block is local to the function only. That is any change in the value inside the function will not be reflected in the variable in the main block.

Eg:

```
n=2
def change():
    n=5
    return n
print(change())
print(n)
```

output:

```
5
2
```

Global variable

If the change in variable inside the function has to be affected to the main block, the declare the variable as global inside the function

Eg:

```
x=50

def alter():
  global x
  print("Value of x=",x)
  x=20
  print("After executing the function value of x changed to ",x)

print("value of x before running the function",x)
alter()
print("The value of x= ",x)
```

output:

value of x before running the function 50
Value of x= 50
After executing the function value of x changed to 20
The value of x= 20

4) Parameters are the variable that are declared inside the parantheses in the function definition. Arguments are the values that are given during the function call.

5) Examples

- Default arguments

```
def new(a,b=5):
        return a+b
```

- Arbitrary arguments

```
def new(*args):
    return args
print(new(10,20,30))

(10, 20, 30)

#Returns a tuple of elements as output
```

- Arbitrary keyword arguments

```
def details(**a):

  return a

details(name='aswin',age=26)

#returns the values as a dictionary
```

6) Lambda function is also called as anonymous function. It consists of a number of variables and only one expression.

Syntax: lambda arguments: expression

7) None

8)
```
def rect(a,b):
 print("Area  of the rectangle= ",a*b)
 print("Perimeter  of the rectangle= ",2*(a+b))
rect(3,5)
```

9)
```
def circle(r):
    print("Area  of the circle= ",3.14*r*r)
    print("circumference  of the circle= ",2*3.14*r)
circle(10)
```

10)
```
x=lambda a:a+15
print(x(5))
```

11) No

12)
```
def listmanipulation(lst,c,l,v):
    if c=='remove' and l=='end':
        print(lst.pop())
    if c=='remove' and l=='beginning':
        print(lst[0])
    if c=='add' and l=='beginning':
        lst.insert(0,v)
        print(lst)
    if c=='add' and l=='end':
        lst.append(v)
        print(lst)
listmanipulation([1,2,3],'remove','end',0)
listmanipulation([1,2,3],'remove','beginning',0)
listmanipulation([1,2,3],'add','beginning',20)
listmanipulation([1,2,3],'add','end',30)
```

13) 1) 5
10

2) x is 50

Changed global x to 2

Value of x is 2

3) [1,1,1]

[1,1,1]

[4,8,16]

[9,27,81]

4) No Docstring

3

10

5) 0112

6) [1,2,3,4,[10,20,30]]

[1,2,3,4,[10,20,30]]

14)
```
def speed(s):
    if s<=70:
        print("Ok")
    if s>70:
        d=s-70
        p=d//5
        print(p,"points")
        if p>12:
            print("License suspended")

speed(80)
```

15)
```
def palindrome(st):
    temp=st
    l=len(st)
    s=""
    for i in range(l-1,-1,-1):
        s+=st[i]
    if s==temp:
        print("Palindrome")
    else:
        print("Not palindrome")
palindrome('level')
```