1. In Python, what is the difference between a built-in function and a user-defined function? Provide an

example of each.

**Built in function** - These functions are already available in python.it can be used without any additional imports EX - len(),print()

**User Defined function** - This function is created by the user. User defined function should be called before using it.The ‘Def’ keyword is used to create a function.

Ex

Def testfunction(a,b)

print(a+b)

testfunction(5,10)

**Output : 15**

2. How can you pass arguments to a function in Python? Explain the difference between positional

arguments and keyword arguments.

These are 2 ways to pass arguments to a function

**Positional Arguments**

Positional arguments are passed to the function based on the positions or order. The order of the arguments in a function call must match with the order of the parameters in function definition.

Def add(a,b)

print(a+b)

add(5,10)

Output -15

**Keyword arguments:**

Keyword arguments are passed to a function using the name of the parameter as a keyword. The order of the arguments does not matter in this case since each argument is explicitly associated with the parameter it corresponds to.

Def add(a,b)

print(a+b)

add(b=10,a=5)

Output - 15

3. What is the purpose of the return statement in a function? Can a function have multiple return

statements? Explain with an example.

The return statement in a function is used to specify the values that the function will return when it’s called.

Yes function can have multiple return statements. First return statement will return the value specified and exit the function immediately. Next return statement will not execute.

Ex

Def testreturn(num):

If num==0:

return 0

Elif num >1

return 1

Else:

return -1

testreturn(0)

4. What are lambda functions in Python? How are they different from regular functions? Provide an

example where a lambda function can be useful.

Lambda functions are anonymous functions which are created in single line without any name using the keyword - lambda

**How are they different from regular functions ?**

Lambda function have no name regular function have name,

Lambda function defined using the keyword - lambda, regular function defined using keyword - def

No need to call the lambda function.regular function should be called before using it.

**Uses of lambda function**

1.sorting a list of tuples

2.Filtering elements from list

3.mapping elements to new list

5. How does the concept of "scope" apply to functions in Python? Explain the difference between local

scope and global scope.

Scope refers to the accessibility and visibility of variables,objects and functions within the program.

Local scope - Variables which are defined within the function called local scope and they can be accessed within the function only. They are not access outside functions or any other function.

Global Scope -Global scope refers to the visibility and accessibility of variables throughout the entire program. Variables defined outside any function or block, at the top-level of the script, have global scope. They can be accessed from any part of the program, including inside functions. you need to use the **global** keyword to explicitly indicate that you are referring to the global variable and not creating a new local variable.

6. How can you use the "return" statement in a Python function to return multiple values?

Def multireturn(a,b,c):

return a,b,c

a,b,c=multireturn(1,2,3)

print(a,b,c)

**Output -**  1,2,3

7. What is the difference between the "pass by value" and "pass by reference" concepts when it comes to function arguments in Python?

There is no pass by value and pass by reference in python. **Python uses pass by object reference**

If the object is mutable ex-list,set,dict when you pass this intoa function. The function will modify the original object. Because its uses the shared reference

Example:

def functiontest(li):

li.append(4)

return li

li=[1,2,3]

print(functiontest(li)) #[1,2,3,4]

print(li) #[1,2,3,4]

For immutable objects (e.g., strings, numbers) the function cannot modify the original object. Instead, it can only create a new object and reassign it to the local variable.

Example

def modify\_string(s):

s += " World"

print( s) # Hello world

my\_string = "Hello"

modify\_string(my\_string)

print(my\_string) #Hello

8. Create a function that can intake integer or decimal value and do following operations:

a. Logarithmic function (log x)

import math

def Logarithmic(x):

return math.log10(x)

print(Logarithmic(100))

Output =2.0

b. Exponential function (exp(x))

import math

def Logarithmic(x):

return math.exp(x)

print(Logarithmic(2))

Output = 7.38905609893065

c. Power function with base 2 (2)

def Logarithmic(x):

return pow(2,x)

print(Logarithmic(4))

Output = 8

d. Square root

import math

def Logarithmic(x):

return math.sqrt(x)

print(Logarithmic(25))

Output =5.0

9. Create a function that takes a full name as an argument and returns first name and last name.

def fulname(name):

li=name.split()

return li[0],li[1]

a,b=fulname("manoj prabakaran")

print(a)

print(b)

Output

manoj

prabakaran