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

* Built-in functions are functions that are already defined in the Python interpreter.

Ex: Some examples of built-in functions include print(), len(), min(), and max().

* User-defined functions are functions that you define yourself.To define a user-defined function,

you use the def keyword.

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

* Positional arguments

Must be passed in the order they are defined, Less flexible

Example : add\_numbers(1, 2)

* Keyword arguments

Can be passed in any order,More flexible

Example : add\_numbers(x=1, y=2)

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 value that the function should evaluate to and send back to the caller.
* It allows a function to provide a result or output that can be stored in a variable, used in further computations, or displayed to the user.

The return statement serves multiple purposes:

* Returning a value
* Terminating the function

Ex: ef add\_numbers(a, b):

return a + b

result = add\_numbers(5, 7)

print(result) # Output: 12

4. What are lambda functions in Python? How are they different from regular functions? Provide an example where a lambda function can be useful.

* In Python, lambda functions, also known as anonymous functions, are small, one-line functions that don't have a name.
* They are defined using the lambda keyword and can take any number of arguments but can only have one expression.

The syntax for a lambda function is:

lambda arguments: expression

Ex:

# Lambda function

multiply = lambda a, b: a \* b

result = multiply(5, 7)

print(result) # Output: 35

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

In python, Scope refers to the visibility of variables.

A variable's scope determines where in the program it can be accessed.

In Python, there are two types of scope: local scope and global scope.

Local scope is the scope of a variable that is defined inside a function.

Variables in local scope can only be accessed from within the function that defines them.

Global scope is the scope of a variable that is defined outside of any function.

Variables in global scope can be accessed from within any function in the program.

Ex:

def my\_function():

"""This function defines a local variable."""

local\_variable = "This is a local variable."

print(local\_variable)

global\_variable = "This is a global variable."

my\_function()

print(global\_variable)

6. How can you use the "return" statement in a Python function to return multiple values? 7. What is the difference between the "pass by value" and "pass by reference" concepts when it comes to function arguments in Python?

Ans:

In Python, you can use the return statement to return multiple values from a function.

To do this, we simply need to supply several return values separated by commas.

For example:

def my\_function():

"""This function returns multiple values."""

first\_value = 10

second\_value = 20

return first\_value, second\_value

first\_value, second\_value = my\_function()

print(first\_value)

print(second\_value)

In Python, there are two ways to pass arguments to functions: pass by value and pass by reference.

Pass by value means that the value of the argument is copied to the function.

Any changes made to the value in the function will not affect the original value.

Pass by reference means that the reference to the argument is passed to the function.

Any changes made to the value in the function will affect the original value.

8. Create a function that can intake integer or decimal value and do following operations: a. Logarithmic function (log x)

b. Exponential function (exp(x))

c. Power function with base 2 (2x)

d. Square root

Ans:

import math

def math\_operations(value):

log\_value = math.log(value)

exp\_value = math.exp(value)

power\_value = math.pow(2, value)

sqrt\_value = math.sqrt(value)

return log\_value, exp\_value, power\_value, sqrt\_value

result = math\_operations(5.7)

print("Logarithmic value:", result[0])

print("Exponential value:", result[1])

print("Power value (base 2):", result[2])

print("Square root value:", result[3])

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

def extract\_name(full\_name):

names = full\_name.split()

first\_name = names[0]

last\_name = names[-1]

return first\_name, last\_name

name = "John Doe"

first\_name, last\_name = extract\_name(name)

print("First Name:", first\_name)

print("Last Name:", last\_name)