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

Ans - In Python, a built-in function is a function that is provided by the Python language itself. These functions are pre-defined and readily available for use without requiring any additional code. Examples of built-in functions include `len()`, `print()`, and `type()`.

On the other hand, a user-defined function is created by the user to perform a specific task. These functions are defined by the user using the `def` keyword and can be customized as per the user's requirements. Here's an example of a user-defined function that calculates the square of a number

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

Ans - In Python, you can pass arguments to a function by placing them inside the parentheses when calling the function. Positional arguments are passed based on their position, while keyword arguments are passed with a specific parameter name, allowing for more flexibility and clarity.

1. What is the purpose of the return statement in a function? Can a function have multiple return statements? Explain with an example.

Ans - The purpose of the return statement in a function is to specify the value that the function should return when it is called. Yes, a function can have multiple return statements, and when executed, the function will return the value specified in the first executed return statement. Here's an example:

def get\_grade(score):

if score >= 90:

return "A"

elif score >= 80:

return "B"

else:

return "C"

In this example, depending on the score provided, the function will return the corresponding grade: "A", "B", or "C".

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

Ans - Lambda functions in Python are anonymous functions that are defined using the `lambda` keyword. They are different from regular functions because they can be defined in a single line and do not require a function name. Lambda functions are often used for small, one-time operations.

Here's an example where a lambda function can be useful:

# Sorting a list of tuples based on the second element

my\_list = [(2, 'apple'), (1, 'banana'), (3, 'orange')]

sorted\_list = sorted(my\_list, key=lambda x: x[1])

print(sorted\_list)

In this example, the lambda function `lambda x: x[1]` is used as the key parameter in the `sorted()` function to sort the list of tuples based on the second element.

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

Ans - In Python, "scope" refers to the visibility and accessibility of variables within a program. Functions in Python have their own scope. Local scope refers to variables defined within a function and is only accessible within that function. Global scope refers to variables defined outside of any function and can be accessed from anywhere in the program.

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

Ans - To return multiple values from a Python function, you can use the return statement followed by a comma-separated list of values. These values can be unpacked into separate variables when calling the function.

1. 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, function arguments are passed by reference. This means that the memory address of the object is passed to the function, allowing modifications to be made to the original object. However, immutable objects like strings and numbers are effectively passed by value since they cannot be modified in place.

1. Create a function that can intake integer or decimal value and do following operations:
   1. Logarithmic function (log x)
   2. Exponential function (exp(x))
   3. Power function with base 2 (2x)
   4. Square root

Ans -

def math\_operations(x):

log\_value = math.log(x)

exp\_value = math.exp(x)

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

sqrt\_value = math.sqrt(x)

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

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

Ans - For example,

if first name is divya and lastname is pratap, the output should be:

Hello divya Pratap

#Function to display full name

#The requirements are listed below:

#1. The function should have 2 parameters to accept first name and #last name.

#2. Concatenate names using + operator with a space between first #name and last name.

#3. Display full name.

def fullname(first,last):

#+ operator is used to concatenate strings

fullname = first + " " + lastRead

print("Hello",fullname)

#function ends here

first = input("Enter first name: ")

last = input("Enter last name: ")

#function call

fullname(first,last)

Output:

Enter first name: divya

Enter last name: pratap

Hello divya pratap