**ASSIGNMENT 03(Amrutesh)**

1. **Why are functions advantageous to have in your programs?**

Ans: Reusability: Functions can reduce the repeatability of writing code also allow to reuse code multiple times over and over again. This can save time and effort

Modularity: Functions helps us to break down code into smaller, this can make code easy to understand and to have a quick update or change in code if needed

Testing : Functions can be tested independently which can easily to identify errors or bugs in code

Organization: Functions can help to organise code into logical groupings,which can make easier to navigate and understand

1. **When does the code in a function run: when it’s specified or when it’s called?**

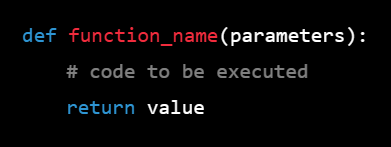
Ans : Code in function runs when it is called, not when it is specified.

When we define a function , we are actually creating a block of code that can be called later in the program, the function is not executed when defined or specified , but waits for call from another part of the program

When the function is called, the program jumps into the function code and execute it, then the func may return a value or perform some action

1. **What statement create the function**

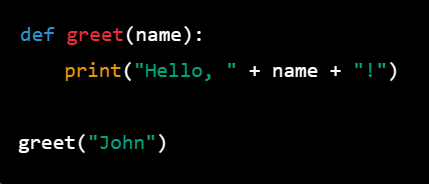
**Ans : def** statement is used



1. **What is the difference between a function and a function call?**

**Ans:** In Python, a function call is the act of executing a defined function with a given set of arguments.

A function is a block of reusable code that performs a specific task. To use a function, you need to call it with the required parameters. When a function is called, the program control jumps to the function definition, executes the statements inside the function body, and returns back to the point where the function was called.

****

In the above example, we define a function **greet()** that takes a parameter **name**. When we call the function **greet("John")**, the value **"John"** is passed as an argument to the function. The function then prints the string **"Hello, John!"** to the console

1. **How many global scopes are there in a Python program? How many local scopes?**

In a Python program, there is one global scope and potentially many local scopes.

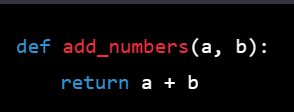
The global scope is the outermost scope, which is created when a Python program starts running. Any names defined at this level are considered to be part of the global scope and can be accessed from anywhere in the program.

Local scopes, on the other hand, are created whenever a function is called. Each function call creates a new local scope, which is used to store the function's local variables and any arguments passed to it. Local variables are not visible outside of the function's scope, and they are destroyed when the function returns.

So the number of local scopes depends on how many function calls are made during the execution of the program, while there is always only one global scope.

**6.** **What is the concept of a return value? Is it possible to have a return value in an expression?**

Ans :Return value is the value that a function or method returns when it completes its execution. When a function or method is called, it may perform some computation and return a value to the caller. The return value can be any valid Python object, including integers, floats, strings, lists, tuples, dictionaries, or even other functions.

To return a value from a function, you can use the **return** statement followed by the value you want to return. For example, the following function returns the sum of two 

**7.What happens to variables in a local scope when the function call returns?**

**Ans** : when the function call returns in python the variables in the local scope of the function are destroyed and their memory block is released

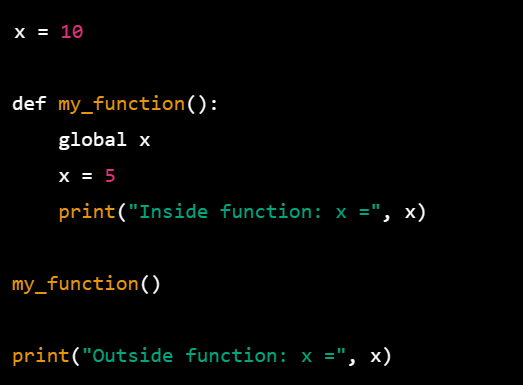
This means that any values assigned to local variable during the execution of the function are lost and cannot be accessed outside the function

**8.** **If a function does not have a return statement, what is the return value of a call to that function?**

Ans : If the function does not have a “**return**” statement, return value of call to that function is **“None”**

**9** . **How do you make a function variable refer to the global variable?**

**Ans:** we can use **‘global’** keyword to indicate that variable inside a function should refer to global variable with the same name

****

The x is initially set to 10 but in function we use the global keyword to make x variable global

Inside the function we assigned the x value is 5 and the output of that will be 5;

Again, if we print the value of x outside the value will be again 5

**10.what is the data type of None?**

**Ans:** Data type of None is **‘NoneType’**

**None**  is special constant the represents the absence of value or to indicate that a variable of function does not return a value or has not yet assigned a value

**11.** **What does the sentence import areallyourpetsnamederic do?**

**Ans:**  if we try to run in interpreter, we will get the error as **ModuleNotFound**

As it does not refer any existing module or package in python

**12.** **If you had a bacon() feature in a spam module, what would you call it after importing spam?**

**13.** **What can you do to save a programme from crashing if it encounters an error?**

**Ans:** We can use the **try** and **except** block to encounters error to save a programme from crashing and we can log those errors to debug

**14.** **What is the purpose of the try clause? What is the purpose of the except clause?**

**Ans: Try** and **Except** blocks are used to handle errors in Pyhton

Try block is enclosed with block of code that my potentially raise an exception,

when Try block is executed Python will attempt to execute the code normally,

if any exception occurs during execution the remaining block of code will enter to Except zone

when the exception occurs within the try block the python will start to find Except block to match the type of expection if matching except block is found , the code in the except block is executed to handle the exception