**ASSIGNEMENT-3**

1. Modularity, Code Organization, Reusability and Code Efficiency
2. For example

def greet ():

print ("Hello, World!")

print ("Before function call")

greet ()

print ("After function call")

1. Def function name(parameters):
2. The difference between a function and a function call can be summarized as follows:
3. **Function**: A function is a block of reusable code that performs a specific task or a set of operations. It is created using the **def** statement and is defined with a name, parameters (optional), and a code block. Functions allow you to encapsulate and organize code into modular units, making it easier to manage, understand, and reuse.
4. **Function Call**: A function call is the act of invoking or executing a function to perform its defined task. It is the statement that triggers the execution of the code within the function. When a function is called, the program jumps to the function's code block, executes the statements within it, and then returns to the point of the function call to continue the program's execution.
5. In a Python program, there can be one global scope and multiple local scopes.
6. **Global Scope**: The global scope refers to the outermost level of the program, outside of any functions or classes. Variables defined in the global scope are accessible throughout the entire program, including inside functions or classes. There is only one global scope in a Python program.
7. **Local Scopes**: Local scopes are created when you define a function or a class. Each function or class has its own local scope. Variables defined within a function or class are accessible only within that specific scope. Local scopes are created and destroyed as functions or classes are called and return. It's important to note that each function or class has its own separate local scope, meaning that variables with the same name can exist in different local scopes without interfering with each other.
8. 1. Function execution completes

2.Stack frame removal takes place

3.Local variables deallocation happens.

4.and the only it returns.

1. The concept of return value is similar to that of print where the final product in print is shown but the final product in return only takes the value back and stores.
2. If a function does not have a return statement, the return value of a call to that function is a special value called "None" in Python. "None" is a built-in constant that represents the absence of a value. It is similar to "null" in other programming languages.
3. We use the keyword called ‘global.’
4. Its datatype is: None type.
5. It can be imported as a module and its functions can be used
6. In Python, when you import a module, you can access its functions, classes, and variables using the dot notation. The general syntax to access an item within a module is module\_name.item\_name. In this case, bacon() is a function within the spam module, so we use spam.bacon() to call it.
7. We can use both ‘try’ and ‘except’ blocks
8. The try block lets you test a block of code for errors. The except block lets you handle the error. The else block lets you execute code when there is no error. The finally block lets you execute code, regardless of the result of the try- and except blocks.