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

**Answer –**

Functions enhance code modularity, readability, maintainability, and reusability, leading to more efficient and manageable programs.

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

**Answer –**

The code in a function runs when it is called, not when it is specified.

**3.What statement creates a function?**

**Answer –**

The ‘**def’** statement creates a function.

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

**Answer –**

A function is the definition or blueprint of a specific functionality, while a function call is the actual execution of that function, invoking its code and performing the desired actions.

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

**Answer –**

Python programs have one global scope.

Each function call creates its own local scope. Therefore, the number of local scopes depends on the number of function calls made in the program.

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

**Answer –**

When a function call returns, the local scope, including the variables defined within it, is destroyed.

Any variables defined within the local scope of a function are no longer accessible or existent outside of that function.

The values of local variables are typically discarded, and they cannot be accessed or referenced once the function call returns.

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

**Answer –**

The concept of a return value refers to the value that a function can provide back to the caller after its execution.

Yes, it is possible to have a return value in an expression. A return value can be assigned to a variable or used directly in an expression for further computations or operations.

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

**Answer –**

If a function does not have a return statement, the return value of a call to that function is **None**.

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

**Answer –**

To make a function variable refer to the global variable, you can use the global keyword followed by the variable name inside the function.

This indicates that the variable being referenced is the global variable with the same name, rather than creating a new local variable within the function.

x = 10 # Global variable

def my\_function():

global x

x = 20 # Modifying the global variable

my\_function()

print(x) # Output: 20

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

**Answer –**

The data type of **None** is **NoneType.**

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

**Answer –**

**ModuleNotFoundError**: No module named 'areallyourpetsnamederic'

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

**Answer –**

**"spam.bacon()"**

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

**Answer –**

1. Identify the section of code where the error might occur.
2. Wrap that section of code within a try block.
3. Catch and handle specific exceptions that you expect might occur within an except block.
4. Optionally, you can include a finally block to specify code that should be executed regardless of whether an exception occurs or not.

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

**Answer –**

The purpose of the try clause in Python is to enclose code that might raise an exception and allows you to monitor it for errors.

The purpose of the except clause is to handle specific exceptions that may occur within the try block and define how the program should respond to those exceptions.