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

Ans:Functions offer several advantages in programming, making them an essential part of writing clean, efficient, and maintainable code. Here are some key reasons why functions are beneficial:

(a) Code Reusability (b)Modularity (c)Abstraction (d)Maintainability (e)Improved Readability (f) Debugging and Testing (g)Parameterization

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

When you define a function, you're essentially telling Python what the function should do. However, the code inside the function won't execute until you explicitly call the function in your program.

def greet(name):

print(f"Hello, {name}!")

greet("Alice")

greet("Bob")

3. What statement creates a function?

The statement that creates a function in Python is the def statement.

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

Function:A function is a block of reusable code designed to perform a specific task. It is defined once and can be called whenever needed in the program.

A function may accept input in the form of parameters and may return a result using the return statement.The function is created using the def keyword in Python.

Function Call:A function call is the execution of a function. It is an instruction that tells Python to run the code inside the function with the given arguments (if any).

When a function is called, the control of the program is transferred to the function, the code inside it is executed, and then control is returned to the place where the function was called.

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

Global scope: Only one global scope, accessible throughout the program.

Local scope: Multiple local scopes, each associated with functions or methods. Each function has its own local scope.

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

When the function my\_function() is called, the variables a and b are created in the local scope of the function.

After the function finishes executing and returns, the local scope is destroyed, and the variables a and b are no longer accessible outside the function.

The attempt to print a outside the function will result in a NameError because a is a local variable, and its scope was destroyed once the function returned.

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

In Python, a return value is the value that a function sends back to the caller when it finishes execution. The return statement is used to specify the value to be returned from a function. Once the return value is provided, the function terminates, and the execution control is passed back to where the function was called.

Yes, a return value can be part of an expression. When a function returns a value, you can use that value directly in an expression or computation.

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

In Python, if a function does not explicitly return a value using the return statement, it will implicitly return None by default.

This means that even if you don't specify a return statement, Python will still consider the function to have a return value, and that value will be

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

To make a function variable refer to a global variable, you can use the global keyword inside the function. This allows the function to modify the global variable instead of creating a new local variable with the same name.

The global keyword tells Python that the variable being referenced inside the function should refer to the global variable (defined outside the function) rather than creating a new local variable.Without the global keyword, any variable assigned a value within a function will be treated as a local variable, and changes made to it will not affect the global variable.

10. What is the data type of None?

The data type of None in Python is NoneType.None is a special constant in Python that represents the absence of a value or a null value.

It is commonly used as a default return value for functions that do not explicitly return anything.It is also used to indicate "no value" or "empty state" in variables.

11. What does the sentence import areallyourpetsnamederic do?

The sentence import areallyourpetsnamederic is an import statement in Python. However, unless a module named areallyourpetsnamederic actually exists, this statement will raise an ImportError because Python won't find a built-in or installed module with that name.

If a user creates a custom Python module named areallyourpetsnamederic.py, then the import statement will work.This is sometimes used as a joke or example to show how Python's import system works.

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

If you have a bacon() function inside a module named spam, you can call it after importing the spam module using dot notation:

Importing only the bacon function from spam.

Importing with an alias.

Importing everything from spam (not recommended due to namespace pollution).

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

To prevent a Python program from crashing when it encounters an error, you can use exception handling with the try-except block. This allows your program to catch and handle errors gracefully instead of terminating unexpectedly.

Use try-except blocks to catch errors

Handle specific exceptions instead of using a generic except:

Use finally to ensure cleanup tasks are completed

Use logging instead of just printing errors

Validate user input before processing

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

The try clause is used to wrap a block of code that might cause an exception. If an error occurs within this block, Python will stop executing it and look for an appropriate except block to handle the error.

The except clause catches and handles exceptions that occur inside the try block. This prevents the program from crashing and allows you to define how to handle the error gracefully.