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

Ans. Functions are very useful it because you ca use them at any time in any program just directly by calling no need to write full

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

Ans. When it is called

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

Ans def function\_name(parameter)

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

Ans. function is the block of code that can perform a particular task. Function call:- A function call is an instruction that tells the program to execute that particular task

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

Ans. It's important to note that local scopes are nested within the global scope. So, while there can be multiple local scopes within a program (depending on the number of function calls or code blocks executed), there is only one global **scope.**

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

Ans. When a function call returns, the local scope and the variables defined within it are destroyed. This process is known as "cleaning up the stack."

In most programming languages, when a function is called, a new stack frame is created to store the local variables, function arguments, and other related information. The local variables declared within the function are stored within this stack frame. The stack frame is a specific region of memory allocated for the function's execution.

Once the function call completes, either by reaching the end of the function or encountering a return statement, the stack frame associated with that function call is deallocated, and the memory used by the local variables is freed.

As a result, the local variables within the function's scope are no longer accessible or valid outside of the function. They are destroyed, and any values they held are lost. Attempting to access those variables after the function call has returned can result in undefined behavior or errors.

It's important to note that global variables or variables defined in an outer scope of the function will continue to exist and retain their values even after the function call returns.

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

Ans. A return value serves as the output or result of a function, providing information or data that can be used further in the program. It allows functions to communicate their results to the calling code, enabling the code to use or process that value as needed.

Yes, it is possible to have a return value in an expression. In many programming languages, including popular ones like Python, Java, and C++, you can use a function call as part of an expression and assign its return value to a variable or use it directly in calculations.

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

Ans. If a function does not have a return statement, the return value of a call to that function will be None in many programming languages, including Python. None is a special value that represents the absence of a value or a null value.

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

Ans. in most programming languages, including Python, you can make a function variable refer to a global variable by using the global keyword within the function. This tells the function that the variable being referenced is a global variable, rather than a local variable defined within the function's scope. Here's an example in Python:

**10. What is the data type of None?**  
Ans. In Python, the data type of None is called NoneType. NoneType represents the absence of a value or a null value. It is a built-in type in Python that has only one possible value, which is None.

**11. What does the sentence import a really our pets name deric do?**

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

Ans. After importing the "spam" module, you can access the "bacon()" feature using the module name as a prefix. By using the module name followed by a dot, you can access any function or attribute defined within the "spam" module. In this case, you would call the "bacon()" function using the "spam.bacon()" syntax.

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

Ans. To save a program from crashing when it encounters an error, you can implement error handling mechanisms using exception handling. Exception handling allows you to gracefully handle and recover from errors or exceptional situations that may occur during program execution.

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

Ans. The purpose of **the try clause** is to enclose a block of code that might raise an exception. By using a try block, you tell the program to execute the statements inside it, and if any exception occurs, the program will not immediately terminate but will instead jump to the corresponding except block.

When the code inside the try block encounters an exception of type Exception Type, the program flow transfers to the except block. The purpose of **the** **except clause** is to specify how to handle the exception. It allows you to define the actions or code that should be executed when a specific type of exception occurs.