1. Why are functions advantageous to have in your programs?
   * At different segments of the program, a specific set of statements may need to be executed multiple times with varying inputs. The set of sentences that must be repeated can be saved as a function and called whenever needed by executing just one line of code rather than retyping the entire set of statements.
   * Certain sections of the code may be quite complicated. These can be defined in isolated locations and only called when needed, keeping the basic structure of the code simple and easy to understand.
   * It reduces chances of errors
   * The function defined enables reusability in other programs as well

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

Only when the function is called, the code is executed.

3. What statement creates a function?

The function definition statement is used to define a function. It is illustrated by the following statement:

**def f(x):**

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

The process of defining the purpose of a specific function is known as function definition. A function is a collection of statements that performs the desired task. However, the function is not executed during the definition phase. Only when the function is called is it executed. The statement ‘Function call' instructs the program to invoke the function. If the function accepts inputs, known as arguments, they can be supplied at the time of the function call.

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

In Python, there is only **ONE** global scope.

The number of local scopes is equal to the number of times a specific function is called (or executed) within which the variable is locally specified.

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

The expression in the function call's return statement is evaluated based on the values stored in the variables' local scope (unless global scope is explicitly mentioned).

Python returns to global scope after calling the function and executing the return statement, and the local scope variable and the data contained in it are removed from memory.

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

The expression ‘return' at the end of a function description governs whether the function returns any value when called. A return statement is not always necessary. ‘return’, on the other hand, may have something specified in it, such as a returned variable, an expression, or even a print statement.

‘return’ value could be an expression inside the function but we can’t use ‘return’ statement in an expression outside the function definition.

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

When there is no return statement in the function specification, it **returns nothing**. This is expressed as the function returning **‘None'**.

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

When the keyword **‘global'** is used before the variable name, the function variable refers to the global variable .

10. What is the data type of None?

It is **‘NoneType’**

11. What does the sentence import areallyourpetsnamederic do?

It imports the module ‘areallyourpetsnamederic’

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

**spam.bacon()**

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

The use of **"Exception Handling"** prevents the program from crashing when an error occurs. It is made up of try and except blocks.

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

* + The actual code block to be executed is included within the **‘try'** block.
  + The **‘except'** block includes a series of commands that will be executed if a specific form of error occurs within the ‘try' block.