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

→ **Functions reduce the need for duplicate code. This makes programs shorter, easier to read, and easier to update.**

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

→ **The code in a function executes when the function is called, not when the function is defined.**

3. What statement creates a function?

→ **A function can be defined with “def” keyword followed by name of the function**

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

→ **A function is a piece of code which enhanced the reusability and modularity of your program. It means that piece of code need not be written again. A function call means invoking or calling that function. Unless a function is called there is no use of that function.**

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

→**There is one global scope in a Python program.**

→**Local scopes are created whenever a function is called or a block of code is executed, and multiple local scopes can exist simultaneously depending on the number of function calls and code blocks.**

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

→ **Variables are no longer accessible: Once the function call returns, any variables defined within the local scope are no longer accessible from outside the function. Any attempts to access those variables will result in a NameError.**

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

→A return is a value that a function returns to the calling script or function when it completes its task. A return value can be any one of the four variable types: handle, integer, object, or string. The type of value your function returns depends largely on the task it performs.

→A Python function will always have a return value. There is no notion of procedure or routine in Python. So, if you don't explicitly use a return value in a return statement, or if you totally omit the return statement, then Python will implicitly return a default value for you.

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

→Every function in Python returns something. If the function doesn't have any return statement, then it returns None .

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

→The global Keyword

> **Normally, when you create a variable inside a function, that variable is local, and can only be used inside that function. To create a global variable inside a function, you can use the global keyword**.

10. What is the data type of None?

→ NoneType

→ The None keyword is used to define a null value, or no value at all. None is not the same as 0, False, or an empty string. None is a data type of its own (NoneType) and only None can be None.

11. What does the sentence import areallyourpetsnamederic do?

→ `ModuleNotFoundError: No module named 'areallyourpetsnamederic'`

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

→This function can be called with spam. bacon().

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

→If an error occurs in a program, we don't want the program to unexpectedly crash on the user. Instead, error handling can be used to notify the user of why the error occurred and gracefully exit the process that caused the error.

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

→ 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.