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

ANSWER: The use of functions helps the readability of a programme. Frequently, reading a large programme is challenging. By segmenting the code into smaller functions, the programme remains well-organized, comprehensible, and reusable. After being defined, the function can be utilised countless times.

1. 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 the function is called, not when the function is specified.

1. What statement creates a function?

ANSWER:  ‘**def’** keyword creates a function.

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

ANSWER: A function is a piece of code which helps the readability of a programme. It means that piece of code need not be written again. A function call means calling that function. Unless a function is called there is no use of that function.

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

ANSWER: Per programme execution, only one global Python scope is allowed. This scope continues to exist after the programme ends and all of its names are lost.

The scope of local variable is limited to the function where it is defined.

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

ANSWER : Each call of the function creates new local variables, and their lifetimes expire when the function returns to the caller.

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

ANSWER: 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.

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

ANSWER: If no return statement appears in a function definition, control automatically returns to the calling function after the last statement of the called function is executed. In this case, the return value of the called function is undefined.

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

ANSWER: By the help of "global" keyword.

1. What is the data type of None?

ANSWER: 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?

ANWER: That import statement imports a module named areallyourpetsnamederic. (This isn't a real Python module, by the way.)

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

ANSWER: This function can be called with spam. bacon()

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

ANSWER: Exception handling helps to save a programme from crashing if it encounters an error.

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

ANSWER: 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.