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

**Ans:**

* It helps to divide the large programs into small groups so that we can read the code, and debug the program faster and better.
* Python Functions stop us from writing the same logic various times. We can bind the logic in one function and then call the same over and over.
* Many persons can work on the same program by assigning different functions to each of them.
* It encourages us to call the same function with different inputs over multiple times.

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

**Ans:** The code executes when **it’s** called.

3. What statement creates a function?

**Ans:** def statement creates a function.

**def** name (list of parameters):

write logic here

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

**Ans:** Using a function to do a particular task any point in program is called as function call. So the difference between the function and function call is, A function is procedure to achieve a particular result while function call is using this function to achieve that task

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

**Ans:** There is one global scope, and a local scope is created whenever a function is called.

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

**Ans:** When a function returns, the local scope is destroyed, and all the variables in it are forgotten.

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

**Ans:** A return value is the value that a function call evaluates to. Like any value, a return value can be used as part of an expression.

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

**Ans**: If there is no return statement for a function, its return value is None.

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

**Ans:** A global statement will force a variable in a function to refer to the global variable

10. What is the data type of None?

**Ans:** The data type of None is NoneType.

11. What does the sentence import areallyourpetsnamederic do?

**Ans:** 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?

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

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

**Ans:** Place the line of code that might cause an error in a try clause.

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

**Ans**: The code that could potentially cause an error goes in the **try** clause.

The code that executes if an error happens goes in the **except** clause.