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

Ans. Function increases granularity and hence more effective testing.

2ndly, if a specific requirement is implemented using specific function, then this specific portion of the code can be reused, wherever there is same requirement. This reuse can be by a simple function call, instead of rewriting a complete code. Thus use of function reduces the program length. This makes program shorter, easier to update and easier to read.

3rdly using function pointer dynamic memory can be allocated and freed for a specific code.

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

Ans. Code in a function runs, when its called.

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

Ans. Let’s say iNeuron() is a function. Then following statement creates a function:

def iNeuron():

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

Ans. Function call informs the compiler / interpreter to go to the definition of the function. Once program encounters function call, compiler / interpreter loads the present register values and current program counter value into the stack memory. After loading the values, it loads the program counter value with the memory location, where the function definition is placed. After completion of execution of the called function, again compiler / interpreter retrieves the data stored in stack location to the program counter and respective registers.

Thus function call is an indication to the compiler / interpreter to process the function. By receiving the process request, compiler / interpreter processes the function definition of the requested function.

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

Ans. The part of a program where a variable is accessible is called its scope. There are four major types of variable scope and is the basis for the **LEGB rule**. LEGB stands for **Local -> Global -> Built-in**.

**Local Scope**: Whenever you define a variable within a function, its scope lies ONLY within the function. It is accessible from the point at which it is defined until the end of the function and exists for as long as the function is getting executed. Which means its value cannot be changed or even accessed from outside the function.

**Global Scope**: This is perhaps the easiest scope to understand. Whenever a variable is defined outside any function, it becomes a global variable, and its scope is anywhere within the program. Which means it can be used by any function.

**Built-in Scope**: This is the widest scope that exists! All the special reserved keywords fall under this scope. We can call the keywords anywhere within our program without having to define them before use.

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

Ans. When the function call returns, means when the program counter holds back the address of calling location, now onwards, the variable becomes undefined. So any use of this variable in this scope will give rise to “NameError”

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

Ans. If its expected to receive some value from a function, then the function must return a value. Once a value is returned, the called location must hold the value in a variable of same datatype as that of returned value.

Yes, Its possible to have return value in an expression. Example;

a, b = 10, 20

min = (b, a) [a < b]

min value is the minimum between a and b.

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

Ans. 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. That default return value will always be None.

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

Ans. 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, we can use the global keyword.

**10. What is the data type of None?**

Ans. It is a data type of the class NoneType object.

**11. What does the sentence import areallyourpetsnamederic do?**

Ans. Import statement imports areallyourpetsnamedericmodule.

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

Ans. Import spam

Spam.bacon()

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

Ans. Put the line of code that may cause error, shall be kept under try clause.

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

Ans. The line of code, that is suspected to cause error shall be kept under try clause and the line of codes, that should convey the error message due to a specific error shall be kept under except clause.