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

The advantages of Python functions are as follows:

* With the help of functions, we can avoid rewriting the same logic or code again and again in a program.
* In a single Program, we can call Python functions anywhere and also call multiple times.
* We can track a large Python program easily when it is divided into multiple functions.
* The main achievement of Python functions is its Reusability.

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?

In Python, you define a function with the def keyword, then write the function identifier (name) followed by parentheses and a colon. The syntax is:

def functionName():

# What to make the function do

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

A function is a block of code that does a particular operation and returns a result. It usually accepts inputs as parameters and returns a result. The parameters are not mandatory.

E.g:

Function add(a,b)

return a+ b

A function call is the code used to pass control to a function.

E.g.:

b = add(5,6)

Now b will have the value 11.

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

There's only one global Python scope per program execution. This scope remains in existence until the program terminates and all its names are forgotten. Otherwise, the next time you were to run the program, the names would remember their values from the previous run.

Local (or function) scope is the code block or body of any Python function or lambda expression. This Python scope contains the names that you define inside the function. These names will only be visible from the code of the function. It’s created at function call, not at function definition, so you’ll have as many different local scopes as function calls. This is true even if you call the same function multiple times, or recursively. Each call will result in a new local scope being created.

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

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?

A return statement is used to end the execution of the function call and “returns” the result (value of the expression following the return keyword) to the caller. The statements after the return statements are not executed. If the return statement is without any expression, then the special value None is returned. A return statement is overall used to invoke a function so that the passed statements can be executed.

‘return’ cannot be used outside function. So it is not possible to have a return value in an expression.

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

In Python, it is possible to compose a function without a return statement. Functions like this are called void, and they return None, Python's special object for "nothing".

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

Variables that are declared outside of a function are known as global variables. You can access global variables in Python both inside and outside the function.

**Code**:

x = “This global variable” #Global variable declaration

def myfunc():

print(“We are going to print: ” + x) #Refering to the global variable

myfunc()

10. What is the data type of None?

None is a data type of its own (NoneType) and only None can be None.

11. What does the sentence import areallyourpetsnamederic do?

The sentence import areallyourpetsnamederic imports the library called ‘areallyourpetsnamederic’.

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

import spam

spam.bacon()

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

We incorporate exception handling and logging into the programme to handle and log the errors for future debugging process, respectively.

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.