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

The Functions plays a very important role in building a program for anything. Functions helps the coder to finish the program by providing a easy to use functionality of using it whenever he/she requires.

Some of the basic features of the functions are:

1. **Code Reusability**: Functions provide the reusability for the code. Once a code is written, we can reuse it with the help of Functions.
2. **Clean Programs**: Functions help the programs to be clean and short by enabling the code to be written only one.
3. **Complex problems to the simpler problems**: Functions help the complex programs to be divided into simpler problems.
4. **When does the code in a function run: when it's specified or when it's called?**

The function is run when it is **called** as we are able to **see the output when the function is called**.

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

In python keyword “**def** “ is used to create the function.

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

**Function** is a series steps that are supposed to be repeated in a program. So these steps are defined under a particular name and it is defined by keyword “**def**”.

A **function call** is used to run a function which is already defined.

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

There can be only **one global** active scope at a time and for **local** there can be **four** such scopes.

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

The local variable in a function only exists when a function is being executed. When the function terminates itself, the local variables are destroyed itself.

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

Return value means that the result is returned at the end of a function call by using a keyword called “**return**”. We can return an expression as a return value but can’t have a return value in an expression.

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

If a function is not having any return value, then it may return a string in the form of a print statement OR it may return a expression as a print statement.

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

We can simply call the variable name inside the definition of the function. The keyword **global** is used to make any variable as a global variable.

For example:

Global x=20

def simo():

if x>10:

return “the value for x is lower than expected”

else:

return “The value for is sufficient to proceed further”

simo()

**OUTPUT:**

The value for is sufficient to proceed further

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

The data type for None is NoneType.

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

If we have defined this areallyourpetsnamederic as a library or as a function then we can import this. Otherwise it will throw a error as there is no such python library.

If we have defined areallyourpetsnamederic as a library then **import areallyourpetsnamederic** will import the areallyourpetsnamederic library.

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

We will first create a object for the spam module and then we will use objectname.bacon() to execute this bacon() feature.

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

I use “**try and except**” statements to prevent my python code from crashing.

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

The purpose of the try clause is that the code inside the try block will get executed only if there is no error.

The purpose of except block is that the code inside this block will get executed when there is an error in the try block. Moreover, we can also catch the error in the try block in this except block.