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

**Ans:**

If we need to do a particular operation with a lot of variables, then a function helps us to just write the required operation once in the function description then we can pass all the variables as the arguments of the function one by one or by using a loop and get the whole task done in one go without defining the operation every time.

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

**Ans:**

when it's called

3. What statement creates a function?

**Ans:**

def *function\_*name(variable*(s*)*)*:

{

“*operation to be done*”

}

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

**Ans:**

Function: defining the name of the function and code what’s to be done by that function

def *function\_*name(variable*(s*)*)*:

{

“*operation to be done*”

}

Function Call: calling a function to use it’s operations on some values/values variables stored in variables

*function\_*name(*values/variable*)

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

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

7. What is the concept of a return value? Is it 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?

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

10. What is the data type of None?

11. What does the sentence import areallyourpetsnamederic do?

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

13. What can you do 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?