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

Ans: Functions shorten the program code length and they can be called from anywhere in or outside the program.

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

Ans: When it’s called the code in function executes.

3. What statement creates a function?

Ans : def statement creates a function.

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

Ans: Function is a separate part of program where it is declared and defined.

Function call is to call that function by passing desired values and after its execution it returns value.

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

Ans: There is one global and one local scope.

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

Ans: When function call returns the local variables scope finishes.

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

Ans: The return value is value which evaluate by a function. Yes it is possible to 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?

Ans: None.

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

Ans: By the help of global statement.

10. What is the data type of None?

Ans: None type.

11. What does the sentence import areallyourpetsnamederic do?

Ans: The import statement imports the module areallyourpetsnamederic.

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

Ans: spam.bacon().

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

Ans: In that case we can use the code in try clause.

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

Ans: The code which might cause an error will come under try clause.

After executing the code if any error occurs then it will come under except clause.