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

ANS- to create modular code and it also enhance code reusability.

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

ANS- when it’s specified.

3. What statement creates a function?

ANS- “def” statement create a function.

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

ANS- function is the block of code use to perform some task while function call use to pass some parameter to the function.

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

ANS- there can be a single global scope per code execution and there can be many local scope for defining a function.

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

ANS-it will execute the function

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

ANS- return value execute the operation by calling the code. No.

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

ANS-it will pass to none type function

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

ANS- by the declairing the the value out side the function.

10. What is the data type of None?

ANS- none data type

11. What does the sentence import areallyourpetsnamederic do?

ANS- it will import areallyyourpetsnamederic module which is not really a python module.

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

ANS- this function can be called with **spam. bacon()**

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

ANS- error handeling can be used.

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

ANS- The try block lets you test a block of code for errors. The except block **lets you handle the error**.