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

**Answer:**
By using functions, we can avoid rewriting the same logic/code again and again in a program. We can call functions any number of times in a program and from any place in a program. We can track a large program easily when it is divided into multiple functions. Reusability is the main achievement of functions.

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

**Answer:**
The code in a function executes when the function is called, not when the function is defined.

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

**Answer:**
A function is defined by using the `def` keyword, followed by a name of our choosing, followed by a set of parentheses which hold any parameters the function will take (they can be empty), and ending with a colon(:).
Example:

```
def addition(a,b):
    return a+b
```

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

**Answer:**
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.
Calling a function means invoking a function so that it performs the required task.When we call a function then the control passes to the function and the statements written therein are executed one by one there by carrying out the required task.

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

**Answer:**
A variable is only available from inside the region it is created. This is called scope.

A variable created inside a function belongs to the ***local scope*** of that function, and can only be used inside that function.
Example:

```
def myfunc():
  x = 300
  print(x)
myfunc()
```

If we operate with the same variable name inside and outside of a function, Python will treat them as two separate variables, one available in the **global scope** (outside the function) and one available in the local scope (inside the function).

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

**Answer:**
A local variable retains its value until the next time the function is called, a local variable becomes undefined after the function call completes.

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

**Answer:**
Return values are the values that a function returns when it has completed.
We can use that value in a math expression or any other kind of expression in which the value has a logical or coherent meaning.
Return statements can not be used outside the function.

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

**Answer:**
In Python, every function returns something. If there are no return statements, then it returns None.

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

**Answer:**
When we create a variable inside a function, that variable is local, and can only be used inside that function. To create a global variable inside a function, we use the "global" keyword.

```
def func():

  x = "world"

func()

print("hello " + x)
```

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

**Answer:**

The None keyword is used to define a null value, or no value at all. None is not the same as 0, False, or an empty string. None is a data type of its own (NoneType) and only None can be None.

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

**Answer:**

That import statement imports a module named areallyourpetsnamederic.

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

**Answer:**

This function can be called with spam.bacon().

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

**Answer:**

We use the `try` and `except` statements to handle exceptions. Whenever the code breaks down, an exception is thrown without crashing the program.

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

**Answer:**

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