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

**Answer:**

With the help of functions, we can avoid rewriting the same logic or code again and again in a program. In a single Program, we can call Python functions anywhere and also call multiple times.

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

**Answer:**

A function must be defined before the function call.   
otherwise, the Python interpreter gives an error.   
To call the function, we use the function name followed by the parentheses.

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

**Answer:**

define a function with the def keyword, then write the function identifier (name) followed by parentheses and a colon.

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

**Answer:**

A function is a piece of code which enhanced the reusability and modularity of your program. It means that piece of code need not be written again. A function call means invoking or calling that function. Unless a function is called there is no use of that function.

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

**Answer:**

1. **Global scope:** There’s only one global Python scope per program execution. This scope remains in existence until the program terminates and all its names are forgotten. Otherwise, the next time you were to run the program, the names would remember their values from the previous run.
2. **Local scope:** The names that you define in this scope are only available or visible to the code within the scope.

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

**Answer:**

A variable declared inside a function has a function local scope. It has been allocated memory when the function is called, and once the function returns something, the function execution ends and with it, the variable goes out of scope, i.e. it gets **deleted from the memory**.

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

**Answer:**

* The **Python return statement** is a special statement that you can use inside a function  to send the function’s result back to the caller. A return statement consists of the return keyword followed by an optional **return value**.  
  The return value of a Python function can be any Python object. Everything in Python is an object. So, your functions can return numeric values (int, float, and complex values), collections and sequences of objects (list, tuple, dictionary, or set objects), user-defined objects, classes, functions, and even modules or packages.
* **Yes,** return value in any expression:

def funName():

Return 2

print(funName() \* 2)

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

**Answer:**

 If the function doesn't have any return statement, then it returns None

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

**Answer:**

To create a global variable inside a function, you can use the global keyword

def myfunc():  
  global x  
  x = "somevalue"

**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:**

Place the line of code that might cause an error in a try clause.

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

**Answer:**

The code that could potentially cause an error goes in the try clause.

The code that executes if an error happens goes in the except clause.