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

Functions helps in reducing the need of using at multiple places. It makes program shorter, easier to read, and easier to update.

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

The code in function runs only when it is called.

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

“def” statement creates a function.

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

A function starts with def statement and the code which fulfils the definition of the function.

A function call is what moves the program execution into the function and the function call evaluates to the function’s return value.

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

At any given time during execution, We’ll have at most four active Python scopes—local, enclosing, global, and built-in—depending on where we are in the code. On the other hand, we’ll always have at least two active scopes, which are the global and built-in scopes.

So, there is one global scope and a local scope is created whenever a function is called.

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

When a function returns, the local scope is destroyed and all the variables in it are forgotten.

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

A return value is what the function call evaluates to.

Yes, it’s possible to have a return value in a expression.

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

Its return value is None.

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

Normally, when we create a variable inside a function, that variable is local, and can only be usedinside that function.

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

Ex:

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

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

NoneType

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

It simply imports a module named areallyourpetsnamederic. But it’s actually not a real python module. It won’t give any error while importing.

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

spam.bacon()

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

lines of code that might cause an error and suspicious, place them in a try clause

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

The try block lets you test a block of code for errors.

The except block lets you handle the error.

**try**: the code with the exception(s) to catch. If an exception is raised, it jumps straight into the except block.

**except**: this code is only executed if an exception occured in the try block. The except block is required with a try block, even if it contains only the pass statement.