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

**Ans:**

Functions are very useful for many reasons. Usually, function is bunch of lines of code which some purpose. When you define such a function, you can use the same function whenever it necessary. This reduces the duplication of code. And it improves the reusability of the code as we will same code defined in the function again and again without writing it repeatedly. Also, it makes our code clearer as it’s like you are replacing bunch of lines of codes with a single line of code.

Also, as function is defined with specific purpose, this is good practice as it divides the big problem into a smaller one.

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

**Ans:**

When we define a function, python store this information as an object to be precise it is function object. And when you call a function, that time this function performs the operation it was supposed to. And hence, at this point we can call that a function is ran.

*3. What statement creates a function?*

**Ans:**

A function is defined using “def” keyword. The syntax is as follows:

def <function name>(<arguments>):

<function code>

Here, the function name has same rules as we have for the variable names. Usually, function name should not start with capital letters.

Also, we have one liner function called lambda function. The syntax is as follows:

<function name> = lambda <arguments>: <function code>

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

**Ans:**

To call function, first it must be defined. So, when you define a function using “def” keyword, it is called a function object and it will be stored in the memory. After defining the function, now you can call it as many times you want obviously with arguments as mentioned while defining it.

# Function definition

def do\_something():

return

# Function call

do\_something()

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

**Ans:**

There is only one global scope in the function and where we run the code is called global scope. This can be checked using \_\_name\_\_ variable. This parameter will have value of ‘\_\_main\_\_’ if you are in global scope.

In contrast, local scopes can be in any number. Local scopes are basically associates with functions and classes. With each call made to the function and classes a new or separate local scope is created and all variables created in that local scope are only accessible to that local scope only.

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

**Ans:**

When you call a function, the arguments are passes to that function and function does the task what it is defined to do. In the process, it creates the variables as defined. And at the end, it returns the variable as defined. When the variables are returned, and function call is completed. The variable from local scope is collected by the garbage collector of python and if not used further they will be deleted.

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

**Ans:**

A function can return any number of variables. And it is possible to use the return of function directly in the expression.

Ex. 2 + get\_square(3) \* 2 🡪 consider get\_square returns a float or int, it can be used directly in the expression.

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

**Ans:**

If nothing is returned from function, the “None” object is returned by default.

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

**Ans:**

Global variables are directly accessible from function which are defined in the global scope.

Ex:

num1 = 100

def do\_something():

num2 = num1 + 100

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

**Ans:**

None itself is a data type. This is special data type.

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

**Ans:**

It imports the module or packages named *areallyourpetsnamederic.*

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

**Ans:**

It will be called a method.

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

**Ans:**

You can make use of try and except block to catch that error and show some helpful message.

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

**Ans:**

In try clause, you can write a code that you may thing can cause an error, possibly crash the program or raise an error difficult to understand by the user.

Many except clause can be defined to catch different types of errors and some appropriate tasks in each except clause. If the try clause encounters some error, it finds the except clause which matches with the error encountered. If the error matches, then it executes the code from that except clause.