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

* **Python Functions**is a block of code that return the specific task.
* **In a python function we can put the block of code which is** repeatedly **used in**

**program for different input values.**

* **So instead of writing a same code again and again for different inputs, we can put it in a function and reuse it whenever required by making a function call.**
* **Once the function is defined we can call it** many times and from anywhere in a program.
* If our Python program is large, it can be separated into numerous functions which is simple to track.

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

Code in the function is run only when it is called.

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

* In python, keyword “def” is used to create a function followed by function name followed by parameters and colon (:)
* We can pass arguments to the defined function using parameters.
* We can use a return expression to return a value from a defined function.

**Syntax:** def function\_ name(parameters):

# Block of code

return expression

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

* 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.

**Syntax:** def function\_ name(parameters):

# Block of code

return expression

* A function call is the code used to pass control to a function.
* If the function has been defined to receive parameters, the values that are to be sent into the function are listed inside the parentheses of the function call operator.
* When a program calls a function, the program control is transferred to the called function.

**Syntax:** function\_name(arguments)

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

The scope of a variable in python is that part of the code where it is visible.

Four types of scopes in python:

**Local scope**- variables declared inside function can only be utilized inside the function.

**Global scope**-variables declared outside any other python variable scope and hence can be utilized anywhere in the program.

**Enclosed scope** –variables which are not global or local, e.g,

def red():

a=1

def blue():

b=2

print(a)

print(b)

blue()

print(a)

red()

In this code, ‘b’ has local scope in function ‘blue’, and ‘a’ has nonlocal scope in‘blue’.

**Built in scope**- The built-in scope has all the names that are loaded into python variable scope when we start the interpreter.

e.g print(), id()

There are 1 global scope and 4 local scopes in python program.

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

The local variables are destroyed when the function call returns and memory occupied by them

are freed for any other variables.

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

* A return is a value that a function returns to the calling script or function when it completes its task.
* A return value can be any one of the four variable types: handle, integer, object, or string.
* Yes, It is possible to have a return value in an expression.

def greater\_than\_1(n):

return n > 1

print(greater\_than\_1(1))

False

print(greater\_than\_1(2))

True

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

None

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

We can use “global” keyword before the variable defined inside the function.

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

None is a data type of its own( NoneType) and only None can be (None).None keyword is an object, and it is a data type of the class NoneType

x = None

print(x)

print(type(x))

**Output :** None

<class 'NoneType'>

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

This sentence imports the module “areallyourpetsnamederic “

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

spam.balcon()

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

Error handling can be used to notify the user of why the error occurred and gracefully exit the process that caused the error

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

* Errors in Python can be of two types i.e. Syntax errors and Exceptions.
* Try and Except clause is used to handle these errors within our code in Python.
* The try block is used to check some code for errors i.e the code inside the try block will execute when there is no error in the program.
* Whereas the code inside the except block will execute whenever the program encounters some error in the preceding try block.
* A try statement can have more than one except clause
* If any exception occurs, but the except clause within the code doesn’t handle it, it is passed on to the outer try statements.
* If the exception is left unhandled, then the execution stops.

**Syntax:**

try:

# Some Code

except:

# Executed if error in the try block

**Python code to illustrate**

def divide(x, y):

    try:

              result = x // y

        print("Division :", result)

    except ZeroDivisionError :

        print("You are dividing by zero ")

divide(3, 2)

Output: Division : 1

**Python code to illustrate**

def divide(x, y):

    try:

        result = x // y

        print("Division :", result)

    except ZeroDivisionError:

        print("You are dividing by zero ")

divide(3, 0)

Output: You are dividing by zero

**It may be combined with the else** **and finally** **keywords.**

* **else**: Code in the else block is only executed if no exceptions were raised in the try block.
* **finally**: The code in the finally block is always executed, regardless of if a an exception was raised or not.