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

**Ans)** By creating a function we can do a particular action multiple times by just calling it. Code becomes modular and understandable when functions are implemented in the code.

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

**Ans)** The code in the function will only run when it is called. A function call is necessary to run the code in it.

1. What statement creates a function?

**Ans)** **def** statments creates a function. The syntax for creating a function is **def function\_name(arguments): .**

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

**Ans)** A function is the one that contains all the code in it where a function call is the way that the code in the function is executed. **def** is the statement to declare a function. **function\_name( )** is the syntax to call a function.

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

**Ans)** **Global Scope:** The global scope refers to the top-level scope of a Python program. It is the outermost scope and is accessible throughout the entire program.

**Local Scopes:** Local scopes are created whenever a function or a code block is executed. These scopes are temporary and exist only during the execution of the function or code block.

A Python program typically has one global scope that is accessible throughout the program, and multiple local scopes created whenever functions or code blocks are executed. Variables defined in the global scope can be accessed globally, while local variables are confined to their respective scopes.

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

**Ans)** The variables in the local scope are destroyed or cannot be accessed outside of the function unless they are declared globally.

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

**Ans)** The concept of a return value in programming refers to the value that a function provides as output when it is called and executed. When a function is designed to return a value, it can compute a result or generate data that can be useful for further processing or assignments.

Expression with Return Value: Yes, it is possible to have a return value in an expression. Since a function call that returns a value can be considered an expression itself, the return value can be directly used within a larger expression.

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

**Ans)** If a function does not have a return statement, or if it has a return statement without any value specified, the return value of a call to that function is **None**. **None** is a special value in Python that represents the absence of a value or a null value.

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

**Ans)** In Python, if you want to make a function variable refer to a global variable, you can use the **global** keyword. The **global** keyword allows you to access and modify a global variable from within a function.

**Example:**

global\_var = 10 # Global variable

def access\_global\_variable():

global global\_var # Declare the variable as global

function\_var = global\_var # Assign global variable value to function variable

print("Function Variable:", function\_var)

access\_global\_variable() # Output: Function Variable: 10

1. What is the data type of None?

**Ans)** The data type of None in Python is **NoneType**. **NoneType** is a built-in data type in Python that has a single value, which is **None**. It represents the absence of a value or a null value.

1. What does the sentence import areallyourpetsnamederic do?

**Ans)** The sentence import areallyourpetsnamederic is not a valid Python import statement and would raise a **ModuleNotFoundError**. In Python, the **import** statement is used to import modules or packages that contain reusable code.

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

**Ans)** After importing spam as **import spam .** Bacon( ) function can be accessed by **spam.bacon( ).**

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

**Ans)** To save a program from crashing when it encounters an error, you can use error handling techniques to catch and handle exceptions. Python provides built-in mechanisms for handling exceptions, which allow you to gracefully manage errors and prevent your program from abruptly terminating.

Example :

try:

# Code that may raise an exception

except ExceptionType:

# Code to handle the exception

1. What is the purpose of the try clause? What is the purpose of the except clause?

**Ans)** The **try** and **except** clauses are used together in Python for error handling, allowing you to handle exceptions gracefully and prevent your program from crashing.

**Purpose of the try clause:** The try clause is used to enclose the code that may potentially raise an exception. It defines a block of code where you anticipate an error or exception might occur. The purpose of the try clause is to mark the section of code that needs to be protected and monitored for exceptions.

**Purpose of the except clause:** The except clause is used to specify how to handle specific exceptions that occur within the corresponding try block. It allows you to define the code that should execute when a particular exception is raised.