**6 ) Class and Object (OOP Concepts)**

* **Understanding the concepts of classes, objects, attributes, and methods in Python.**

1. **Class**

A class is a blueprint for creating objects. It defines the structure (attributes and methods) that the objects will have.

class Student:

name = ""

age = 0

1. **Object**

An object is an instance of a class. When a class is defined, no memory is allocated until an object of that class is created. Objects are used to access the attributes and methods defined in the class.

s1 = Student()

s1.name = "Rahul"

s1.age = 20

1. **Attributes**

Attributes are variables that belong to a class or an object. They store information about the object. Attributes can be defined directly inside the class or initialized through a constructor (the \_\_init\_\_() method).

class Student:

def \_\_init\_\_(self, name, age):

self.name = name # attribute

self.age = 20

1. **Methods**

Methods are functions defined inside a class that operate on objects of that class. They can access and modify object attributes using the self keyword.

class Student:

def \_\_init\_\_(self, name, age):

self.name = name

self.age = age

def display(self): # method

print(f"Name: {self.name}, Age: {self.age}")

* **Difference between local and global variables.**

1. **Local Variables**

A local variable is a variable that is declared inside a function and is accessible only within that function. It is created when the function starts and destroyed when the function ends.

* Declared inside a function.
* Accessible only within that function.
* Created when the function is called.
* Destroyed when the function ends.
* Cannot be used outside the function.

def my\_function():

x = 10 # local variable

print("Local x:", x)

my\_function()

1. **Global Variables**

A global variable is declared outside all functions and is accessible throughout the program, including inside functions (if not redefined there). It has a global scope.

* Declared outside all functions.
* Accessible throughout the entire program.
* Exists as long as the program is running.
* Can be accessed inside functions.
* To modify a global variable inside a function, use the global keyword.

x = 50 # global variable

def my\_function():

print("Global x inside function:", x)

my\_function()

print("Global x outside function:", x)