**Que.1-Understanding the concepts of classes,objects,attributes and methods in python.**

**Ans.**

* **Classes:~**

A class is a blueprint for creating objects(instances).it defines a set of attributes and methods that the created objects can use.classes allow you to structure your code in a reusable and organised way.

**Ex.,** class sum:

No1=45

No2=56

Result=No1+No2

Print(Result)

* **Objects:~**

An object is an instance of a class. When you create an object, you call the class itself as if it were a function. Each object can have different attributes while being created from the same class.

**Ex.,**Addition=sum()

* **Attributes:~**

Attributes are variables that belong to a class or an object. There are two types of attributes:

* Class attributes:

Belong to the class itself and are shared by all objects of the class.

* Instance attributes:

Belong to the object (instance) and can have different values for different objects.

* **Methods:~**

Methods are functions that belong to a class. They define the behaviors of an object. There are different types of methods:

* **Instance methods**: Operate on the instance of the class (the object).
* **Class methods**: Operate on the class itself and are defined with a @classmethod decorator.
* **Static methods**: Defined with a @staticmethod decorator and don't access or modify the class or instance attributes.

**Ex.,** class Dog:

species = "Canis familiaris"

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

self.name = name

self.age = age

# Instance method

def description(self):

return f"{self.name} is {self.age} years old."

# Another instance method

def speak(self, sound):

return f"{self.name} says {sound}"

# Creating an object

my\_dog = Dog("Buddy", 3)

# Calling methods

print(my\_dog.description()) # Output: Buddy is 3 years old.

print(my\_dog.speak("Woof")) # Output: Buddy says Woof

**Que.2-Diffrence between local and global variable.**

**Ans.**

**Local variables:~**

**Global variables:~**