Name : Usman Ul haq

Roll number : SU92-BSAIM-S24-035

Section : 2A

Semester : 2nd

Task = 01

**Class Concepts:**

1. Class vs. Object:

a. Explain the difference between a class and an object in Python.

b. Provide an example.

2. Constructor Method (\_\_init\_\_) vs \_\_str\_\_() Function:

a. Explain the difference between them in Python.

b. Provide an example

**CODE**

**1. Class vs. Object**

**a. Difference between a Class and an Object:**

* **Class**: A class is a blueprint or template for creating objects. It defines a set of attributes and methods that the objects created from the class can use. A class encapsulates data for the object and defines its behavior.
* **Object**: An object is an instance of a class. It is created using the class and represents a specific entity that can hold data (attributes) and perform actions (methods). Each object can have unique values for its attributes.

**: Example:**

class Dog:

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

self.name = name

self.age = age

def bark(self):

return "Woof!"

dog1 = Dog("Buddy", 3)

dog2 = Dog("Max", 5)

print(dog1.name)

print(dog2.age)

print(dog1.bark())

**2. Constructor Method (\_\_init\_\_) vs \_\_str\_\_() Function**

**a. Difference between \_\_init\_\_ and \_\_str\_\_:**

* **\_\_init\_\_ (Constructor)**: This is a special method in Python that is called when an object is created from a class. It allows the class to initialize its attributes with specific values at the time of object creation.
* **\_\_str\_\_ (String Representation)**: This is another special method that defines how an object should be represented as a string. When you use print() on an object or convert it to a string, the \_\_str\_\_() method is called to return a string representation of the object.

**:Example:**

class human:

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

self.name = name

self.age = age

def \_\_str\_\_(self):

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

human1 = human("Alice", 30)

print(human1)