

## OOP LAB TASK 2

( 1 )

### . Class vs. Object

#### *a. Difference Between a Class and an Object in Python*

- A **class** is a blueprint or template for creating objects. It defines properties (attributes) and behaviors (methods) that the objects created from it will have.
- An **object** is an instance of a class. It is a concrete realization of the class, with specific values assigned to the attributes defined in the class.

In simpler terms:

- **Class:** A general concept or definition.
- **Object:** A specific realization of that concept.
  - EXAMPLE:-

```
class Car:
    def __init__(self, make, model):
        self.make = make
        self.model = model

    def display_info(self):
        return f"Car Make: {self.make}, Model: {self.model}"
```

```
my_car = Car("Toyota", "Corolla")
```

```
print(my_car.display_info())
```

( 2 )

*a. Difference Between `__init__` and `__str__`*

- **`__init__` (Constructor Method):**
  - Used to initialize an object's attributes when it is created.
  - Automatically called when an object is instantiated.
  - Example: Setting the initial values for attributes.
- **`__str__` (String Representation Method):**
  - Used to provide a string representation of an object when `print()` or `str()` is used.
  - Helps in making the object output user-friendly.
  - Not called automatically when creating an object (unlike `__init__`).

EXAMPLE:-

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def __str__(self):
        return f"Person(Name: {self.name}, Age: {self.age})"
```

```
person = Person("Alice", 30)
```

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

```
print(person)
```