OOPs (Class and Objects)

What is OOP?

- OOP stands for **Object-Oriented Programming**.
- Procedural programming is about writing procedures or methods that perform operations on the data, while object-oriented programming is about creating objects that contain both data and methods.

Object-oriented programming has several advantages

- OOP is faster and easier to execute
- OOP provides a clear structure for the programs
- OOP helps to keep the code DRY "Don't Repeat Yourself", and makes the code easier to maintain, modify and debug
- OOP makes it possible to create full reusable applications with less code and shorter development time.

What are Classes and Objects?

- Classes and objects are the two main aspects of object-oriented programming.
- A class is a template for objects, and an object is an instance of a class.
- A Class is like an object constructor, or a "blueprint" for creating objects.

class	objects
Fruit	Apple Banana Mango

class	objects
Car	Volvo Audi Toyota

In Java

- Java is an object-oriented programming language.
- Everything in Java is associated with classes and objects, along with its attributes and methods.
- For example: in real life, a car is an object. The car has **attributes**, such as weight and color, and **methods**, such as drive and brake.

```java
// Person's Class
public class Class_Object {
    // Class variables
    String name;
    int age;

    // Main method
    Run | Debug
    public static void main(String[] args) {
        // Creating objects
        Class_Object person1 = new Class_Object();
        Class_Object person2 = new Class_Object();

        // Assigning values to object properties
        person1.name = "Alice";
        person1.age = 25;

        person2.name = "Bob";
        person2.age = 30;

        // Displaying object properties
        System.out.println("Person 1: " + person1.name + ", Age: " + person1.age);
        System.out.println("Person 2: " + person2.name + ", Age: " + person2.age);
    }
}
```

In Python

```python
class Person:
    pass  # Empty class

# Creating objects
person1 = Person()
person2 = Person()

# Assigning values to object properties
person1.name = "Alice"
person1.age = 25

person2.name = "Bob"
person2.age = 30

# Displaying object properties
print("Person 1:", person1.name, ", Age:", person1.age)
print("Person 2:", person2.name, ", Age:", person2.age)
```

Task

1. Student Information: Create a class Student to represent student information with properties like name, rollNumber, and grade. Create objects for two students and display their information.
2. Book Details: Define a class Book with properties title, author, and ISBN. Create two book objects and print their details.
3. Rectangle Dimensions: Implement a class Rectangle to store the dimensions (length and width) of a rectangle. Create two rectangle objects and display their dimensions.
4. Car Information: Create a class Car with properties like make, model, and year. Instantiate two car objects and print their details.
5. Employee Records: Define a class Employee to store employee information, including name, designation, and salary. Create two employee objects and display their records.