DAY 4

Java Classes:

What is a Class?

In Java, a **class** is a blueprint or a template for creating objects. It defines the properties (or **attributes**) and behaviors (or **methods**) that an object will have. Think of a class as a blueprint for a house: it outlines the number of rooms, windows, and doors, but it isn't the physical house itself. An object is the actual house built from that blueprint.

How to Create a Class

You create a class by using the class keyword followed by the class name. It's a best practice to capitalize the first letter of the class name.

Java

```
// This is the blueprint for a 'Car'
public class Car {
  // These are the attributes (properties) of a Car
  String color;
  String model;
  int year;

  // This is a method (behavior) of a Car
  public void startEngine() {
    System.out.println("The engine has started.");
  }
}
```

Members of a Class

The members of a class are the variables and methods that belong to it.

- **Variables:** These are the **attributes** or **fields** that define the state of an object. In the Car example, color, model, and year are the variables.
- **Methods:** These are the **behaviors** or **functions** that define what an object can do. In the Car example, startEngine() is a method.

DAY 4

Creating a Class in Eclipse

When creating a class in an Integrated Development Environment (IDE) like Eclipse, you are usually presented with several options:

- **Package:** The package is a way to organize your classes into logical groups.
- **Modifiers:** You can select modifiers like public, private, or abstract to control access to the class.
- **Superclass:** You can specify a parent class from which your new class will inherit.
- **Interfaces:** You can choose to implement one or more interfaces.
- **Stubs:** Eclipse can automatically generate method stubs, such as a main() method, constructors, or overridden methods.

Interview Questions on Classes

1. **What is the difference between a class and an object?**
    - **Class:** A class is a blueprint or a template. It defines the structure and behavior.
    - **Object:** An object is an instance of a class. It is a concrete entity created from the class blueprint.
2. **What are the members of a class?**
    - The members of a class are its **variables** (also known as fields or attributes) and its **methods** (also known as functions or behaviors).
3. **Explain the purpose of a class in Java.**
    - The primary purpose is to act as a template for creating objects. It allows you to model real-world concepts in your code and is the foundation of **Object-Oriented Programming (OOP)**.
4. **Can a class be empty?**
    - Yes, a class can be defined without any variables or methods. It would simply be a blueprint that can't do anything yet.
5. **What is a package in Java, and why is it important when creating a class?**
    - A package is a mechanism for organizing related classes and interfaces. It helps to prevent naming conflicts and provides a way to control access to classes.