DAY 13

What is an Object?

An **object** is an instance of a class. It's a real-world entity that has a **state** (data/variables) and **behavior** (methods). Think of a class as the general idea of a "car," and an object as your specific "blue Ford Mustang."

- **Real-time use case:** In an e-commerce application, a Product class defines what a product is (name, price, ID). An object is a specific product, like product1 with the values name="Laptop", price=1200.00, and ID=101.

How Do We Create an Object?

You create an object using the new keyword, which allocates memory for the new object and calls its constructor to set its initial state. This process is called **instantiation**.

- **Syntax:** ClassName objectName = new ClassName();
- **Code Example:**

Java

```
public class Car {
    String model;
    int year;
}


public class Main {
    public static void main(String[] args) {
        // Creating an object of the Car class
        Car myCar = new Car();
        myCar.model = "Toyota";
        myCar.year = 2023;


        System.out.println("My car is a " + myCar.year + " " + myCar.model);
    }
}
```

DAY 13

What an Object Can Contain in It?

An object contains two main things:

1. **Instance Variables (State):** These are the data members that hold the object's unique state. For a Car object, this would be its color, model, and speed. Each object of the Car class will have its own copy of these variables.
2. **Methods (Behavior):** These are the actions or functions that the object can perform. For a Car object, this would include startEngine(), accelerate(), and brake().

- **Real-time use case:** A User object in a social media app would contain instance variables like username, email, and profilePicture, and methods like postStatus(), sendMessage(), and addFriend().

How to Call Methods and Variables?

You use the **dot operator (.)** to access the members of an object. The way you access them depends on whether you're inside or outside the class.

1. **Within the Class:** You can directly use the variable or method name.
2. **Outside the Class:** You must use an object reference followed by the dot operator.

- **Code Example:**

Java

```
public class Dog {
  String name; // Instance variable
  public void bark() { // Method
    System.out.println("Woof!");
  }
}
public class Main {
  public static void main(String[] args) {
    Dog myDog = new Dog();
    myDog.name = "Buddy"; // Accessing an instance variable
    myDog.bark();       // Calling a method
  }
}
```

DAY 13

Common Interview Questions

1. **What is the difference between a class and an object?**
   - **Answer:** A class is a blueprint or template, while an object is a concrete instance of that class. You create an object from a class.
2. **How do you create an object in Java?**
   - **Answer:** You use the new keyword followed by the class's constructor. For example, MyClass myObject = new MyClass();.
3. **What does an object contain?**
   - **Answer:** An object contains instance variables (its state) and methods (its behavior).
4. **What is the purpose of the new keyword?**
   - **Answer:** The new keyword is used to create an instance of a class by allocating memory for the new object and returning a reference to it.
5. **How do you access the members (variables and methods) of an object?**
   - **Answer:** You use the dot operator (.) on the object reference. For example, myObject.myVariable or myObject.myMethod().