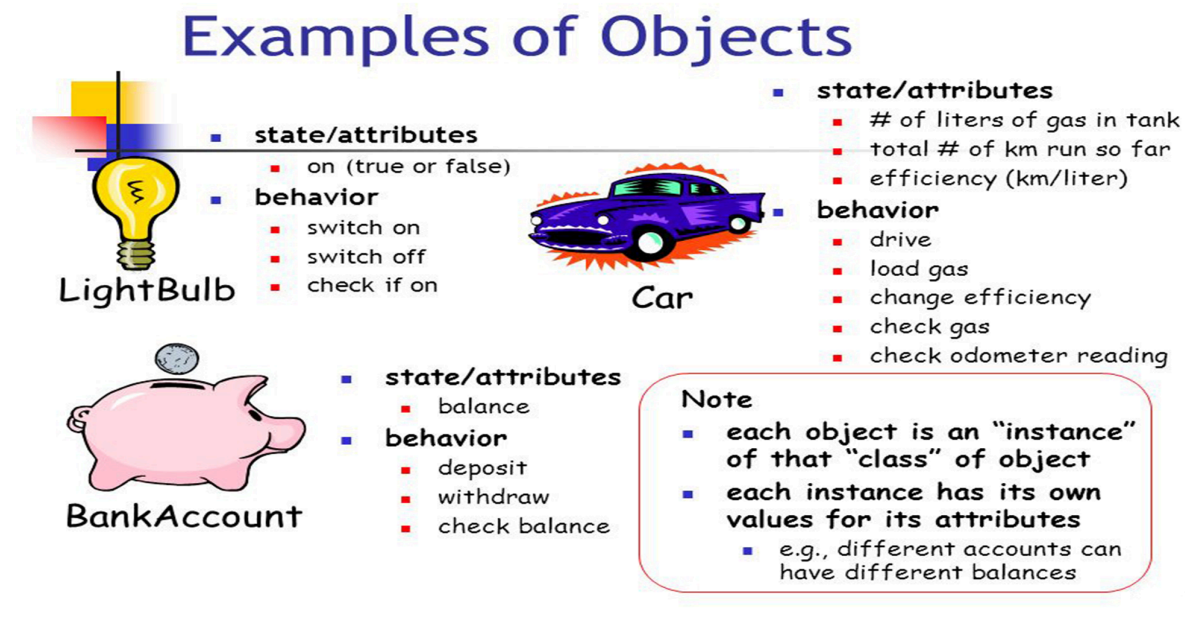


## DAY- 4

### Exercise:

1. Write a Java class structure for the following cases and print relevant SOP (printed messages).



### 1. LightBulb Class

```
class LightBulb {
    boolean isOn;

    // Constructor
    LightBulb() {
        this.isOn = false; // Default state is off
    }

    // Behavior methods
    void switchOn() {
        isOn = true;
        System.out.println("LightBulb is switched ON.");
    }

    void switchOff() {
```

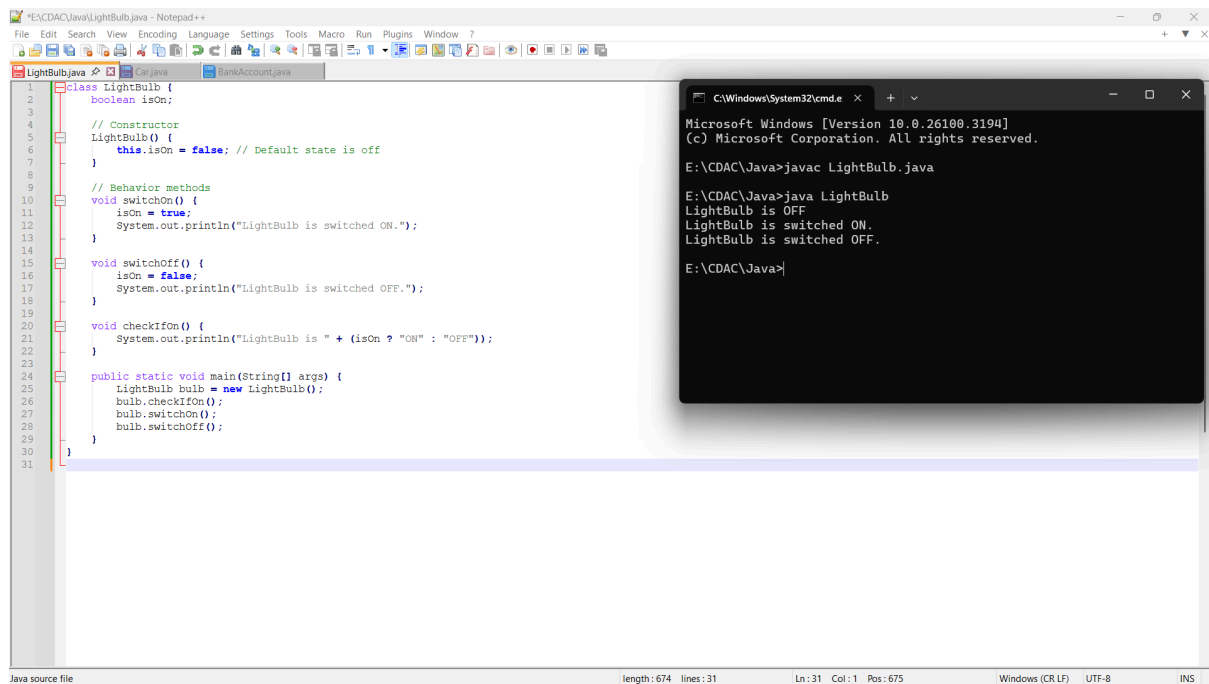
```
        isOn = false;
        System.out.println("LightBulb is switched OFF.");
    }

    void checkIfOn() {
        System.out.println("LightBulb is " + (isOn ? "ON" : "OFF"));
    }

    public static void main(String[] args) {
        LightBulb bulb = new LightBulb();
        bulb.checkIfOn();
        bulb.switchOn();
        bulb.switchOff();
    }
}
```

## **OUTPUT:**

```
E:\CDAC\Java>java LightBulb
LightBulb is OFF
LightBulb is switched ON.
LightBulb is switched OFF.
```



## 2. Car Class

```
class Car {  
    double fuelInLiters;  
    double totalKmRun;  
    double efficiency; // km per liter  
  
    // Constructor  
    Car(double fuelInLiters, double efficiency) {  
        this.fuelInLiters = fuelInLiters;  
        this.efficiency = efficiency;  
        this.totalKmRun = 0;  
    }  
  
    // Behavior methods  
    void drive(double distance) {  
        double fuelNeeded = distance / efficiency;  
        if (fuelInLiters >= fuelNeeded) {  
            totalKmRun += distance;  
            fuelInLiters -= fuelNeeded;  
            System.out.println("Car drove " + distance + " km.");  
        }  
    }  
}
```

```

        } else {
            System.out.println("Not enough fuel to drive.");
        }
    }

    void loadGas(double liters) {
        fuelInLiters += liters;
        System.out.println(liters + " liters of fuel added.");
    }

    void checkGas() {
        System.out.println("Remaining fuel: " + fuelInLiters + " liters.");
    }

    void checkOdometer() {
        System.out.println("Total kilometers run: " + totalKmRun);
    }

    public static void main(String[] args) {
        Car myCar = new Car(10, 15); // 10 liters of fuel, 15 km/l efficiency
        myCar.checkGas();
        myCar.drive(100);
        myCar.checkOdometer();
        myCar.loadGas(5);
        myCar.checkGas();
    }
}

```

## OUTPUT:

```

E:\CDAC\Java>java Car
Remaining fuel: 10.0 liters.
Car drove 100.0 km.
Total kilometers run: 100.0
5.0 liters of fuel added.
Remaining fuel: 8.333333333333332 liters.

```

The image shows a Notepad++ window with a Java file named Car.java. The code defines a Car class with attributes fuelInLiters, efficiency, and totalKmRun. It includes methods for driving, loading gas, checking gas, and checking the odometer. A main method creates a Car object and demonstrates its usage.

```
7 Car(double fuelInLiters, double efficiency) {
8     this.fuelInLiters = fuelInLiters;
9     this.efficiency = efficiency;
10    this.totalKmRun = 0;
11 }
12
13 // Behavior methods
14 void drive(double distance) {
15     double fuelNeeded = distance / efficiency;
16     if (fuelInLiters >= fuelNeeded) {
17         totalKmRun += distance;
18         fuelInLiters -= fuelNeeded;
19         System.out.println("Car drove " + distance + " km.");
20     } else {
21         System.out.println("Not enough fuel to drive.");
22     }
23 }
24
25 void loadGas(double liters) {
26     fuelInLiters += liters;
27     System.out.println(liters + " liters of fuel added.");
28 }
29
30 void checkGas() {
31     System.out.println("Remaining fuel: " + fuelInLiters + " liters.");
32 }
33
34 void checkOdometer() {
35     System.out.println("Total kilometers run: " + totalKmRun);
36 }
37
38 public static void main(String[] args) {
39     Car myCar = new Car(10, 15); // 10 liters of fuel, 15 km/l efficiency
40     myCar.checkGas();
41     myCar.drive(100);
42     myCar.checkOdometer();
43     myCar.loadGas(5);
44     myCar.checkGas();
45 }
46
47 }
```

The Windows Command Prompt shows the execution of the Java code:

```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

E:\CDAC\Java>javac Car.java

E:\CDAC\Java>java Car
Remaining fuel: 10.0 liters.
Car drove 100.0 km.
Total kilometers run: 100.0
5.0 liters of fuel added.
Remaining fuel: 8.333333333333332 liters.

E:\CDAC\Java>
```

Java source file length: 1,318 lines: 47 Ln: 47 Col: 1 Pos: 1,319 Windows (CR LF) UTF-8 INS

### 3. BankAccount Class

```
class BankAccount {
    double balance;

    // Constructor
    BankAccount(double initialBalance) {
        this.balance = initialBalance;
    }

    // Behavior methods
    void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited: " + amount + ". New balance: " + balance);
    }

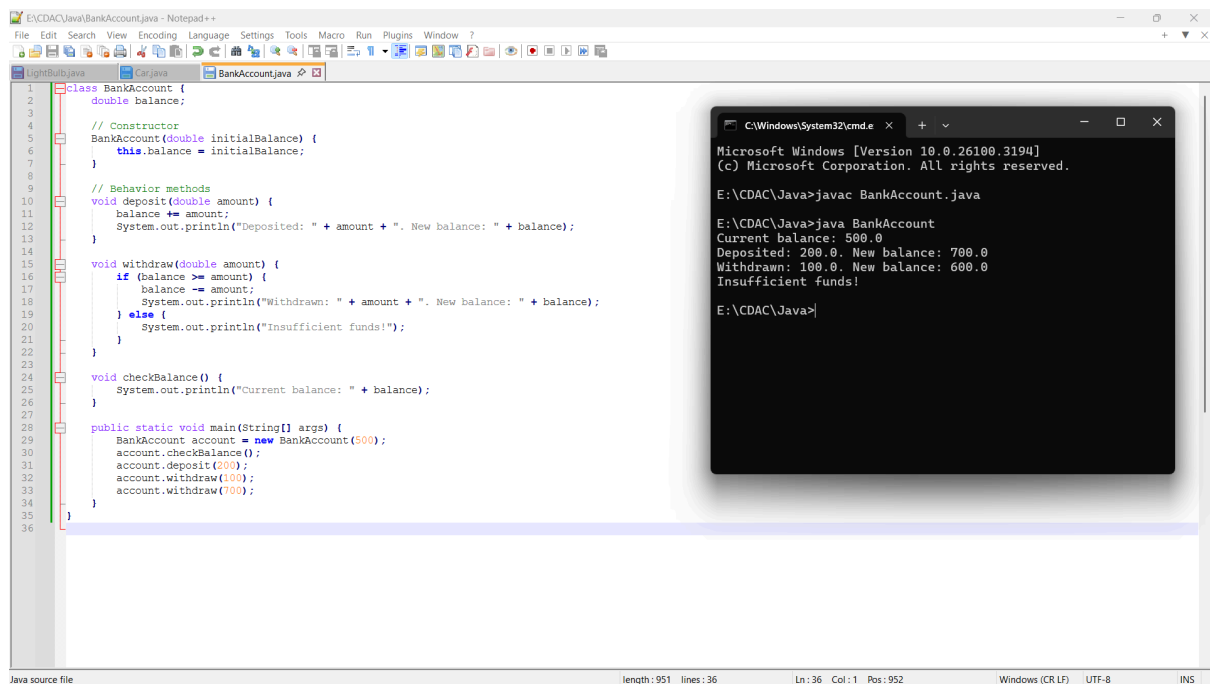
    void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount + ". New balance: " + balance);
        } else {
            System.out.println("Insufficient funds!");
        }
    }

    void checkBalance() {
        System.out.println("Current balance: " + balance);
    }

    public static void main(String[] args) {
        BankAccount account = new BankAccount(500);
        account.checkBalance();
        account.deposit(200);
        account.withdraw(100);
        account.withdraw(700);
    }
}
```

## OUTPUT:

```
E:\CDAC\Java>java BankAccount
Current balance: 500.0
Deposited: 200.0. New balance: 700.0
Withdrawn: 100.0. New balance: 600.0
Insufficient funds!
```



The screenshot displays a Java IDE with the source code for `BankAccount.java` on the left and a Windows command prompt window on the right showing the program's execution output.

```
1 class BankAccount {
2     double balance;
3
4     // Constructor
5     BankAccount(double initialBalance) {
6         this.balance = initialBalance;
7     }
8
9     // Behavior methods
10    void deposit(double amount) {
11        balance += amount;
12        System.out.println("Deposited: " + amount + ". New balance: " + balance);
13    }
14
15    void withdraw(double amount) {
16        if (balance >= amount) {
17            balance -= amount;
18            System.out.println("Withdrawn: " + amount + ". New balance: " + balance);
19        } else {
20            System.out.println("Insufficient funds!");
21        }
22    }
23
24    void checkBalance() {
25        System.out.println("Current balance: " + balance);
26    }
27
28    public static void main(String[] args) {
29        BankAccount account = new BankAccount(500);
30        account.checkBalance();
31        account.deposit(200);
32        account.withdraw(100);
33        account.withdraw(700);
34    }
35
36 }
```

The command prompt window shows the following output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

E:\CDAC\Java>javac BankAccount.java

E:\CDAC\Java>java BankAccount
Current balance: 500.0
Deposited: 200.0. New balance: 700.0
Withdrawn: 100.0. New balance: 600.0
Insufficient funds!

E:\CDAC\Java>
```