Program 5A.

```java
import java.util.Scanner;

//Bankinterface
interface BankInterface{
    double getBalance();
    double getInterestRate();
}

//bank A
class BankA implements BankInterface{
    private double balance;

    //constructor
    public BankA(double depositAmount){
        this.balance = depositAmount;
    }

    @Override
    public double getBalance(){
        return balance;
    }

    @Override
    public double getInterestRate() {
        return 0.07;
    }
}

//bank B
class BankB implements BankInterface{
    private double balance;

    //constuctor
    public BankB(double depositAmount){
        this.balance = depositAmount;
    }

    @Override
    public double getBalance() {
        return balance;
    }

    @Override
    public double getInterestRate(){
        return 0.074;
    }
}
```

```java
//bank C
class BankC implements BankInterface{
    private double balance;

    //constuctor
    public BankC(double depositAmount){
        this.balance = depositAmount;
    }

    @Override
    public double getBalance() {
        return balance;
    }

    @Override
    public double getInterestRate(){
        return 0.079;
    }
}

public class lab5a{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        //get deposit amount
        System.out.print("Enter deposit amount for Bank A: ");
        double depositA = scanner.nextDouble();
        BankA bankA = new BankA(depositA);

        System.out.print("Enter deposit amount for Bank B: ");
        double depositB = scanner.nextDouble();
        BankB bankB = new BankB(depositB);

        System.out.print("Enter deposit amount for Bank C: ");
        double depositC = scanner.nextDouble();
        BankC bankC = new BankC(depositC);

        //display the balance and interest rates for each bank
        System.out.println("\nBank A - Balance: Rs. " +bankA.getBalance()+",
Interest Rate: "+bankA.getInterestRate() * 100 + "%");
        System.out.println("\nBank B - Balance: Rs. " +bankB.getBalance()+",
Interest Rate: "+bankB.getInterestRate() * 100 + "%");
        System.out.println("\nBank C - Balance: Rs. " +bankC.getBalance()+",
Interest Rate: "+bankC.getInterestRate() * 100 + "%");

        scanner.close();
    }
}
```

Program 5B:

```java
import java.util.Scanner;
// WaterConservationSystem.java
interface WaterConservationSystem {
    int calculateTrappedWater(int[] blockHeights);
}

// RainySeasonConservation.java
abstract class RainySeasonConservation implements WaterConservationSystem {
    // Common methods or attributes can be added here
}

// CityBlockConservation.java
class CityBlockConservation extends RainySeasonConservation {

    @Override
    public int calculateTrappedWater(int[] blockHeights) {
        int n = blockHeights.length;
        if (n <= 2) {
            return 0; // Not enough blocks to trap water
        }

        int[] leftMaxHeight = new int[n];
        int[] rightMaxHeight = new int[n];

        // Precompute the maximum height to the left of each block
        leftMaxHeight[0] = blockHeights[0];
        for (int i = 1; i < n; i++) {
            leftMaxHeight[i] = Math.max(leftMaxHeight[i - 1],
blockHeights[i]);
        }

        // Precompute the maximum height to the right of each block
        rightMaxHeight[n - 1] = blockHeights[n - 1];
        for (int i = n - 2; i >= 0; i--) {
            rightMaxHeight[i] = Math.max(rightMaxHeight[i + 1],
blockHeights[i]);
        }

        int trappedWater = 0;
        for (int i = 0; i < n; i++) {
            int minHeight = Math.min(leftMaxHeight[i], rightMaxHeight[i]);
            trappedWater += Math.max(0, minHeight - blockHeights[i]);
        }

        return trappedWater;
    }
}
```

```java
// Main.java
public class lab5b {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        WaterConservationSystem conservationSystem = new
CityBlockConservation();
        // Take input from the user for block heights
        System.out.print("Enter the number of blocks: ");
        int n = scanner.nextInt();

        int[] blockHeights = new int[n];
        System.out.println("Enter the heights of each block:");
        for (int i = 0; i < n; i++) {
            System.out.print("Block " + (i + 1) + ": ");
            blockHeights[i] = scanner.nextInt();
        }

        // Calculate and display the trapped water
        int result = conservationSystem.calculateTrappedWater(blockHeights);
        System.out.println("Total trapped water: " + result + " units");

        // Close the scanner
        scanner.close();
    }
}
```