

Rajalakshmi Engineering College

Name: KAMALESH V. S

Email: 241801114@rajalakshmi.edu.in

Roll no: 241801114

Phone: 9361878560

Branch: REC

Department: AI & DS - Section 3

Batch: 2028

Degree: B.E - AI & DS

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 5_Q2

Attempt : 1

Total Mark : 10

Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

You are working as a developer for CityBank, which wants to build a basic account management system.

Each customer at the bank has:

An Account Number (integer)
A Customer Name (string)
An Initial Balance (double)

The bank allows two types of transactions:

Deposit – increases the balance.
Withdrawal – decreases the balance only if enough funds are available.

If the withdrawal amount is greater than the balance, the withdrawal should not happen, and the balance should remain the same.

You are required to implement this system using:

A class with attributes for account details. A constructor to initialize account details. Setter methods to update details if needed. Getter methods to retrieve details. Objects of the class to represent customers.

Finally, display each customer's account details after all transactions.

Input Format

The first line of input contains an integer N, representing the number of customers.

For each customer:

- The next line contains the account number (integer).
- The following line contains the customer name (string).
- The next line contains the initial balance (double).
- The next line contains the deposit amount (double).
- The next line contains the withdrawal amount (double).

Output Format

For each customer, print the details in the following format:

1. Account Number: <account_number>
2. Customer Name: <customer_name>
3. Final Balance: <final_balance> (rounded to one decimal place)

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1

1234

Rahul Sharma

5000

2000

3000

Output: Account Number: 1234

Customer Name: Rahul Sharma

Final Balance: 4000.0

Answer

```
// You are using Java
import java.util.Scanner;

class Customer {
    private int accountNumber;
    private String customerName;
    private double balance;

    // Constructor
    public Customer(int accountNumber, String customerName, double initialBalance) {
        this.accountNumber = accountNumber;
        this.customerName = customerName;
        this.balance = initialBalance;
    }

    // Setter methods (if needed)
    public void setAccountNumber(int accountNumber) {
        this.accountNumber = accountNumber;
    }

    public void setCustomerName(String customerName) {
        this.customerName = customerName;
    }

    public void setBalance(double balance) {
        this.balance = balance;
    }

    // Getter methods
    public int getAccountNumber() {
        return accountNumber;
    }

    public String getCustomerName() {
        return customerName;
    }

    public double getBalance() {
```

```
        return balance;
    }

    // Deposit method
    public void deposit(double amount) {
        if (amount >= 0) {
            balance += amount;
        }
    }

    // Withdraw method
    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
        }
        // If amount > balance, do nothing (as per requirement)
    }

    // Display customer info
    public void displayInfo() {
        System.out.printf("Account Number: %d%n", accountNumber);
        System.out.printf("Customer Name: %s%n", customerName);
        System.out.printf("Final Balance: %.1f%n", balance);
    }
}

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

        int N = Integer.parseInt(scanner.nextLine()); // Number of customers

        Customer[] customers = new Customer[N];

        for (int i = 0; i < N; i++) {
            int accountNumber = Integer.parseInt(scanner.nextLine());
            String customerName = scanner.nextLine();
            double initialBalance = Double.parseDouble(scanner.nextLine());
            double depositAmount = Double.parseDouble(scanner.nextLine());
            double withdrawalAmount = Double.parseDouble(scanner.nextLine());

            // Create customer object
        }
    }
}
```

```
Customer customer = new Customer(accountNumber, customerName,  
initialBalance);  
  
        // Perform transactions  
        customer.deposit(depositAmount);  
        customer.withdraw(withdrawalAmount);  
  
        // Store in array  
        customers[i] = customer;  
    }  
  
    // Display all customer details  
    for (Customer customer : customers) {  
        customer.displayInfo();  
    }  
  
    scanner.close();  
}  
}
```

Status : Correct

Marks : 10/10