7/1/24

Q Develop a java program to create a class Bank that maintains two kinds of accounts for it's customers, one called savings account and the other current account.

The Savings account provides compound interest and withdrawal facilities but no cheque book facility.

The current Account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if balance falls below this level, a service charge is imposed

Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements.

Include the necessary methods in order to achieve the following task:

(a) Accept deposit from customer and update the balance

(b) Display the balance

(c) Compute the deposit interest

(d) permit withdrawal and update the balance

Check for minimum balance, impose penalty if necessary and update the balance.

Class

```
import java.util.Scanner

Class Account {

    String CustomerName;
    String typeAccount;
    double balance;
    long accnum;

Account( string CustomerName, String typeAccount,
    double balance, long accnum)
    {
        this.CustomersName = CustomerName;
        this.typeAccount = typeAccount;
        this.balance = balance;
        this.accnum = accnum;
    }

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

            else
                {
                System.out.println("Invalid Amount");
                }
        }
```

```
public void display()
    {
     System.out.println("The balance is " + balance);
    }

Class Savacc extends Account {

        double intrestRate;

Savacc(String CustomerName, double balance, long
       accNum, double intrestRate)
        {  super(CustomerName, balance, accnum,
                 intrestRate);
           this.intrestRate = intrestRate;
        }

public CompountIntrest()
    {
     double intrest = balance * (intrestRate/100);
      deposite(intrest);

     System.out.println("Intrest Compound = " +
                        intrest);
    }

public withdraw(double amount)
    {
     if(amount > 0 && amount <= balance)
                {
                 balance -= amount;
                }
```

```
else
    {
    System.out.println("Invalid amount");
    }
}

class curAcc extends Account
    {
        final double MinBal = 4000;
        final double Service_charge = 100;

        public void withdraw (double amount)
            {
            if(amount > 0 && amount <= balance)
                {
                balance -= amount;
                System.out.println("withdrawn" +
                                    amount);
                checkMinBalance();
                }
            }

        public void checkMinBalance()
            {
            else
                {
                System.out.println("Invalid
                                    amount");
                }
            }
```

```
public void checkMinBalance(){
    if (balance < Min_Bal)
        {
        balance -= Service_charge;
        System.out.println("Minimum Balance is not
                            maintained");
        }

} }
```

```
public class Bank{

    public static void main(String[] args){

        SavAcc savings = new SavAcc("Aman", 12345791L,
                                    5500, 9.4);

        savings.display();
        savings.deposit(1000);
        savings.display();
        savings.withdraw(500);
        savings.display()
        savings.compoundInterest();
        savings.display();

        System.out.println("Current account operations
```

```
CurAcc current = new CurAcc ("sanchit",
                 1234567899L, 2000 );
current.display();
current.deposit(500);
current.display();
current.withdraw(1200);
current.display();
}
}
```

Output :→ The balance is: 5500
Deposited : 1000
The balance is : 6500
Withdrew : 500
The balance is: 6000
Deposited : 564
Interest Compounded : 564
The balance is : 6564

Current account operations

The balance is : 2000
Deposited : 500
The balance is : 2500
Withdrew : 1200
The balance is : 1300

28/11/24
LS

```java
import java.util.Scanner;

class Account {
    String customerName;
    String accountNum;
    double balance;

    public Account(String customerName, String accountNum, double balance) {
        this.customerName = customerName;
        this.accountNum = accountNum;
        this.balance = balance;
    }

    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: " + amount);
        } else {
            System.out.println("Invalid amount");
        }
    }

    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrew: " + amount);
        } else {
            System.out.println("Invalid request or insufficient funds");
        }
    }

    public void display() {
        System.out.println("The balance is: " + balance);
    }
}

class SavAcc extends Account {
    double interestRate;

    public SavAcc(String customerName, String accountNum, double balance, double interestRate) {
        super(customerName, accountNum, balance);
        this.interestRate = interestRate;
    }
```

```
    public void compoundInterest() {
        double interest = balance * (interestRate / 100);
        deposit(interest);
        System.out.println("Interest compounded: " + interest);
    }
}

class CurAcc extends Account {
    static final double MIN_BAL = 1000;
    static final double SERVICE_CHARGE = 100;

    public CurAcc(String customerName, String accountNum, double balance) {
        super(customerName, accountNum, balance);
    }


    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrew: " + amount);
            checkMinBalance();
        } else {
            System.out.println("Invalid request or insufficient balance");
        }
    }

    public void checkMinBalance() {
        if (balance < MIN_BAL) {
            balance -= SERVICE_CHARGE;
            System.out.println("Minimum balance not maintained. Service charge imposed: " + SERVICE_CHARGE);
        }
    }
}

public class lab5 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter customer name for savings account: ");
        String savName = scanner.nextLine();
        System.out.print("Enter account number for savings account: ");
        String savAccNum = scanner.nextLine();
        System.out.print("Enter initial balance for savings account: ");
        double savBalance = scanner.nextDouble();
        System.out.print("Enter interest rate for savings account: ");
        double savInterestRate = scanner.nextDouble();

        scanner.nextLine();
```

```
        SavAcc savings = new SavAcc(savName, savAccNum, savBalance, savInterestRate);
        savings.display();

        System.out.print("Enter the amount to deposit in savings account: ");
        savings.deposit(scanner.nextDouble());
        savings.display();

        System.out.print("Enter the amount to withdraw from savings account: ");
        savings.withdraw(scanner.nextDouble());
        savings.display();

        savings.compoundInterest();
        savings.display();

        System.out.print("\nEnter customer name for current account: ");
        String currName = scanner.nextLine();
        System.out.print("Enter account number for current account: ");
        String currAccNum = scanner.nextLine();
        System.out.print("Enter initial balance for current account: ");
        double currBalance = scanner.nextDouble();

        scanner.nextLine();

        CurAcc current = new CurAcc(currName, currAccNum, currBalance);
        current.display();

        System.out.print("\nEnter the amount to deposit in current account: ");
        current.deposit(scanner.nextDouble());
        current.display();

        System.out.print("Enter the amount to withdraw from current account: ");
        current.withdraw(scanner.nextDouble());
        current.display();

        scanner.close();
    }
}
```

```
C:\1bm23cs299>java lab5
Enter customer name for savings account: sanchit
Enter account number for savings account: 111
Enter initial balance for savings account:
23
Enter interest rate for savings account: 12
The balance is: 23.0
Enter the amount to deposit in savings account: 222
Deposited: 222.0
The balance is: 245.0
Enter the amount to withdraw from savings account: 21
Withdrew: 21.0
The balance is: 224.0
Deposited: 26.88
Interest compounded: 26.88
The balance is: 250.88

Enter customer name for current account: Enter account number for current account: xyz
Enter initial balance for current account: 1111
The balance is: 1111.0

Enter the amount to deposit in current account: 125678
Deposited: 125678.0
The balance is: 126789.0
Enter the amount to withdraw from current account: 2223
Withdrew: 2223.0
The balance is: 124566.0
```