LAB program 5

Develop a Java program to create a class Bank that maintains two kinds of account for its 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 the 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 tasks:
a) Accept deposit from customer and update the balance.
b) Display the balance.
c) Compute and deposit interest
d) Permit withdrawal and update the balance
Check for the minimum balance, impose penalty if necessary and update the balance.

CODE:

```
import java.util.Scanner;

class Account {
    protected String customerName;
    protected int accountNumber;
    protected String accountType;
    protected double balance;

    public Account(String customerName, int accountNumber,
```

```
    String accountType, double balance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = balance;
    }

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

    public void displayBalance() {
        System.out.println("Account Balance: " + balance);
    }
}

class SavAcct extends Account {
    private static final double INTEREST_RATE = 0.07;

    public SavAcct(String customerName, int accountNumber, doubl
        super(customerName, accountNumber, "Savings", balance);
    }

    public void computeAndDepositInterest() {
        double interest = balance * INTEREST_RATE;
        balance += interest;
        System.out.println("Interest of " + interest +
        " has been added. New balance: " + balance);
    }

    public void withdraw(double amount) {
```

```
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal successful. New balan
        } else {
            System.out.println("Invalid withdrawal amount or ins
        }
    }
}

class CurAcct extends Account {
    private static final double MIN_BALANCE = 500.0;
    private static final double SERVICE_CHARGE = 50.0;

    public CurAcct(String customerName, int accountNumber, doubl
        super(customerName, accountNumber, "Current", balance);
    }

    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal successful. New balan
            checkMinimumBalance();
        } else {
            System.out.println("Invalid withdrawal amount or ins
        }
    }

    private void checkMinimumBalance() {
        if (balance < MIN_BALANCE) {
            balance -= SERVICE_CHARGE;
            System.out.println("Balance fell below minimum. Serv
            + SERVICE_CHARGE + " applied. New balance: " + balan
        }
    }
}
```

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

        System.out.print("Enter customer name: ");
        String customerName = scanner.nextLine();

        System.out.print("Enter account number: ");
        int accountNumber = scanner.nextInt();

        System.out.print("Enter initial balance: ");
        double initialBalance = scanner.nextDouble();

        System.out.println("Choose account type:");
        System.out.println("1. Savings Account");
        System.out.println("2. Current Account");
        int accountChoice = scanner.nextInt();

        Account account;
        if (accountChoice == 1) {
            account = new SavAcct(customerName, accountNumber, i
        } else if (accountChoice == 2) {
            account = new CurAcct(customerName, accountNumber, i
        } else {
            System.out.println("Invalid account type selection."
            scanner.close();
            return;
        }

        boolean exit = false;
        while (!exit) {
            System.out.println("\nChoose an operation:");
            System.out.println("1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Display Balance");
            if (account instanceof SavAcct) {
```

```
                System.out.println("4. Compute and Deposit Inter
            }
            System.out.println("5. Exit");
            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter amount to deposit: "
                    double depositAmount = scanner.nextDouble();
                    account.deposit(depositAmount);
                    break;

                case 2:
                    System.out.print("Enter amount to withdraw:
                    double withdrawAmount = scanner.nextDouble()
                    if (account instanceof SavAcct) {
                        ((SavAcct) account).withdraw(withdrawAmo
                    } else if (account instanceof CurAcct) {
                        ((CurAcct) account).withdraw(withdrawAmo
                    }
                    break;

                case 3:
                    account.displayBalance();
                    break;

                case 4:
                    if (account instanceof SavAcct) {
                        ((SavAcct) account).computeAndDepositInt
                    } else {
                        System.out.println("Invalid choice for C
                    }
                    break;

                case 5:
                    exit = true;
```

```
                    System.out.println("Exiting...");
                    break;

                default:
                    System.out.println("Invalid choice.");
            }
        }

        scanner.close();
    }
}
```

OUTPUT:

```
Enter customer name:xyz
Enter account number: 1001
Enter initial balance: 1000
Choose account type:
1. Savings Account
2. Current Account
1

Choose an operation:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5. Exit
1
Enter amount to deposit: 200
Deposit successful. New balance: 1200.0

Choose an operation:
1. Deposit
2. Withdraw
3. Display Balance
```

```
4. Compute and Deposit Interest
5. Exit
4
Interest of 48.0 has been added. New balance: 1248.0

Choose an operation:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5. Exit
5
Exiting...
```

OBSERVATION:

Lab Program - 5

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called Saving account and other current account.

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

Current account provides cheque book facility but no interest.

Current account holders should also maintain a minimum balance and if the 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 tasks.

a) Accept deposite from customer and update the balance

b) Display the balance

c) Compute and deposit interest

d) Permit withdrawal and update the balance. Check for the minimum balance impose penalty if necessary and update balance. Java

```
import java.util.Scanner;
class Account {
    String customername;
    int accountNumber;
    String accountType;
    double balance;

    Public Account (String customername,
                    int accountNumber, String
                    accountType, double balance) {

    this.customername = customername;
    this.accountNumber = accountNumber
    this.accountType = accountType;
    this.balance = balance;
}

  Public void deposit (double amount) {
    if (amount >0) {
        balance += amount;
        Sout ("Deposit successfull");
    }
    else {
        Sout ("Invalid deposit amount");
    }
}
```

```
    public void displayBalance() {
        Sout("Account Balance:" + balance);
    }

class SavAcct extends Account {
    private static double interestRate 0.04;

    public SavAcct(String CustName, acctNum, double
                   balance) {
        Super(CustomerName, accountNumber, "Saving"
                                   balance);
    }

    Public void depositInterest() {
        double interest = balance * InterestRate
        balance += interest;
        Sout("Interest of" + interest + has been added)
    }

Public void withdraw(double amount) {
    if(amout > 0 && amount <= balance) {
        balance -= amount;
        System.out.println("Withdrawal
          Successful");
    }
    else {
        Sout("Invalid withdraval");
    }
```

```
class CurAcct extends Account {
    private static double Min_balance = 500.0;
    private static double Service_charge = 50.0;

    public CurAcct (String customerName, int
            accountNumber, double balance) {
        super (customerName, accountNumber,
            "current", balance);
    }

    public void withdraw (double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            Sout ("Withdrawal successful");
            checkMinimumBalance();
        } else {
            Sout ("Invalid withdrawal");
        }
    }

    private void checkMinimumBalance () {
        if (balance < MIN_BALANCE) {
            balance -= SERVICE_CHARGE;
            Sout ("Balance fell below min")
        }
    }
}

Public class Bank {
    PSVM () {
    Scanner sc = new Scanner (System.in);
    Account account = null;
    Sout ("Select Bank Type");
    Sout ("1. SB");
    Sout ("2. current Account");
```

```
int accountchoice = sc.nextInt();
Sout ("Enter your name: ");
String customername = sc.nextLine();
sout ("enter your account number: ");
int accountnumber = sc.nextInt();
sout ("enter initial deposit");
double balance = sc.nextDouble();

switch (accountchoice) {
    case 1:
        account = new SavAcct(customername, accountnu
                -balance);
        Sout ("Saving account created");
        break;

    case 2:
        account = new CurAcct(CustOmername, accountnu
                balance);
        Sout ("current Account created");
        break;
    default:
        Sout ("Invalid choice");
        return;
}

While (true) {
    Sout ("Choose an operation:");
    Sout ("1. Deposit");
    Sout ("2. Display Balance");
    Sout ("3. Compute and Deposit Interest
                (Savings Account only)");
    Sout ("4. withdraw");
    Sout ("5. Exit");
```

```
            case 5:
                System.out.println("Thank You");
                sc.close();
                return;
            default:
                System.out.println("Invalid choice");
        }
    }
}
```

output

```
Select Type of account:
1. Savings account
2. Current account
1
Enter your name: Shrinanda
Enter your acct number: 123
Enter initial deposit: 2220
Savings Account created:

Choose an operation
1. Deposit
2. Display balance
3. Compute and Deposit Interest
4. Withdrawal
5. Exit
1
Enter deposit amount: 70006
Deposit Successful.
```

Choose an operation:

1. Deposit
2. Display balance
3. Compute and deposit interest
4. Withdrawal
5. exit

2

Account Balance : ₹2220.

07/11/24