Q]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: " + 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, double balance) {
        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) {</pre>
            balance -= amount;
            System.out.println("Withdrawal successful. New balance: $" + balance
        } else {
            System.out.println("Invalid withdrawal amount or insufficient balance
        }
    }
}
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, double balance) {
        super(customerName, accountNumber, "Current", balance);
    }
    public void withdraw(double amount) {
```

```
if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal successful. New balance: " + balance)
            checkMinimumBalance();
       } else {
            System.out.println("Invalid withdrawal amount or insufficient balance
       }
    }
    private void checkMinimumBalance() {
        if (balance < MIN_BALANCE) {</pre>
            balance -= SERVICE_CHARGE;
            System.out.println("Balance fell below minimum. Service charge of "
            + SERVICE_CHARGE + " applied. New balance: " + balance);
       }
   }
}
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, initialBalance);
       } else if (accountChoice == 2) {
            account = new CurAcct(customerName, accountNumber, initialBalance);
            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 Interest ");
    }
    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(withdrawAmount);
            } else if (account instanceof CurAcct) {
                ((CurAcct) account).withdraw(withdrawAmount);
            break;
        case 3:
            account.displayBalance();
            break;
        case 4:
            if (account instanceof SavAcct) {
                ((SavAcct) account).computeAndDepositInterest();
                System.out.println("Invalid choice for Current Account."
            }
            break;
        case 5:
            exit = true;
```

## **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
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
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 Interest5. Exit5Exiting...
```

## **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 chique book facility but no interest.

Reverent account holders should also maintain a minimum kalance and if the balance falls below this level, a service charge is imposed

rame, account number and type of account.

From this derive the classes Lux-act and Sav-act to make them of more specific to their requirements. Include the necessary methods in order to achieve the following tasks.

- a) Accept deposite from customer and in update the balance
- 3) Display the balance
- c) Compute and deposit interest

  D resmit withdrawal and supdate the balance.

  Theck for the minimum balance impose penalty,

  if necessary and supdate balance. java

```
împort java. util. scanner;
     class account &
          Prt account Number;
       Strêng account Type;
chouble balance;
          Public Account (storing customername,
       ent account Number, Storney
                     accountType, double balance) 2
          this customername = customername;
  this. account Number = account Number this. raccount type = account type is
           this. balance = palance ;
Sout l'Isternat plus plus addet
        Public void deposit (Cdouble amount) ?
   3 (constage amount 70) (constation bior sides
                balance + = amount;
   doeurs bette sout l'Opposit successful "D's
else l
          else & sout ("Invalid deposit amount");
        Bout ( " Invalid withdraval " ?;
```

```
Public void display Balance () (
  Sout [" Account Balance: " + bulance);
class Savacet extends Account 1
     prévate statée double interestrate 0.04 :
     Public Sar Acct Estring Cust Name, acct Num, double
                 balane ) (truesde oldug
       Super Constoner Name, account Number, "Saving
 & Campbid should egyithmas balaner;
 Public Void deposit Intrest () 1
      clouble onterest = balance & InterestRate
    balance + = Interest :
    Sout ("Interest of" + interest + has been added)
roid deposit Eduuble ameunt) for
public void with draw (double amount) &
     if Camout >0 && vannount <= balance) {
balance - = amount;
       System. out. println ( " withdrawal
       Successful ");
  sout (" Trivalid diposit amount")
Sout (" Invalid withdraval");
expert 2 and small ...
```

```
class current extends scount of
                 Static double Min - balance = 500.0;
         private static double service-charge = 50-0;
        public aux Acct (String sustamerName, int
               account Number, double balance) (
          super Coustomerhame, account Number,
               "current", balance);
                           applied (decountribeico)
         2
 public void withdraw (double amount) 3
          if Camount xo so amount == balance) ?
              balance - = amount;
               sout (" With drawal successful"):
              reheck Mininum Balance ();
         4 else 1
             sout ("Invalid withdrawal"):
        private void checkplinhumBalance 1)?
           if Chalance < MIN_BALANCE) {
               balana -= SERVICE - CHARGES
               sout ("Balance fell below min")
          3
7
       class Bank (
Public
       ps vm () x
       Scanner Sc= new Scanner ( System in)
Account account = null;
        Sout (" Select Bank Type")
        Sout ("1.5B");
        Sout (" 2. Cuvert Account")
```

```
int account choice = screet Int();
        Sout ("Enter your name: ");
        String austonicnami = 50. next Line ();
       sout c"enter your account rumber: ");
        int account number = sc. restInt();
        sort (" Enter initial deposit")
        double balance = sc. nextDouble();
        swith (account choice) {
      case 1:
              account = new SavAcet( customuranze, accounting
                 balana);
              Sout (" Saving account executed"),
             borak:
         case 2:
          account = new Cur Aect Caust Omernance,
                       balance);
            Sout (" current account created ");
  break;
   A default:
suturn;
 while (true) $
     sout (" chowe an operation: ");
     Sout ("1. Deposit");
    sout (2. Display Bolana's)
  Sout ("3. Conjute and Deposito Interest
(Savings Account only)");
Sout ("4", Withdraw");
Sout (5 Fxit"),
  Sout 65. Exit ");
```

4

```
case 5:
          Sout ("Thank You");
          sc. close's
          returns
      default !
       Sout ( " Invalid chorce")
       4
   output
   select Type of account:
    1. Sawings account
    2. surrent account
   1
    Enter your name: Shrinanda
   Enter your act number: 123
   Enter initial deposit: 2220
    Savengs Account related!
  choose are operation
  1. Deposit
  2. Display balance
3. Compute and Depart Intuisi bus
        if Cascomit motomes of sandoccos i
(15 ay book) account) withdraw the transaction
Entir deposit amous: 70006
Deposit Successful Miles (And alesso (to-Arus ))
```

choose an operation.

- 1. Deposit
- 2. Display balance
- 3. Compute and deposits interes
- 4. Nitroblawal
- 5. Enu'f

Account Balance: 72220.

07/11/24