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, doub.
        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 balar
        } 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) {</pre>
            balance -= amount;
            System.out.println("Withdrawal successful. New balar
            checkMinimumBalance();
        } else {
            System.out.println("Invalid withdrawal amount or in:
        }
    }
    private void checkMinimumBalance() {
        if (balance < MIN_BALANCE) {</pre>
            balance -= SERVICE_CHARGE;
            System.out.println("Balance fell below minimum. Serv
            + SERVICE_CHARGE + " applied. New balance: " + balar
        }
    }
}
```

```
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, :
        } else if (accountChoice == 2) {
            account = new CurAcct(customerName, accountNumber,
        } 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).computeAndDepositIn
        } else {
            System.out.println("Invalid choice for (
        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
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 chique book facility but no interest.

revient 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 Sar-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 update the balance
- 3) Display the balance
- c) Compute and deposit interest
- D Permit withdrawal and repetate the balance. heck for the minimum balance impose penalty, if necessary and repetate balance. java

```
import java. util. scanner;
    class account &
         String automerane;
         Put account Number;
                             charge transfect water
     String account Type;
          double balance:
   public sanked latering instrument act Num, double
         Public Account (strong customername,
     reduced the sont raccount Number, Storney
              accountType, double balance) 2
          this customername = customername;
  this account Number = account Number
          this. raccount type = account type;
          this. balance = palance it amaled
Sout C'Interest of "+ "sternet pings been added?
       Public void deposit (double amount) ?
   3 (unologie = ) in roma
               balance + = amount;
   Sout ("Déposit successful "),
                      Elisabeth "
         else & sout ("Invalid deposit amount");
        Bout ( or Invalid withdrauat " ?;
```

```
Public void display Balance () (
 Sout (" Account Balance: " + bulance);
 4
class Savacet extends Account &
     prévate statéc double interestrate 0.04;
     public San Acct Cotting Cust Name, acct Num, double
                 Chalano) (musica olding
      Super Constomer Name, account Number, "Saving
 & lumbled should a typ? In and balanew;
Public Void deposit Interest () {
      clouble enterest = balance & InterestRate
    balance + = Enterest :
    Sout ("Interest of" + interest + has been added)
Void deposit Educate amounts for
public void with draw Cdouble amount) &
     if Camout >0 & varrount <= balance) [
balance - = amount;
       System. out. println ( " withdrawal
       Successful ");
  sout (" Invalid dipolit amount")
Sout (" Invalid withdraval");
specific and descript a second
net will revold
```

```
class curect extends secount of
         private static double Min - balance = 500.0;
         private static double service-charge = 50-0;
        public curact (String customerName, int
               account Number, double balance) (
           Super Constanuerams, account Dunker,
                " current", balance);
  public void withdraw (double amount) ;
           if camount so so amount == balance) {
               balance - = amount;
               sout (" With drawal successful"):
              reheck Mininum Balance ();
          4 else 1 «Invalid withdrawal");
        private void checkalinimum Balance 1)?
           if Chalance < MIN_BALANCE) {
                balance -= SERVICE_CHARGES
               sout ("Balance jell below min")
7
Public
       ps vw () x
Scanner Sc= new Scanner ( Syptement);
Account account = null;
        Sout (" Select Bank Type");
        Sout ("1.5 B");
      Sout (" 2. Cuvert Account")
```

```
int account choice = screet Int();
       Sout ("Enter your name: ");
        Stoing austonikname = 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 SarAcet( customuname, accounting
                  balana);
              Sout ( " Saving account executed "),
             borak:
         case 2:
           account = new Cur Aect Caust Omernance,
                        balance);
            Sout (" current account created ");
  break;
   adefault:
sutturn;
 while (true) &
     sout (" chows an operation: ");
     Sout ("1. Deposit");
    sout (2. Display Bolana's)
  Sout ("3. Compute and Deposito Interest
(Savings Account only)");
Sout ("4", Withdraw");
Sout (5 Fxit").
  Sout 65. Exit. o).
```

4

```
case 5:
           Sout (Thank You")
           sc. close's
           returns
      default:
       Sout (" Invalid chorce")
       4
   g
   output
 Select Type of account:

1. Sawings account
    2. Auvent account
    1
    Enter your name: Shuinanda
    Enter your act number: 123
   Enter initial deposit: 2220
    Savorgs Account related:
  choose are operation
  1. Deposit
  2. Display balance
() 3 and compute and pepart Interest was
            Cauching Protonce of Sandee
(15ay book) account) withdraw (hisparical &
Enter deposit amous: 70006
Deposition Successful Miles (Am acons (to Amis))
```

choose an operation.

- 1. Deposi+
- 2. Display balance
- 3. Compute and deposits interest
- 4. Nitroblawal
- 5. Enu'f

2 Account Balance: 72220.

07/11/24