9-1-24 Lab Prgm - 05.

Develop a Java prgm to create a class Bank that maintains two kinds of accounts for its customers, one called savings account and 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 the minimum balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;
class Account{
    String customerName;
    String accountNumber;
    String accountType;
    double balance;
    Account(String customerName, String accountNumber,
     String accountType, double balance)
    {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = balance;
    }
  void deposit(double amount)
  {
      balance += amount;
      System.out.println("Deposite successful,
          Updated balance:" + balance);
  }
  void displayBalance()
  {
      System.Out.println("Account Type:" + accountType);
      System.out.println("Customer Name:" + customerName);
      System.out.println("Account Number:" + accountNumber);
      System.out.println("Balance" + balance);
  }
}
void withdrawal()
{
    System.out.println("Enter withdrawal amount");
    int with.sc.nextInt();
```

```
if (with > balance)
{
    System.out.println("Insufficint balance");
}
else
{
    balance.withd;
}
}
void applyInterest()
{
    package java.lab;
    public class Cur-act extends Accocint
    {
        Cur-acct(string CustomerName, int accountNum
        string accocunt type)
        {
            super(Customer Name, Account Number,
                accounttype);
        }
void withdrawal()
{
    System.out.println("Enter withdrawal amt");
    int withd = sc.nextInt();
    if (balance <= 2000)
    {
        double pen.balance/(0.06);
        System.out.println("Insufficinct balance
        penalty to be paid
        balance + pen;
    else
    { balance.withd; } } }
```

```
package java.lab;
public class Sav_account extends Account;
{
    Sav_acct (String CustomerName, accountNum,
        accountType).
    {
        super(CustomerName - AccountNumber,
            accountType);
    }
void applyInterest()
{
    System.out.println("Enter interest rate").
    int rate = sc.nextInt();
    double interest = balance * (rate/100);
    balance + interest;
    SOP("Balance after interest :" + balance)
}
}
package java.lab;
import.java.util Scanner;
public class Book
{
    public static void main(String args[])
    {
        Scanner sc.new Scanner(System.in).
        SOP("Enter customer name").
        String CustomerName.sc.nextLine();
        SOP("Enter account number");
        int acctNum = sc.nextLine();
        account ca.new
        Cur_acct (CustomerName, AccountNum,
            current amount);
```

```
Account sc = new
sav_acct(customer Name, account number, "Savingacct");
int choice;
while (true)
{
   SOP(1. Deposit In 2. Withdrawal In 3. Compute Interest
       In 4. Display);
   choice = sc.nextInt();
   switch (choice)
   {
      case 1: SOP("Enter the type of account In
         1. Saving In 2. Current In");
      int acc = sc.nextInt();
      if (acc == 1)
      {
         sa.deposit();
      }
      else
      {
         ca.deposit();
      }
      break;
   case 2:
      SOP("Enter type of account");
      int acc = sc.nextInt();
      if (acc == 1)
      {
         sa.withdrawal();
      }
      else
      {  ca.withdrawal();
      }
      break;
```

```
case 3:
    sa.applyInterest();
    break;
case 4:
    sop("1.saving In 2.current");
    int acc2 = sc.nextInt();
    if(acc2 == 1)
    {
        sa.display();
    }
    else
    {
        ca.display();
    }
    break;
}
}
}
```

O/P:

```
Enter customer Name: Shreya
Enter Acc no : 25485
-Menu-
1. Deposit
2. Withdrawal
3. Compute Interest
4. Display
1
Enter 1.saving 2.current
1
Enter amount 8000
Enter your choice 2
Enter 1. savings 2. Current
1
Enter amount : 2000
Balance amot : 5000
-Menu-
1. Deposit
2. Withdrawal
3. Compute interest
4. Display
4
1. saving 2 Current
1
Customer Name : Shreya
ACC NO : 25485
Balance : 8000
```