LAB 4

1. Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings, other current. The savings acc provides compound interest and withdrawal facilities but no cheque book facility. The current acc. provides cheque book facility but no interest. Current acc. holders should also maintain a min. balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account no. and type of account. From this derive 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 min. balance, impose penalty if necessary and update the balance.

```
import java.util.*
class Bank{
class Account{
    String customer_name;
    int accno;
    double bal;

    Account(customer_name, accno, bal)
    {
      this.customer_name = customer_name;
      this.accno = accno;
      this.bal = bal;
    }

    void accept(double deposit)
    {
        deposit += bal;
        System.out.println("Deposit Accepted
current balance is: " + bal);
    }

    void displayBal()
    {
      System.out.println("Balance is: " + bal);
    }
}

class curAcct extends Account
{
    double min;
    boolean chequebook;
    float service_charge;
    float fee;
```

```
static
void minbal()
{
    if(bal < minbal)
    {
        System.out.println("Balance is below
min. balance penalty fee");
        bal -= servicecharge;
        System.out.println("Current balance" +bal);
    }
}
```

```
Current(double min, boolean chequebook, float service_charge)
{
    super(
```

```
currAct()
{
    min = 1500.00;
    chequeBook = true;
    Service_charge = 50.0;
}
```

```
static void minbal()
{
    if(bal < min)
    {
        System.out.println("Balance is below min
balance penalty");
        bal -= service_charge;
        System.out.println("Balance : " + bal);
    }
}
```

```
void withdrawl(double amt)
{   this.minbal();
    bal -= amt;
    System.out.println("Withdrawl success,
    Remaining Balance : " + bal);
}
}

class Savacct extends Account
{
    double interst;

    Savacct(double interest);
    {
    this.interest = interest;
    }
    void Interest()
    {
      System.out.println("Compound Interest:");
      interest = interest * bal;
      System.out.println("Ba
    }

    void ChequeBook()
    {
     System.out.println("No. chequeBook facility");
    }

    void withdrawl(double amt)
    {
        bal -= amt;
        System.out.println("Remaining Balance:"+bal);
    }
```

```
void CompoundInt(double interest, int yr)
{
  double int = bal * Math.pow((1+interest), yr)
    - bal;
    bal += int;
    System.out.println("Compound Interest in
Balance : "+bal);
    }
}

class Bank{
  {
    Public static void main (String args[])
    {
  Scanner input = new Scanner(System.in);
  System.out.println("Enter ~~Customer Acc~~ Details")
  System.out.println("Enter name of customer");
  String n = input.nextline();
  int acc_num = input.nextInt()
  System.out.println("Enter accno");
  int acc_num = input.nextInt();
  System.out.println("Enter balance");
float ~~bal~~ b= input.next Float();

    Account obj = new Account("~~Saving~~", n, accno
b ~~215~~, 15000.00);

    obj.Accept(20000.00);
    obj.displayBal();
```

```
import java.util.*;
    CurrAct obj1 = new CurrAct();
    obj1.withdrawl(2000);
    Savacct obj2 = new Savacct(7.5);
    obj2.chequeBook();
    obj2.withdrawl(5000);
    obj2.compoundInt(7.5, 2);
  }
}
```

OUTPUT.

Enter Details
Enter name: xyz
Enter accno: 213
Enter Balance: 120000
Deposit Accepted current Balance is: 32,000
Balance is: 32,000
withdrawl success Remaining balance: 30,000
No chequebook facility
Remaining balance is: 25,000.
Compound Interest
Balance: 28390.6

Algorithm:

Step 1: Start
Step 2: Create a class Account
create a parameterized constructor
Accept and initialize variables
customer_name, accno, bal
Step 3: Take inputs for variables n, acc_num
b
Step 4: call Account constructor
Step 5: call Accept() in Account class
Step 6: call displayBal() in account class

Step 7: create a class CurrAct extending Account
Step 8: call CurrAct() constructor.
Step 9: call withdrawal() in CurrAct class
Step 10: create a class SaveAct extending Account
Step 11: call SaveAct() constructor
Step 12: call chequebook() for SaveAct class
Step 13: call withdrawal() for SaveAct class
Step 14: call CompoundInt() for SaveAct class
Step 15: Stop.

A 19/01/24

```
Enter Details
Enter Name
xyz
Enter Accno
213
Enter Balance
12000
Deposit success Balance is:32000.0
Balance is:32000.0
Withdrawal success remaining Balance:10000.0
NO FACILITY
Remaining Balance is:7000.0
Compound interest
 New Bal8089.374999999999
NAME:ADIKAR CHARVI SREE TEJA USN:1BM22CS012
```