Date __/__/__
Page

Lab program - 5

Q] Develop a Java Program to create a class Bank that maintains two kinds of accounts for its customers, one is savings, other one is current account, the savings account provides compound interest & withdrawal facilities but no checkbook facility, while current account provide checkbook facility but no interest. Current acco holders should 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, Acc no and type of Account. From this derive the classes Curr-acct and Sav-acct to make the more specific to their requirements. Include the necessary methods in order to achieve the follo tasks.

a) Accept deposit from customer & update the balance

b) Display the balance

c) Compute & deposit interest

d) Permit withdrawal & update the balan

Check for minimum balance, impose penalty if necessary & update the balance

Date __/__/__
Page _____

A:

```
import java.util.Scanner;

class Account {
protected String customer_name;
protected int Acc_no;
protected double balance;

public Account (String customerName, int accountNumber, double balance){
    Customer_name = customerName;
    Acc_no = accountNumber;
    this.balance = balance;
}

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

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

class SavAcct extends Account {
    final double interest = 0.05;

public SavAcct (String customer_Name, int accountnumber, double balance)
    {
        super (customerName, accountnumber, balance);
```

Date / /
Page

```
public void CompandIntrest () {
    int time, pamount;
    Scanner sx = new Scanner (System.in);
    System.out.println ("Enter Principle amount & time
    time = sx.nextInt ();
    pamount = sx.nextInt ();
    double c1 = ((Math.pow ((1+(intrest/12)), time*12)) *
     pamount;
    System.out.println ("Amount to pay monthly : " + c1);
}

public void computeAndDeposit Interest () {
    double Interest = balance * interestRate;
    balance += Interest;
    System.out.println ("Interest:" +Intrest +" Added!"
}

public void withdraw (double amount) {
    if (amount > 0 && amount <= balance) {
        balance -= amount;
        system.out.println ("withdrawn:" + amount);
    }
    else {
    System.out.println ("Insufficient balance / invalid
    }
}
} // End of SavingsAcc
```

```
class CurrAcc extends Account{
    final double balancereq = 500;
    final double serviceCharge = 50;

public CurrAcc(String customerName, int accountNumber, double balance){
    super(customerName, accountNumber, balance);
}

public void Checkbook(double amount){
    if(amount >0 && amount <= balance){
        balance -= amount;
        System.out.println("withdrawn :"+amount);
        if(balance < balancereq){
            balance -= serviceCharge;
        System.out.println("Balance below minimum." +
                           "Service charge 50 deducted");
        }
        else{
        System.out.println("Insufficient Balance or invalid amount");
        }
    }
}//End of Savings class

public class Bank{
    public static void main(String[] xx){
        Scanner Sx = new Scanner(System.in);

    CurAcct a1 = new CurrAcct("Rohit", 12345, 1000);
    System.out.println("Current Account:");
     a1.deposit(200);
     a1.displayBalancle();
     a1.withdraw(800);
     a1.displaybalance();
```

Date __/__/__
Page _____

```
SavAcct a2 = new SavAcct ("Sujay", 67890, 150
System.out.println("Savings Account");
    a2.deposit(500);
    a2.displayBalance();
    a2.withdraw(200);
    a2.displayBalance();
    a2.compoundInterest();
    a2.display
    a2.computeAndDepositInterest();
    a2.displayBalance();
    sx.close();
  }
}
```