5) 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. That 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, acc num and type of acc. 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.

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 minbal, impose penalty if necessary and update the bal.

→ import java.util.Scanner;

class Account {

String customerName;

int accnum;

String acctype;

double bal;

public Account(String customername, int accnum, String acctype) {

this.customername = customername;

this.accnum = accnum

this.acctype = acctype;

this.balance = 0.0;

}

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

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

class Saving extends Account {
   double interestRate;

   public savAcct (String customerName, int accNum,
       double interestRate) {
     super (customerName, accNum, interestRate);
     this.interestRate = interestRate;
   }

public void computeAndDepositInterest ()
  {
   double interest = balance * (interestRate / 100);
   balance += interest;
   System.out.println ("Interest added :" + interest);
   System.out.println ("Updated balance" + bal);
  }
```

```
public void withdraw(double amt){
    if(amt <= bal)
    {
    bal = bal - amt;
    Sout("Amount withdrawn " + amt);
    Sout("Update bal " + bal);
  } else {
     Sout("Insufficient balance");
    }
  }
}

Class CurAcct extends Account {
    double minBal;
    double servicecharge;

public CurAct(String custname, int accnum, double minbal,
    double servicecharge){
  Super(custname, accnum, "current");
    this.minbal = minbal;
    this.servicecharge = servicecharge;
}

public void withdraw(double amt){
  if(amt <= balance){
  balance = bal - amt;
  Sout("withdrawn " + amt);
  if(balance < minbal){
    impose penalty();
  }
  Sout("updated bal " + bal);
  }
    else {
      Sout("Insufficient bal");
  }}
```

```
void imposPenalty(){
    bal -= serviceCharge;
    System.out.println("Balance fell below min, service charge "+serviceCharge);
  }
}

public class Bank{
    public static void main(String[] args)
    Scanner sc = new Scanner(System.in);
    System.out.println("Choose account type 1.Savings 2.current");
    int choice = scanner.nextInt();

    System.out.println("Enter cust name");
    String name = scanner.next();
    System.out.println("Enter acc num");
    int accnum = Scanner.nextInt();

if (choice == 1)
  {
System.out.println("Enter interest rate");
double ir = scanner.nextDouble;
SavAcct s1 = new SavAct(name, accnum, ir);

System.out.println("Enter deposit amt");
double deposit = scanner.nextDouble();
sav.Acc s1.deposit(deposit);

s1.computeAndDepositInterest();
System.out.println("Enter amt to withdraw");
double withdrawamt = scanner.nextDouble();
s1.withdraw(withdrawamt);
```

```
else if (choice == 2) {
        sout ("Enter min bal for acct");
        double minbal = scanner.nextDouble();
        Sout ("Enter service charge for falling below min bal");
        double servicecharge = scanner.nextDouble();
        CurAcct c2 = new CurAcct (name, accNum, minbal, servicecharge);

        sout ("Enter deposit amt");
        double deposit = scanner.nextDouble();
        c2.deposit (deposit);

        Sout ("Enter amt to withdraw");
        double w = scanner.nextDouble();
        c1.withdraw (w))
    }
    else {
        Sout ("Invalid acc type");
    }
}
}
```

Output

1) Choose an account type
1. Savings Account
2. Current Account
1
Enter customer name
Vinay
Enter account number
6533
Enter interest rate for savings account
7.5
Enter amount to deposit
45333
amount deposited : 45333.0
Updated balance : 45333
Interest : 3399.975
updated balance : 48732.975
Enter amount to withdraw
456
Amount withdrawn : 456.0
Updated balance : 48276.975

2) Choose an account type
1. Savings Account
2. Current Account
2
Enter customer name :
Vinay
Enter account number :
4556
Enter min bal for current account :
2000

Enter servic. charge for balance below min balance:
100
Enter amount to deposit:
2000
amount deposited : 2000.0
updated bal : 2000.0
Enter amt to withdraw:
1500
Amount withdrawn 1500.0
Balance below min.. servic charge 100.0
Updated balance 400.0

21/11/24

```java
import java.util.Scanner;

class Account {
     String customerName;
     int accountNumber;
     String accountType;
     double balance;

    public Account(String customerName, int accountNumber, String accountType) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = 0.0;
    }

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

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

class SavAcct extends Account {
    private double interestRate;

    public SavAcct(String customerName, int accountNumber, double interestRate) {
        super(customerName, accountNumber, "Savings");
        this.interestRate = interestRate;
    }

    public void computeAndDepositInterest() {
        double interest = balance * (interestRate / 100);
        balance += interest;
        System.out.println("Interest added: " + interest);
        System.out.println("Updated balance: " + balance);
    }
```

```
    public void computeAndDepositInterest() {
        double interest = balance * (interestRate / 100);
        balance += interest;
        System.out.println("Interest added: " + interest);
        System.out.println("Updated balance: " + balance);
    }

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Amount withdrawn: " + amount);
            System.out.println("Updated balance: " + balance);
        } else {
            System.out.println("Insufficient balance!");
        }
    }
}

class CurAcct extends Account {
   double minimumBalance;
    double serviceCharge;

    public CurAcct(String customerName, int accountNumber, double minimumBalance, double serviceCharge) {
        super(customerName, accountNumber, "Current");
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Amount withdrawn: " + amount);
            if (balance < minimumBalance) {
                imposePenalty();
            }
            System.out.println("Updated balance: " + balance);
        } else {
            System.out.println("Insufficient balance!");
        }
    }

    private void imposePenalty() {
        balance -= serviceCharge;
        System.out.println("Balance fell below minimum. Service charge imposed: " + serviceCharge);
    }
```

```
    private void imposePenalty() {
        balance -= serviceCharge;
        System.out.println("Balance fell below minimum. Service charge imposed: " + serviceCharge);
    }
}

public class bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Choose account type:\n1. Savings Account\n2. Current Account");
        int choice = scanner.nextInt();
        scanner.nextLine();

        System.out.println("Enter customer name: ");
        String name = scanner.nextLine();
        System.out.println("Enter account number: ");
        int accNum = scanner.nextInt();

        if (choice == 1) {
            System.out.println("Enter interest rate for savings account: ");
            double interestRate = scanner.nextDouble();
            SavAcct savAccount = new SavAcct(name, accNum, interestRate);

            System.out.println("Enter amount to deposit: ");
            double deposit = scanner.nextDouble();
            savAccount.deposit(deposit);

            savAccount.computeAndDepositInterest();
            System.out.println("Enter amount to withdraw: ");
            double withdrawAmount = scanner.nextDouble();
            savAccount.withdraw(withdrawAmount);

        } else if (choice == 2) {
            System.out.println("Enter minimum balance for current account: ");
            double minBalance = scanner.nextDouble();
            System.out.println("Enter service charge for falling below minimum balance: ");
            double serviceCharge = scanner.nextDouble();
            CurAcct curAccount = new CurAcct(name, accNum, minBalance, serviceCharge);

            System.out.println("Enter amount to deposit: ");
            double deposit = scanner.nextDouble();
            curAccount.deposit(deposit);

            System.out.println("Enter amount to withdraw: ");
            double withdrawAmount = scanner.nextDouble();
            curAccount.withdraw(withdrawAmount);
```

```
} else if (choice == 2) {
    System.out.println("Enter minimum balance for current account: ");
    double minBalance = scanner.nextDouble();
    System.out.println("Enter service charge for falling below minimum balance: ");
    double serviceCharge = scanner.nextDouble();
    CurAcct curAccount = new CurAcct(name, accNum, minBalance, serviceCharge);

    System.out.println("Enter amount to deposit: ");
    double deposit = scanner.nextDouble();
    curAccount.deposit(deposit);

    System.out.println("Enter amount to withdraw: ");
    double withdrawAmount = scanner.nextDouble();
    curAccount.withdraw(withdrawAmount);

} else {
    System.out.println("Invalid account type selected.");
}

  }
```

```
C:\Users\shett\OneDrive\Documents\javaclasslab>javac bank.java

C:\Users\shett\OneDrive\Documents\javaclasslab>java bank
Choose account type:
1. Savings Account
2. Current Account
1
Enter customer name:
sanath
Enter account number:
123
Enter interest rate for savings account:
7.5
Enter amount to deposit:
45333
Amount deposited: 45333.0
Updated balance: 45333.0
Interest added: 3399.975
Updated balance: 48732.975
Enter amount to withdraw:
456
Amount withdrawn: 456.0
Updated balance: 48276.975

C:\Users\shett\OneDrive\Documents\javaclasslab>java bank
Choose account type:
1. Savings Account
2. Current Account
2
Enter customer name:
virat
Enter account number:
234
Enter minimum balance for current account:
1000
Enter service charge for falling below minimum balance:
100
Enter amount to deposit:
2000
Amount deposited: 2000.0
Updated balance: 2000.0
Enter amount to withdraw:
1500
Amount withdrawn: 1500.0
Balance fell below minimum. Service charge imposed: 100.0
Updated balance: 400.0
```