

name : prema

usn :1bm19cs121

program:

```
import java.util.Scanner;
```

```
class Account{
```

```
    String name;
```

```
    int accountNo;
```

```
    String accountType;
```

```
    double balance;
```

```
    Account(String name,int accountNo,String accountType,double balance){
```

```
        this.name = name;
```

```
        this.accountNo = accountNo;
```

```
        this.accountType = accountType;
```

```
        this.balance = balance;
```

```
}
```

```
    Account(){
```

```
}
```

```
    void DisplayStatus() {
```

```
        System.out.println("****"+this.accountType+"****");
```

```
        System.out.println("Name: "+this.name);
```

```
        System.out.println("Account no.: "+this.accountNo);
```

```
        System.out.println("Account Type: "+this.accountType);
```

```
        System.out.println("Balance: "+this.balance);
```

```
}
```

```
}
```

```
class SavAcct extends Account{  
    double depositAmount;  
    double Withdrawmount;  
    SavAcct(String name,int accountNo,String accountType,double balance){  
        super(name,accountNo,accountType,balance);  
    }  
    SavAcct(String name,int accountNo,String accountType){  
        super();  
    }  
    static Scanner input = new Scanner(System.in);  
    private void checkBalance() {  
        if(balance<0) {  
            System.out.println("Transaction is not possible. Balance becomes less  
than zero");  
            balance+=Withdrawmount;  
            Withdrawmount=0;  
            Withdraw();  
        }  
    }  
    void CallInterest() {  
        System.out.println("Interest To Be added");  
        System.out.println("Annual rate of interest: 4%");  
        System.out.println("Enter the tenure in terms of year");  
        int tenure = input.nextInt();  
        balance = balance*Math.pow(1.04, tenure);  
    }  
    void Deposit() {
```

```

        System.out.println("Enter the Deposit amount");

        depositAmount = input.nextDouble();

        balance+=depositAmount;

    }

    void Withdraw() {

        System.out.println("Enter the Withdrawal amount");

        Withdrawmount = input.nextDouble();

        balance-=Withdrawmount;

        checkBalance();

        System.out.println("Withdraw amount = "+Withdrawmount);

    }

}

```

```

class CurrAcct extends Account{

    double minBalance = 1000;

    double depositAmount;

    double Withdrawmount;

    static Scanner input = new Scanner(System.in);

    CurrAcct(String name,int accountNo,String accountType,double balance){

        super(name,accountNo,accountType,balance);

    }

    private void checkBalance() {

        if(balance<minBalance) {

            System.out.println("Transaction is not possible. Balance becomes less
than minimum balance.");

            balance+=Withdrawmount;

            System.out.println("Do u still want to do the transaction with added
service charges");

```

```
String ans = input.next();

if(ans.toLowerCase().equals("yes")) {

    balance-=(Withdrawmount+(0.05*Withdrawmount)+1000);

    System.out.println("ALERT: Negative balance.\nService Charge
added: "+(0.05*Withdrawmount));

}else {

    Withdrawmount = 0;

}

}

void Deposit() {

    System.out.println("Enter the Deposit amount");

    depositAmount = input.nextDouble();

    balance+=depositAmount;

}

void Withdraw() {

    System.out.println("Enter the Withdrawal amount");

    Withdrawmount = input.nextDouble();

    balance-=Withdrawmount;

    checkBalance();

    System.out.println("withdraw amount = "+Withdrawmount);

}

}

public class BankTest {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);
```

```
System.out.println("Enter the name");

String name = in.next();

System.out.println("Enter the account no.");

int num = in.nextInt();

int i=0;

while(i<2) {

    System.out.println("Enter the account type\ncurr-current acc.\nsav-savings
acct.\t And Balance.");

    String type = in.next();

    if(type.equals("curr")) {

        double bal = in.nextInt();

        CurrAcct c1 = new CurrAcct(name,num,"Current Account",bal);

        c1.DisplayStatus();

        c1.Deposit();

        c1.DisplayStatus();

        c1.Withdraw();

        c1.DisplayStatus();

    }else if(type.toLowerCase().equals("sav")) {

        double bal = in.nextInt();

        SavAcct s1 = new SavAcct(name,num,"Savings Account",bal);

        s1.DisplayStatus();

        s1.Deposit();

        s1.DisplayStatus();

        s1.Withdraw();

        s1.DisplayStatus();

        s1.CallInterest();

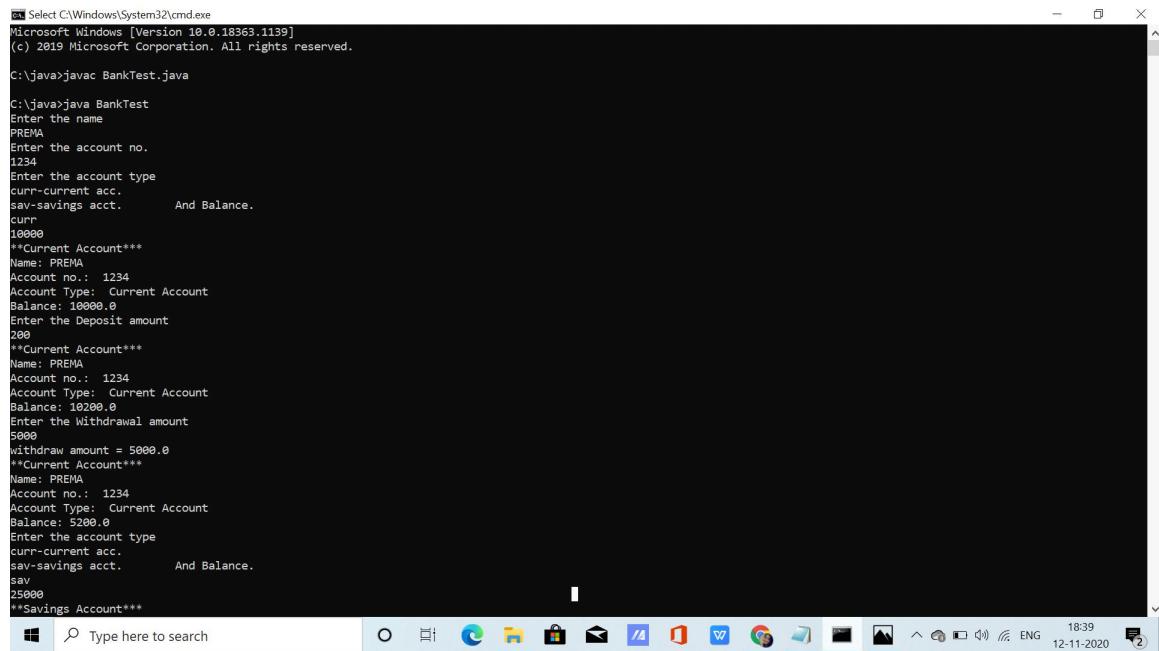
        s1.DisplayStatus();

    }

}
```

```
i++;}in.close();}}
```

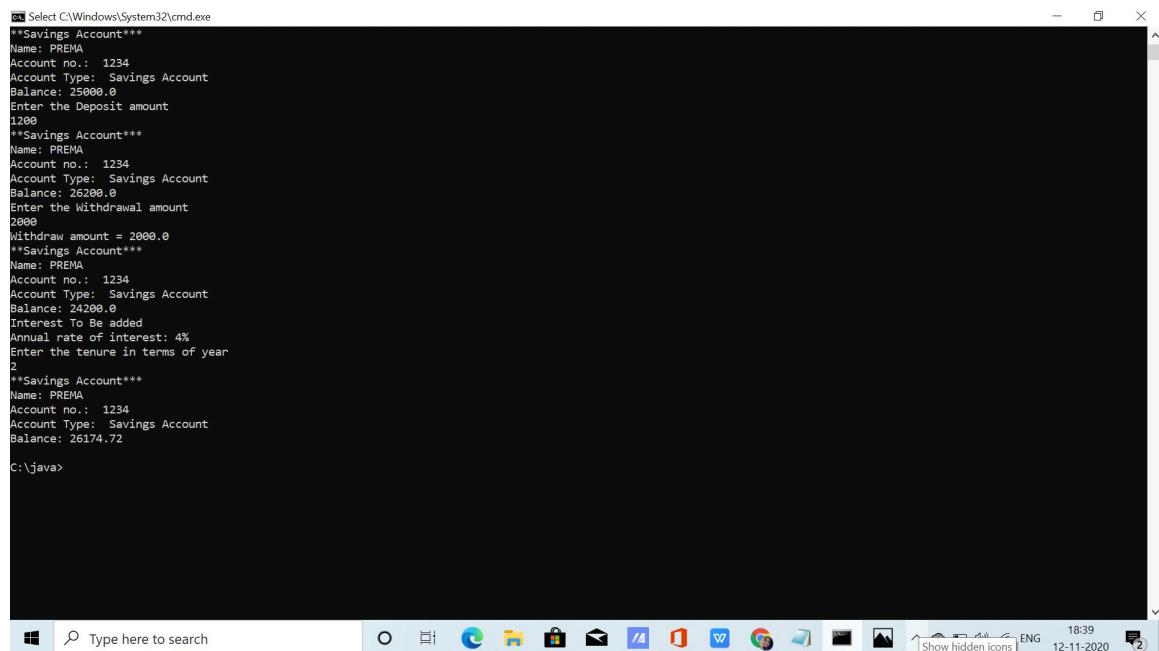
## SCREENSHOTS:



```
c:\ Select C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18363.1139]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\java>javac BankTest.java

C:\java>java BankTest
Enter the name
PREMA
Enter the account no.
1234
Enter the account type
curr-current acc.
sav-savings acct.      And Balance.
curr
10000
**Current Account**
Name: PREMA
Account no.: 1234
Account Type: Current Account
Balance: 10000.0
Enter the Deposit amount
200
**Current Account**
Name: PREMA
Account no.: 1234
Account Type: Current Account
Balance: 10200.0
Enter the Withdrawal amount
5000
withdraw amount = 5000.0
**Current Account**
Name: PREMA
Account no.: 1234
Account Type: Current Account
Balance: 5200.0
Enter the account type
curr-current acc.
sav-savings acct.      And Balance.
sav
25000
**Savings Account**
```



```
c:\ Select C:\Windows\System32\cmd.exe
**Savings Account**
Name: PREMA
Account no.: 1234
Account Type: Savings Account
Balance: 25000.0
Enter the Deposit amount
1200
**Savings Account**
Name: PREMA
Account no.: 1234
Account Type: Savings Account
Balance: 26200.0
Enter the Withdrawal amount
2000
withdraw amount = 2000.0
**Savings Account**
Name: PREMA
Account no.: 1234
Account Type: Savings Account
Balance: 24200.0
Interest To Be added
Annual rate of interest: 4%
Enter the tenure in terms of year
2
**Savings Account**
Name: PREMA
Account no.: 1234
Account Type: Savings Account
Balance: 26174.72

C:\java>
```

## WRITTEN PROGRAMS:

Lab 5.

```
import java.util.*;  
class Account {  
    string name;  
    int accountNo;  
    string accountType;  
    double balance;  
    Account(string name, int accountNo, string  
            accountType, double balance) {  
        this.name = name;  
        this.accountNo = accountType;  
        this.balance = balance;  
    }  
    Account() {}  
    void displayStatus() {  
        System.out.println(" * " + this.accountType + " * ");  
        System.out.println(" Name: " + this.name);  
        System.out.println(" Account No: " + this.accountNo);  
        System.out.println(" Account Type: " + this.accountType);  
        System.out.println(" Balance: " + this.balance);  
    }  
}
```

```

class SavAcct extends Account {
    double depositAmount;
    double withdrawAmount;
    SavAcct ( string name, int accountNo, string accountType,
              double balance) {
        super (name,accountNo,accountType, balance);
    }
    SavAcct (string name, int accountNo, string accountType)
        super();
    static Scanner input = new Scanner (System.in);
    private void checkBalance() {
        if (balance < 0) {
            System.out.println (" Withdrawal amount = +");
            System.out.println (" Transaction not possible ");
            balance += withdrawAmount;
        }
    }
    void withdraw () {
        withdrawAmount = 0;
        withdraw();
    }
    void callInterest () {
        System.out.println (" interest to be added ");
        System.out.println (" Annual rate of interest: A% ");
        System.out.println (" Enter the tenure in terms ");
        System.out.println (" of year ");
        int tenure = input.nextInt ();
        balance = balance * Math.pow (1.04, tenure);
    }
    void deposit () {
        System.out.println (" Enter the Deposit amount ");
        depositAmount = input.nextDouble ();
        balance += depositAmount;
    }
    void withdraw () {
        System.out.println (" Enter the withdrawal amount ");
        withdrawAmount = input.nextDouble ();
        balance -= withdrawAmount;
    }
}
class CurrAcct extends Account {
    double minBalance = 1000;
    double depositAmount;
    double withdrawAmount;
    static Scanner input = new Scanner (System.in);
    CurrAcct ( string name, int accountNo, string acu,
               String accountType, double balance) {
        super (name, accountNo, accountType, balance);
    }
}

```

```

private void checkBalance() {
    if (balance < minBalance) {
        System.out.println("Transaction not possible");
        balance += withdrawalAmount;
        System.out.println("Still want to do");
        String ans = input.next();
        if (ans.toLowerCase().equals("yes")) {
            balance = (withdrawalAmount + (0.05 * withdrawalAmount)) +
                1000;
            System.out.println("Service charge " + 0.05 * withdrawalAmount);
        } else {
            withdrawalAmount = 0;
        }
    }
}

void deposit() {
    depositAmount = input.nextDouble();
    balance += depositAmount;
}

void withdraw() {
    System.out.println("Enter withdraw amount");
    balance -= withdrawal;
    checkBalance();
}

```

```

public class BankTest {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter name");
        String name = in.next();
        System.out.println("Enter account no.");
        int num = in.nextInt();
        int i = 0;
        while (i < 2) {
            String type = in.next();
            if (type.equals("curr")) {
                double bala = in.nextInt();
                currAcct c1 = new currAcct(name, num, "curr");
                c1.DisplayStatus();
                c1.Deposit();
                c1.DisplayStatus();
                c1.Withdraw();
                c1.DisplayStatus();
            } else if (type.equals("sav")) {
                saveAcc s1 = new saveAcc(name, num, "sav");
                s1.DisplayStatus();
                s1.Deposit();
                s1.Withdraw();
                s1.callInterest();
                s1.DisplayStatus();
            }
            i++;
        }
    }
}

```