

1. Create **Account** class with the details , Account Number, Name, Balance. Account number need to be generated automatically ( use static with the initial value as 20100). Create another class **Transaction** with the details Account number, amount transacted, Credit/ debit, date (as string). Write appropriate functions to perform the following:
  - Create N objects for **Account** class. Use constructor to initialize values. Minimum balance should be 1000 while creating objects.
  - Create user interaction method to ask for either “withdraw or deposit”. Then ask for account number, amount, and date. Store the transaction as account number, amount, “credit” if it is deposit and “debit” If it is withdrawal and date in **Transaction** class object. Update the amount in the corresponding **Account** class object. While withdrawal check for the minimum balance after deducting.
  - Display total amount of credit and debit in every date.
  - Display the transaction history of a particular account number.
  - Display all the credit details and debit details separately in the ascending order of the amount. Print two separate list.

## **CODE**

```
import java.io.*;
import java.util.*;
import java.lang.*;

class Account
{
```

```

int accNumber;
String name;
double bal;
static int acGen=20100;

Account(String name, double bal)
{
    if(bal<1000)
    {
        System.out.println("Minimum Account balance is: $1000");
    }
    this.bal=bal;
    this.name=name;
    this.accNumber=acGen++;
    System.out.println("Acc no: "+this.accNumber);
}

void printDatewise(Transaction ...tList)
{
    // System.out.println(tList.length);
    String dates[]=new String[tList.length];
    double credit_amnts[]=new double[tList.length];
    double debit_amnts[]=new double[tList.length];
    int cnt=0;
    for(Transaction t: tList)
    {
        boolean flag=true;
        int idx=0;
        for(int i=0;i<cnt;i++)
        {
            if(t.date.equals(dates[i]))
            {
                idx=i;
                flag =false;
                break;
            }
        }
        if(flag)
        {
            dates[cnt]=t.date;
            if(t.type.equals("debit")) debit_amnts[cnt]=t.amt;
            else if(t.type.equals("credit")) credit_amnts[cnt]=t.amt;
            cnt++;
        }
        else
        {
            if(t.type.equals("debit")) debit_amnts[idx]+=t.amt;
            else if(t.type.equals("credit")) credit_amnts[idx]+=t.amt;
        }
    }
    for(int i=0;i<cnt;i++)
    {
        System.out.println("Date: "+dates[i]);
        System.out.println("Total credit is: "+credit_amnts[i]);
        System.out.println("Total debit is: "+debit_amnts[i]);
    }
}

```

```

    }
    void transactionHistory(Transaction ...tList)
    {
        System.out.println("Transaction History for "+this.accNumber+" :");
        for(Transaction t: tList)
        {
            if(t.accNumber==this.accNumber)
            {
                System.out.println(t.type+"\t"+t.amt+"\t"+t.date);
            }
        }
    }
    void creditDebitDetails(Transaction ...tList)
    {
        System.out.println("Credit History for "+this.accNumber+" :");
        for(Transaction t: tList)
        {
            if(t.accNumber==this.accNumber && t.type.equals("credit"))
            {
                System.out.println(t.type+"\t"+t.amt+"\t"+t.date);
            }
        }
        System.out.println("Debit History for "+this.accNumber+" :");
        for(Transaction t: tList)
        {
            if(t.accNumber==this.accNumber && t.type.equals("debit"))
            {
                System.out.println(t.type+"\t"+t.amt+"\t"+t.date);
            }
        }
    }
}

class Transaction
{
    int accNumber;
    String date="";
    double amt;
    String type="";
    private static Account []ac;
    Transaction(Account []ac)
    {
        this.type=uiPanel();
        this.ac=ac;
        this.transcationController();
    }

    String uiPanel()
    {
        System.out.println("Welcome to Chiya International Bank!");
        System.out.print("Please enter Transaction Type \n[ cancel | withdraw |
deposit ]: ");
        Scanner sc=new Scanner(System.in);
        String type= sc.nextLine();
        // sc.close();
    }
}

```

```

        return type;
    }
    private void transactionController()
    {
        switch(this.type)
        {
            case "withdraw":
                this.type="debit";
                this.withdraw();
                break;
            case "deposit":
                this.type="credit";
                this.deposit();
                break;
            case "cancel":
                System.out.println("You Choose Cancel! Thank You :)");
                this.type="cancelled";
                this.type="failed";
                break;
            default:
                System.out.println("Invalid Transaction Type! Transaction Failed
:(");
                this.type="failed";
                break;
        }
    }
    private void withdraw()
    {
        getAccData();
        updateAccData(-this.amt);
    }
    private void deposit()
    {
        getAccData();
        updateAccData(this.amt);
    }
    private void getAccData()
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter Account Number: ");
        this.accNumber=sc.nextInt();
        System.out.print("Enter Amount: ");
        this.amt=sc.nextDouble();
        System.out.print("Enter date: ");
        this.date=sc.next();
    }
    private void updateAccData(double amt)
    {
        boolean flag1=false,flag2=false;
        if(ac!=null)
        for(int i=0;i<this.ac.length;i++)
        {
            if(this.ac[i]!=null)
            {
                if(this.ac[i].accNumber==this.accNumber)

```

```

        {
            flag1=true;
            if(this.ac[i].bal+amt>=1000)
            {
                flag2=true;
                this.ac[i].bal+=amt;
            }
        }
    }

    if(!flag1)
    {
        System.out.println("Account Number doesn't Exist!");
        this.type="failed";
        return;
    }
    if(!flag2)
    {
        System.out.println("Minimum Account Balance to be kept is $1000!");
        this.type="failed";
        return;
    }
}

public class ques1 {

    public static void main(String[] args) {

        Account ac[]=new Account[3];
        ac[0]=new Account("Chiran",21000);
        ac[1]=new Account("Priya",12000);
        ac[2]=new Account("Nancy",10000);
        Transaction []t=new Transaction[4];
        for(int i=0;i<t.length;i++)
        {
            t[i]=new Transaction(ac);
        }
        ac[0].transactionHistory(t);
        ac[0].creditDebitDetails(t);
        ac[0].printDatewise(t);
    }

}

```

## **OUTPUT**

```
<terminated> ques1 [Java Application] C:\Program Files\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_15.0.1.v20201027-0507\jre\bin\javaw.exe (23-Feb-2021, 6:34:29 pm - 6:38:30 pm)
Acc no: 20100
Acc no: 20101
Acc no: 20102
Welcome to Chiya International Bank!
Please enter Transaction Type
[ cancel | withdraw | deposit ]: deposit
Enter Account Number: 20100
Enter Amount: 50
Enter date: 2020-02-23
Welcome to Chiya International Bank!
Please enter Transaction Type
[ cancel | withdraw | deposit ]: deposit
Enter Account Number: 20100
Enter Amount: 100
Enter date: 2020-02-22
Welcome to Chiya International Bank!
Please enter Transaction Type
[ cancel | withdraw | deposit ]: withdraw
Enter Account Number: 20100
Enter Amount: 500
Enter date: 2020-02-23
Welcome to Chiya International Bank!
Please enter Transaction Type
[ cancel | withdraw | deposit ]: withdraw
Enter Account Number: 20100
Enter Amount: 700
Enter date: 2020-02-23
Transaction History for 20100 :
credit 50.0 2020-02-23
credit 100.0 2020-02-22
debit 500.0 2020-02-23
debit 700.0 2020-02-23
Credit History for 20100 :
credit 50.0 2020-02-23
credit 100.0 2020-02-22
Debit History for 20100 :
debit 500.0 2020-02-23
debit 700.0 2020-02-23
Date: 2020-02-23
Total credit is: 50.0
Total debit is: 1200.0
Date: 2020-02-22
Total credit is: 100.0
Total debit is: 0.0
```

2. Create class for Electricity for N flats of an apartment. The required details are Flat number, EB meter number, Owner name, Previous month reading( in units), current month reading ( in Units), amount. Except amount and EB meter number remaining values will be accepted from the user. EB meter number could be random number generated between 10000 to 20000. Amount will be calculates as below:

0-100 Minimum charges =100

100-500 2.5 ₹ per unit

**500-1000 5.0 ₹ per unit**

**Above 1000 7.5 ₹ per unit**

Write appropriate functions to perform the following:

- Display the flat and owner details who paid maximum and minimum EB charges.
- Display the average amount paid by all flats in each floor. Floor will be identified from the first digit of a flat number. For 1001 – first floor , 2005 – second floor, 5010- Fifth floor. Display the floor wise details.
- Display the Flat number, EB amount along with the meter number in the ascending order of the meter number.

## **CODE**

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

class Electricity
{
    int FlatNo;
    String OwnerName;
    int PrevReading,CurrReading;
    double amount;

    int meterNo;

    Electricity(int meterNo)
    {
        this.meterNo=meterNo;
    }

    void input()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the Owner Name: ");
        OwnerName=sc.next();
        System.out.println("Enter the Flat number: ");
        FlatNo=sc.nextInt();
        System.out.println("Enter the Previous Month Reading: ");
```

```

        PrevReading=sc.nextInt();
        System.out.println("Enter the Current Month Reading: ");
        CurrReading=sc.nextInt();
    }

    void cal()
    {
        double diff=CurrReading-PrevReading;
        if(diff>=0 && diff<=100)
            amount=100;
        else if(diff>100 && diff<=500)
            amount=diff*2.50;
        else if(diff>500 && diff <=1000)
            amount=diff*5.0;
        else
            amount=diff*7.5;
    }

    void display()
    {
        System.out.println("Flat Number: "+FlatNo);
        System.out.println("Owner Name: "+OwnerName);
        System.out.println("Meter No: " + meterNo);
        System.out.println("Consumer Previous Units: "+PrevReading);
        System.out.println("Consumer Current Units: "+CurrReading);
        System.out.println("Total Amount(Rs.): "+amount);
    }
    static void displayMaxMin(Electricity a[])
    {
        double max=0,min=Double.POSITIVE_INFINITY;
        for(int i=0;i<a.length;i++)
        {
            max=a[i].amount>max ? a[i].amount : max;
            min=a[i].amount<min ? a[i].amount : min;
        }
        for(int i=0;i<a.length;i++)
        {
            if(a[i].amount==max)
            {
                System.out.println("Maximum EB Charges!");
                a[i].display();
            }
            if(a[i].amount==min)
            {
                System.out.println("Minimum EB Charges!");
                a[i].display();
            }
        }
    }
    static void avgAmtByFloorWise(Electricity []a)
    {

```



```

        double avg[]=new double[9];
        int cnt[]=new int[9];
        for(int i=0;i<a.length;i++)
        {
            int flat = a[i].FlatNo;
            while(flat >= 10)
                flat = flat / 10;
            avg[flat-1]+=a[i].amount;
            cnt[flat]++;
        }
        for(int i=0;i<9;i++)
        {
            if(cnt[i]>0)
                avg[i]/=cnt[i];
        }
        for(int i=0;i<9;i++)
        {
            if(avg[i]!=0)
                System.out.println("Average Bill for Floor [ "+(i+1)+" ] :
"+avg[i]);
        }
    }
    static void displayByMeter(Electricity a[])
    {
        int arr[]=new int[a.length];
        for(int i=0;i<a.length;i++)
        {
            arr[i]=a[i].meterNo;
        }

        for(int i=0;i<a.length;i++)
        {
            int key=arr[i];
            int j=i-1;
            while(j>=0 && arr[j]>key)
            {
                arr[j+1]=arr[j];
                j--;
            }
            arr[j+1]=key;
        }

        for(int i=0;i<arr.length;i++)
        {
            for(int j=0;j<a.length;j++)
            {
                if(arr[i]==a[j].meterNo)
                {
                    System.out.println("For Meter No "+a[j].meterNo+" Amount is
"+a[j].amount+" and Flat Number is "+a[j].FlatNo);
                }
            }
        }
    }
}

```

```

}
public class ques2 {

    public static void main(String[] args) {
        Random r=new Random();
        int rn;
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the number of flats: ");
        int N=s.nextInt();
        Electricity []a=new Electricity[N];
        for(int i=0;i<N;i++)
        {
            rn=r.nextInt((20000-10000))+10000;
            System.out.println("EB Meter Number: "+rn);

            a[i]=new Electricity(rn);
            a[i].input();
            a[i].cal();
            System.out.println("-----");
        }
        Electricity.displayMaxMin(a);
        Electricity.avgAmtByFloorWise(a);
        Electricity.displayByMeter(a);

    }

}

```

**OUTPUT**

```
<terminated> ques2 [Java Application] C:\Program Files\ eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_15.0.1.v20201027-0507\jre\bin\javaw.exe (23-Feb-2021, 10:54:13 pm - 10:56:51 pm)
Enter the number of flats:
3
EB Meter Number: 15970
Enter the Owner Name:
priya
Enter the Flat number:
3101
Enter the Previous Month Reading:
300
Enter the Current Month Reading:
1000
-----
EB Meter Number: 14571
Enter the Owner Name:
chiran
Enter the Flat number:
3110
Enter the Previous Month Reading:
900
Enter the Current Month Reading:
1500
-----
EB Meter Number: 15610
Enter the Owner Name:
chiya
Enter the Flat number:
6123
Enter the Previous Month Reading:
500
Enter the Current Month Reading:
2000
-----
Minimum EB Charges!
Flat Number: 3110
Owner Name: chiran
Meter No: 14571
Consumer Previous Units: 900
Consumer Current Units: 1500
Total Amount(Rs.): 3000.0
Maximum EB Charges!
Flat Number: 6123
Owner Name: chiya
Meter No: 15610
Consumer Previous Units: 500
Consumer Current Units: 2000
Total Amount(Rs.): 11250.0
Average Bill for Floor [ 3 ] : 6500.0
Average Bill for Floor [ 6 ] : 11250.0
For Meter No 14571 Amount is 3000.0 and Flat Number is 3110
For Meter No 15610 Amount is 11250.0 and Flat Number is 6123
For Meter No 15970 Amount is 3500.0 and Flat Number is 3101
```