ElectricityBill.cs

```
using System; using
System.Collections.Generic; using
System.Linq; using System.Text;
using System.Threading.Tasks;

namespace BillAutomation    //DO NOT change the namespace name
{
   public class ElectricityBill        //DO NOT change the class name
   {
     //Implement the fields and properties as per description

     private String consumerNumber;
private String consumerName;
private int unitsConsumed;
private double billAmount;       public
string ConsumerNumber
     {
        get{return this.consumerNumber;}
set{if(value.Substring(0,2).Equals("EB")){
this.consumerNumber=value;}
           else
           throw new FormatException("Invalid Consumer Number");
        }
     }
     public string ConsumerName{get;set;}
public int UnitsConsumed{get;set;}       public
double BillAmount{get;set;}

   }
```

```
}

ElectricityBoard.cs using System;

using System.Collections; using

System.Collections.Generic; using

System.Data; using

System.Data.SqlClient; using

System.Linq; using System.Text;

using System.Threading.Tasks;


namespace BillAutomation      //DO NOT change the namespace name

{

   public class ElectricityBoard  //DO NOT change the class name

   {

      //Implement the property as per the description

public SqlConnection SqlCon{get;set;}


      //Implement the methods as per the description

      public void AddBill(ElectricityBill ebill)

      {

try{

            string qstr=$"insert into ElectricityBill
values('{ebill.ConsumerNumber}','{ebill.ConsumerName}','{ebill.UnitsConsumed}','{ebill.BillA
mount}')";


            SqlCon.Open();

            SqlCommand cmd = new SqlCommand(qstr,SqlCon);

            SqlDataReader reader = cmd.ExecuteReader();

            SqlCon.Close();

         }
```

```
        catch(Exception e)

        {

            Console.WriteLine("Error : "+e.Message);

        }


    }

    public void CalculateBill(ElectricityBill ebill)

    {

        if(ebill.UnitsConsumed<=100) {ebill.BillAmount=0;}             else
if(ebill.UnitsConsumed>100 && ebill.UnitsConsumed<=300)

        {

            int temp = ebill.UnitsConsumed-100;
ebill.BillAmount=temp*1.5;

        }

        else if(ebill.UnitsConsumed>300 && ebill.UnitsConsumed<=600)

        {

            int temp200 = ebill.UnitsConsumed-100;
int temp300 = temp200-200;
ebill.BillAmount=(200*1.5)+(temp300*3.5);

        }

        else if(ebill.UnitsConsumed>600 && ebill.UnitsConsumed<=1000)

        {

            int temp200 = ebill.UnitsConsumed-100;
int temp400 = temp200-500;
            ebill.BillAmount=(200*1.5)+(300*3.5)+(temp400*5.5);

        }

        else

        {
             int temp200 = ebill.UnitsConsumed-100;               int temp400 =
temp200-900;
ebill.BillAmount=(200*1.5)+(300*3.5)+(400*5.5)+(temp400*7.5);
          }
       }
       public List<ElectricityBill> Generate_N_BillDetails(int num)
       {
try{
            string query="Select TOP "+num+"*from ElectricityBill order by consumer_number
desc";
            SqlCon.Open();
             SqlCommand cmd=new SqlCommand(query,SqlCon);
            SqlDataReader    reader=cmd.ExecuteReader();
List<ElectricityBill>l1=new             List<ElectricityBill>();
while(reader.Read())
            {
               ElectricityBill eb1=new ElectricityBill();
               eb1.ConsumerNumber=reader[0].ToString();
eb1.ConsumerName=reader[1].ToString();
eb1.UnitsConsumed=(int)reader[2];                 eb1.BillAmount=(double)reader[3];
               l1.Add(eb1);


            }
            SqlCon.Close();
return l1;
         }catch(Exception e)
         {
            Console.WriteLine("Error 1:"+e.Message);
         }
```

```
           return null;
       }





    }
}
```

DBHandler.cs using System;

```
using System.Collections.Generic;
using System.Linq; using
System.Text; using
System.Threading.Tasks; using
System.Data.SqlClient; using
System.Configuration;

namespace BillAutomation      //DO NOT change the namespace name
{
    public class DBHandler    //DO NOT change the class name
    {
        //Implement the methods as per the description
public DBHandler(){}        public SqlConnection
GetConnection()
        {
            SqlConnection con = null;
            string connection =
System.Configuration.ConfigurationManager.ConnectionStrings["MyCon"].ConnectionString
;
            con = new SqlConnection(connection);
            return con;
        }
    }
}
```

Program.cs using System; using

```
System.Collections.Generic; using
System.Linq; using System.Text;
using System.Threading.Tasks;
using System.Data.SqlClient;
using System.Collections; using
System.Data; using
System.Configuration;

namespace BillAutomation        //DO NOT change the namespace name
{
    public class Program       //DO NOT change the class name
    {

        static void Main(string[] args)  //DO NOT change the 'Main' method signature
        {
            //Implement the code here
            ElectricityBoard eb=null;
            DBHandler db= new DBHandler();
            SqlConnection con = db.GetConnection();
            List<ElectricityBill> l2 = new List<ElectricityBill>();
Console.WriteLine("Enter Number of Bills To Be Added :");            int
totBill=Convert.ToInt32(Console.ReadLine());
            for(int i=0;i<totBill;i++)
            {
```

```
         Console.WriteLine("Enter Consumer Number:");
         String conNo=Console.ReadLine();
         Console.WriteLine("Enter Consumer Name:");
        String conName=Console.ReadLine();
       unitsCon:
         Console.WriteLine("Enter Units Consumed:");
         int unit=Convert.ToInt32(Console.ReadLine());
         while(unit<0)
         {
           Console.WriteLine("Given units is invalid");
           goto unitsCon;
         }ElectricityBill ebill = new ElectricityBill();
         try{
           ebill.ConsumerNumber = conNo;
         }
         catch(FormatException e)
         {
           Console.WriteLine(e);
         }
         ebill.ConsumerName = conName;
ebill.UnitsConsumed = unit;                eb=
new ElectricityBoard();                eb.SqlCon
= con;                eb.CalculateBill(ebill);
eb.AddBill(ebill);                l2.Add(ebill);


       }
       Console.WriteLine();
```

```
            Console.WriteLine("Enter Last 'N' Number of Bills To Generate:");
int num = Convert.ToInt32(Console.ReadLine());
Console.WriteLine();            foreach(var p in l2)
            {
               Console.WriteLine(((ElectricityBill)p).ConsumerNumber);
               Console.WriteLine(((ElectricityBill)p).ConsumerName);
               Console.WriteLine(((ElectricityBill)p).UnitsConsumed);
               Console.WriteLine("Bill Amount:"+((ElectricityBill)p).BillAmount);
            }
            List<ElectricityBill> l1 = eb.Generate_N_BillDetails(num);
Console.WriteLine("Details of last 'N' bills");            foreach(var
p in l1)
            {
               Console.WriteLine("EB Bill for "+((ElectricityBill)p).ConsumerName+" is
"+((ElectricityBill)p).BillAmount);
            }
         }
   }


   public class BillValidator{      //DO NOT change the class name


      public String ValidateUnitsConsumed(int UnitsConsumed)      //DO NOT change the
method signature
      {
         if(UnitsConsumed<0)
         {
            return "Given units is invalid";
         }
         return "";
      }
```

```
    }
}
```