/// Before namespace

/* Weekly Project- 1 - Using OOPs
Problem Statement:
LESCO wants to calculate electricity bill according to the below rules:
The program will ask user for total unit consumed. Ask the user if he is a registered tax payer.
• Residential: If the unit consumed less or equal to 100 units, calculates the total amount of units*5. Commercial: If the unit consumed less or equal to 100 units, calculates the total amount of units*8.
• Residential: If the unit consumed are between 100 to 200, then the rate for units consumed after 100 units is 17/per unit. Commercial: If the unit consumed are between 100 to 200, then the rate for units consumed after 100 units is 21/per unit.

If the unit consumed are between 200 to 500, then the rate for units consumed after 200 units is 23/per unit. If the user is a tax payer then user previous rates for this slab.
• Residential: If the unit consumed are more than 500, then the rate for units consumed after 500 units is 69/per unit. Commercial: If the unit consumed are more than 500, then the rate for units consumed after 500 units is 79/per unit.
• After the calculation, add 17% tax for commercial and 13% tax for residential in the calculated amount.
• By default, all the users are non-tax payers.
*/

```
/*
    All Classes Used in the Program
    */

   // Parent Abstract Class for Electricity Consumer
   abstract class Consumer
   {
      // Defining Class Attributes/Variables
      public long unitsConsumed;
      public bool taxpayerStatus;
      public double electricityBill;

      // Class Functions
      public abstract double calBill();
   }

   // Child Class for a Residential Consumer
   class ResidentialConsumer : Consumer // Stating that it's child class of Parent "Consumer"
```

```
  {
    // default constructor
    public ResidentialConsumer(long userInputUnits = 0, bool userTaxpayerStatus = false)
    {
      unitsConsumed = userInputUnits;
      taxpayerStatus = userTaxpayerStatus;
      electricityBill = 0;
    }

    // Electricity Bill Calculation based on the Tariff Provided in Question
    public override double calBill()
    {
      // 1st Slab
      if (unitsConsumed <= 100)
      {
        electricityBill = unitsConsumed * 5;
      }

      // 2nd Slab
      if (unitsConsumed > 100 && unitsConsumed <= 200)
      {
        electricityBill = 100 * 5; // Calculation for first 100 units
        long extraUnits = unitsConsumed - 100;
        electricityBill = extraUnits * 17; // Calculation for units after 100
      }

      // 3rd Slab for Taxpayers
      if (unitsConsumed > 200 && unitsConsumed <= 500 && taxpayerStatus == true)
      {
        electricityBill = 100 * 5; // Calculation for first 100 units
        long extraUnits = unitsConsumed - 100; // Rate for units after 100 is same for tax
payer
        electricityBill = electricityBill + extraUnits * 17;
      }

      // 3rd Slab for Non-Tax Payers
      if (unitsConsumed > 200 && unitsConsumed <= 500 && taxpayerStatus == false)
      {
        electricityBill = 100 * 5; // Calculation for first 100 units
        electricityBill = electricityBill + 100 * 17; // Calculation for second 100 units
        long extraUnits = unitsConsumed - 200;
        electricityBill = electricityBill + 23 * extraUnits;
      }
```

```
        // 4th Slab for Tax Payers
        if (unitsConsumed > 500 && taxpayerStatus == true)
        {
            electricityBill = 100 * 5; // Calculation for first 100 units
            electricityBill = electricityBill + 400 * 17; // Calculation for next 400 units for TaxPayer
            long extraUnits = unitsConsumed - 500;
            electricityBill = electricityBill + 23 * extraUnits;

        }

        // 4th Slab for Non-Tax Payers
        if (unitsConsumed > 500 && taxpayerStatus == false)
        {
            electricityBill = 100 * 5; // Calculation for first 100 units
            electricityBill = electricityBill + 100 * 17; // Calculation for next 100 units
            electricityBill = electricityBill + 300 * 23; // Calculation for next 300 units
            long extraUnits = unitsConsumed - 500;
            electricityBill = electricityBill + 69 * extraUnits;
        }

        // Calculating Tax for Residential Consumers
        double totalTax = electricityBill * 0.13;
        electricityBill = electricityBill + totalTax;

        return electricityBill;

    }
}

// Child Class for Commericial Consumer
class CommercialConsumer : Consumer
{

    // default constructor
    public CommercialConsumer(long userInputUnits = 0, bool userTaxpayerStatus = false)
    {
        unitsConsumed = userInputUnits;
        taxpayerStatus = userTaxpayerStatus;
        electricityBill = 0;
    }

    // Electricity Bill Calculation based on the Tariff Provided in Question
    public override double calBill()
    {
```

```
        // 1st Slab
        if (unitsConsumed <= 100)
        {
            electricityBill = 8 * unitsConsumed;
        }

        // 2nd Slab
        if (unitsConsumed > 100 && unitsConsumed <= 200)
        {
            electricityBill = 100 * 8; // Calculation for first 100 units
            long extraUnits = unitsConsumed - 100;
            electricityBill = extraUnits * 21; // Calculation for units after 100
        }

        // 3rd Slab for Commercial Taxpayers
        if (unitsConsumed > 200 && unitsConsumed <= 500 && taxpayerStatus == true)
        {
            electricityBill = 100 * 8; // Calculation for first 100 units
            long extraUnits = unitsConsumed - 100; // Rate for units after 100 is same for tax
payer
            electricityBill = electricityBill + extraUnits * 21;
        }

        // 3rd Slab for Commercial Non-Tax Payers
        if (unitsConsumed > 200 && unitsConsumed <= 500 && taxpayerStatus == false)
        {
            electricityBill = 100 * 8; // Calculation for first 100 units
            electricityBill = electricityBill + 100 * 21; // Calculation for second 100 units
            long extraUnits = unitsConsumed - 200;
            electricityBill = electricityBill + 23 * extraUnits;
        }

        // 4th Slab for Tax Payers
        if (unitsConsumed > 500 && taxpayerStatus == true)
        {
            electricityBill = 100 * 8; // Calculation for first 100 units
            electricityBill = electricityBill + 400 * 21; // Calculation for next 400 units for TaxPayer
            long extraUnits = unitsConsumed - 500;
            electricityBill = electricityBill + 79 * extraUnits;

        }

        // 4th Slab for Non-Tax Payers
        if (unitsConsumed > 500 && taxpayerStatus == false)
```

```
        {
            electricityBill = 100 * 8; // Calculation for first 100 units
            electricityBill = electricityBill + 100 * 21; // Calculation for next 100 units
            electricityBill = electricityBill + 300 * 23; // Calculation for next 300 units
            long extraUnits = unitsConsumed - 500;
            electricityBill = electricityBill + 79 * extraUnits;
        }

        // Calculating Tax for Commercial Consumers
        double totalTax = electricityBill * 0.17;
        electricityBill = electricityBill + totalTax;

        return electricityBill;

    }
  }
  // Main Program Starts



 long unitsConsumed=0;
        bool consumerType=false ;
        bool taxpayerStatus=false;

        // Taking User Input
        Console.WriteLine("Enter the no. of Electricity units you consume Monthly");
        unitsConsumed=Convert.ToInt32(Console.ReadLine());
        // Asking about the Customer Type
        Console.WriteLine("Are you are Residential or Commericial Consumer? Press 1 for
residential and 0 for Commericial");
        int consumerVariable = Convert.ToInt32(Console.ReadLine());
        if (consumerVariable==1)
        {
            consumerType=true;
        }
        // Asking about Tax Status
        Console.WriteLine("Are you are TaxPayer? Press 1 if you are Tax-Payer and 0 if you are
not");
        int taxVariable = Convert.ToInt32(Console.ReadLine());
        if (taxVariable==1)
        {
            taxpayerStatus=true;
        }
        // Bill Calculation by creating an Object
```

```
        Consumer C1 = GetBill(consumerType, unitsConsumed, taxpayerStatus);

        // Displaying the Electricity Bill
        Console.WriteLine("Your electricity Electricity Bill is : " + C1.calBill());

        // Ending Program
        Console.WriteLine("\nThe program has finished. Press any key to EXIT.");
        Console.ReadKey();



private static Consumer GetBill(bool customerType, long unitsConsumed, bool taxpayerStatus)
    {
        if (customerType == false)
        {
            return new ResidentialConsumer(unitsConsumed, taxpayerStatus);
        }
        else
        {
            return new CommercialConsumer(unitsConsumed, taxpayerStatus);
        }
    }
```