1.

a. Create a Person class with Age as a property and read the age with the property

b. Implement a user defined exception to handle Age eligibility to vote (18-120 years eligible.

Others not eligible)

//Stub Code

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

namespace Assessment
{

    internal class Person
    {
        private int age;

        public int Age
        {
            get => age;
            set => age = value;
        }

        public void CheckAge(int age)
        {

                if (age >= 18 && age <= 120)
                {
                    Console.WriteLine("You are Eligible to vote");
                }
                else
                {
                    throw new ArithmeticException ("You must be at least 18 years old to vote.");
                }



        }

    }


}
```

// Driver Code

```
using Assessment;

class Program
{
    public static void Main(string[] args)
    {

        Console.Write("Enter Age :");
        int age = Convert.ToInt32(Console.ReadLine());
        Person person = new Person();
        try
```

```
        {
            person.CheckAge(age);
        }
        catch (ArithmeticException ex)
        {
            Console.WriteLine("Sorry You are not eligible, " + ex.Message);
        }


    }
}
```

```
Enter Age :15
Sorry You are not eligible, You must be at least 18 years old to vote.

D:\C#\Programs\Assessment\Assessment\bin\Debug\net6.0\Assessment.exe (process 25352) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

```
Enter Age :72
You are Eligible to vote

D:\C#\Programs\Assessment\Assessment\bin\Debug\net6.0\Assessment.exe (process 13508) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

c. Implement a Currency Converter Class with choice for at least 5 conversions. Ask choice for conversion from the users and implement class(es) with suitable methods to display the converted currency value.

//Stub Code

```
internal class CurrencyConverter
{
    private double inrtousd = 0.012;
    private double inrtoyen = 1.63;
    private double inrtoeuro = 0.011;
    private double inrtosgd = 0.016;
    private double inrtolkr = 3.89;

    public double Inrtousd { get => inrtousd; set => inrtousd = value; }
    public double Inrtoyen { get => inrtoyen; set => inrtoyen = value; }
    public double Inrtoeuro { get => inrtoeuro; set => inrtoeuro = value; }
    public double Inrtosgd { get => inrtosgd; set => inrtosgd = value; }
    public double Inrtolkr { get => inrtolkr; set => inrtolkr = value; }

    public double ConvertInrToUsd(double inr)
    {
        return inr * inrtousd;
    }

    public double ConvertInrToYen(double inr)
    {
        return inr * inrtoyen;
    }
```

```
        public double ConvertInrToEuro(double inr)
        {
            return inr * inrtoeuro;
        }

        public double ConvertInrToSgd(double inr)
        {
            return inr * inrtosgd;
        }

        public double ConvertInrToLkr(double inr)
        {
            return inr * inrtolkr;
        }
    }
//Driver Code

class Program
{
    public static void Main(string[] args)
    {


  CurrencyConverter converter = new CurrencyConverter();


        Console.WriteLine("Please select a currency conversion:");
        Console.WriteLine("1. INR to USD");
        Console.WriteLine("2. INR to YEN");
        Console.WriteLine("3. INR to EURO");
        Console.WriteLine("4. INR to SGD");
        Console.WriteLine("5. INR to LKR");

        Console.Write("Enter Choice : ");
        int choice = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("Enter the amount in Inr");
        double inr = Convert.ToDouble(Console.ReadLine());

        switch (choice)
        {
            case 1: Console.WriteLine("INR " + inr + "= Usd " + converter.ConvertInrToUsd(inr));
                break;
            case 2:
                Console.WriteLine("INR " + inr + "= Yen " + converter.ConvertInrToYen(inr));
                break;
            case 3:
                Console.WriteLine("INR " + inr + "= Euro " + converter.ConvertInrToEuro(inr));
                break;
            case 4:
                Console.WriteLine("INR " + inr + "= Sgd " + converter.ConvertInrToSgd(inr));
                break;
            case 5:
                Console.WriteLine("INR " + inr + "= Lkr " + converter.ConvertInrToLkr(inr));
                break;


        }


    }
```

```
Please select a currency conversion:
1. INR to USD
2. INR to YEN
3. INR to EURO
4. INR to SGD
5. INR to LKR
Enter Choice : 1
Enter the amount in Inr
85
INR 85= Usd 1.02

D:\C#\Programs\Assessment\Assessment\bin\Debug\net6.0\Assessment.exe (process 24820) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

2.

a. Implement an EB Calculator class with suitable methods to calculate the bill. The class should hold the methods EnterReading() & CalculateBill(). Use appropriate OOP concepts wherever applicable

```
internal class EBCalculator
   {

       private double pre_read;
       private double curr_read;

       public double Pre_read { get => pre_read; set => pre_read = value; }
       public double Curr_read { get => curr_read; set => curr_read = value; }

       public const double unit_charge = 10;

       public void EBReading(double pre_read,double curr_read)
       {
         Pre_read= pre_read;
         Curr_read= curr_read;
       }

       public void EnterReading(double pre_read,double curr_read)
       {
          EBReading(pre_read,curr_read);
       }

       public void CalculateBill(double pre_read, double curr_read)
       {
          if (pre_read == null || curr_read == null)
          {
             throw new InvalidOperationException("Electricity readings have not been entered.");
          }
          double units = curr_read - pre_read;
          double bill = units* unit_charge;
          Console.WriteLine("Total Amount You have to Pay : "+bill);

       }

   }
//Driver Code

class Program
{
   public static void Main(string[] args)
   {
           EBCalculator calculator= new EBCalculator();
```

```
        Console.WriteLine("Enter Previous Reading");
        double pre_read = Convert.ToDouble(Console.ReadLine());
        Console.WriteLine("Enter current Reading");
        double curr_read = Convert.ToDouble(Console.ReadLine());

        calculator.EnterReading(pre_read, curr_read);
        calculator.CalculateBill(pre_read, curr_read);



    }
```

```
Enter Previous Reading
0
Enter current Reading
10
Total Amount You have to Pay : 100

D:\C#\Programs\Assessment\Assessment\bin\Debug\net6.0\Assessment.exe (process 8584) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

b. Implement a Temperature Converter class with suitable methods to display the converted temperature

```
internal class TemperatureConverter
    {
        public static double CelsiusToFahrenheit(double celsius)
        {
            return (celsius * 9 / 5) + 32;
        }

        public static double CelsiusToKelvin(double celsius)
        {
            return celsius + 273.15;
        }

        public static double FahrenheitToCelsius(double fahrenheit)
        {
            return (fahrenheit - 32) * 5 / 9;
        }

        public static double FahrenheitToKelvin(double fahrenheit)
        {
            return (fahrenheit + 459.67) * 5 / 9;
        }

        public static double KelvinToCelsius(double kelvin)
        {
            return kelvin - 273.15;
        }

        public static double KelvinToFahrenheit(double kelvin)
        {
            return (kelvin * 9 / 5) - 459.67;
        }
    }

    //Driver Code

class Program
{
```

```
    public static void Main(string[] args)
    {

            Console.WriteLine("Enter Celsius");
        double celsius = Convert.ToDouble( Console.ReadLine()) ;
        double fahrenheit = TemperatureConverter.CelsiusToFahrenheit(celsius);
        double kelvin = TemperatureConverter.CelsiusToKelvin(celsius);
        Console.WriteLine(celsius + "°C = " + fahrenheit + "°F = " + kelvin + "K");

        Console.WriteLine("Enter Fahrenheit");
        fahrenheit = Convert.ToDouble(Console.ReadLine());
        celsius = TemperatureConverter.FahrenheitToCelsius(fahrenheit);
        kelvin = TemperatureConverter.FahrenheitToKelvin(fahrenheit);
        Console.WriteLine(fahrenheit + "°F = " + celsius + "°C = " + kelvin + "K");

        Console.WriteLine("Enter Kelvin");
        kelvin = Convert.ToDouble(Console.ReadLine());
        celsius = TemperatureConverter.KelvinToCelsius(kelvin);
        fahrenheit = TemperatureConverter.KelvinToFahrenheit(kelvin);
        Console.WriteLine(kelvin + "K = " + celsius + "°C = " + fahrenheit + "°F");

        TemperatureConverter temperatureConverter = new TemperatureConverter();
    }

}
```

```
Enter Celsius
10
10°C = 50°F = 283.15K
Enter Fahrenheit
10
10°F = -12.222222222222221°C = 260.92777777777775K
Enter Kelvin
10
10K = -263.15°C = -441.67°F

D:\C#\Programs\Assessment\Assessment\bin\Debug\net6.0\Assessment.exe (process 21384) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

3. Create the files. (allyears.txt, leapyears.txt, nonleapyears.txt) with appropriate data and do the following actions.

i. Read the allyears.txt file
ii. Write the leapyears.txt with leap years based on the input from years.txt
iii. Write the nonleapyears.txt with non-leap years based on the input from years.txt

i)

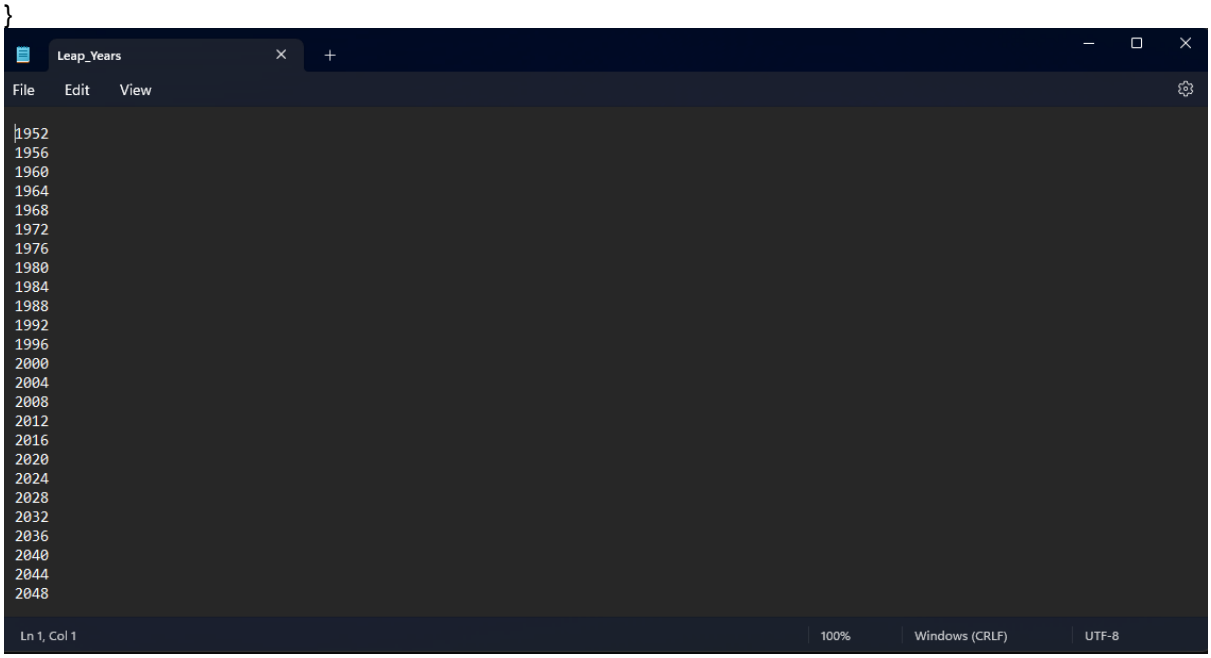
```
internal class FileCreate
{
    public void file()
    {
        //CREATING ALL YEAR FILE WITH YEARS FROM 1950 TO 2050

        FileInfo fi = new FileInfo("D:\C#\Programs\\All_Years.txt");
        using StreamWriter stwr = fi.CreateText();
        Console.WriteLine("File has been created");
        for (int i = 1950; i < 2050; i++)
        {
            stwr.WriteLine(i);
        }
    }
    public void Readfile()
    {
```

```
        string file_details = " D:\C#\Programs\\All_Years.txt";
        try
        {
            StreamReader sr = new StreamReader(file_details);
            string text = File.ReadAllText(file_details);
            Console.WriteLine(text);
        }
        catch (FileNotFoundException e)
        {
            Console.WriteLine(e.Message);
        }


    }
}
//Driver Code
internal class Program
{
    public static void Main(string[] args)
    {
        FileCreate filecrt = new FileCreate();
        filecrt.Readfile();
    }
}
}
```

ii)

```
internal class FileCreate
{
    public void file()
    {
        //CREATING ALL YEAR FILE WITH YEARS FROM 1950 TO 2050

        FileInfo fi = new FileInfo("D:\C#\Programs\\Leap_Years.txt");
        using StreamWriter stwr = fi.CreateText();
        Console.WriteLine("File has been created");

}
    public void Writefile()
    {
```

```
        FileStream fs = new FileStream("D:\C#\Programs\\All_Years.txt", FileMode.Open, FileAccess.Read);

        StreamReader sr = new StreamReader(fs);

        fs = new FileStream("D:\C#\Programs\\Leap_Years.txt", FileMode.Create, FileAccess.Write);
        StreamWriter swLeapYear = new StreamWriter(fs);
        string line;
        while ((line = sr.ReadLine()) != null)
        {
            int year = int.Parse(line);

            if (year % 4 == 0)
            {
                swLeapYear.WriteLine(year);
            }
        }
        swLeapYear.Close();

    }
}
//Driver Code

internal class Program
{
    public static void Main(string[] args)
    {
        FileCreate filecrt = new FileCreate();
        filecrt.Writefile();
    }
}
}
```



iii)

```
internal class FileCreate
{
    public void file()
    {
        //CREATING ALL YEAR FILE WITH YEARS FROM 1950 TO 2050

        FileInfo fi = new FileInfo("D:\\C#.NET PROGRAMS\\Non_Leap_Years.txt");
        using StreamWriter stwr = fi.CreateText();
        Console.WriteLine("File has been created");
```

```
    }


    public void Writefile()
    {
        FileStream fs = new FileStream("D:\C#\Programs\\Non_Leap_Years.txt", FileMode.Create,
FileAccess.Write);
        StreamWriter swNonLeapYear = new StreamWriter(fs);
        fs = new FileStream("D:\C#\Programs\\All_Years.txt", FileMode.Open, FileAccess.Read);
        StreamReader sr = new StreamReader(fs);
        string line;
        while ((line = sr.ReadLine()) != null)
        {
            int year = int.Parse(line);

            if (year % 4 != 0)
            {
                swNonLeapYear.WriteLine(year);
            }
        }
        swNonLeapYear.Close();
        sr.Close();
        fs.Close();
    }
}
//Driver Code
internal class Program
{
    public static void Main(string[] args)
    {
        FileCreate filecrt = new FileCreate();
        filecrt.Writefile();
    }
}
}
```