**ASSESSMENT 12-04-2023**

**1. Answer the problems**

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

namespace Exceptionhandling

{

internal class Program

{

static void Main(string[] args)

{

int age;

age = Convert.ToInt32(Console.ReadLine());

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Exceptionhandling

{

internal class Person

{

private int age;

public Person(int age)

{

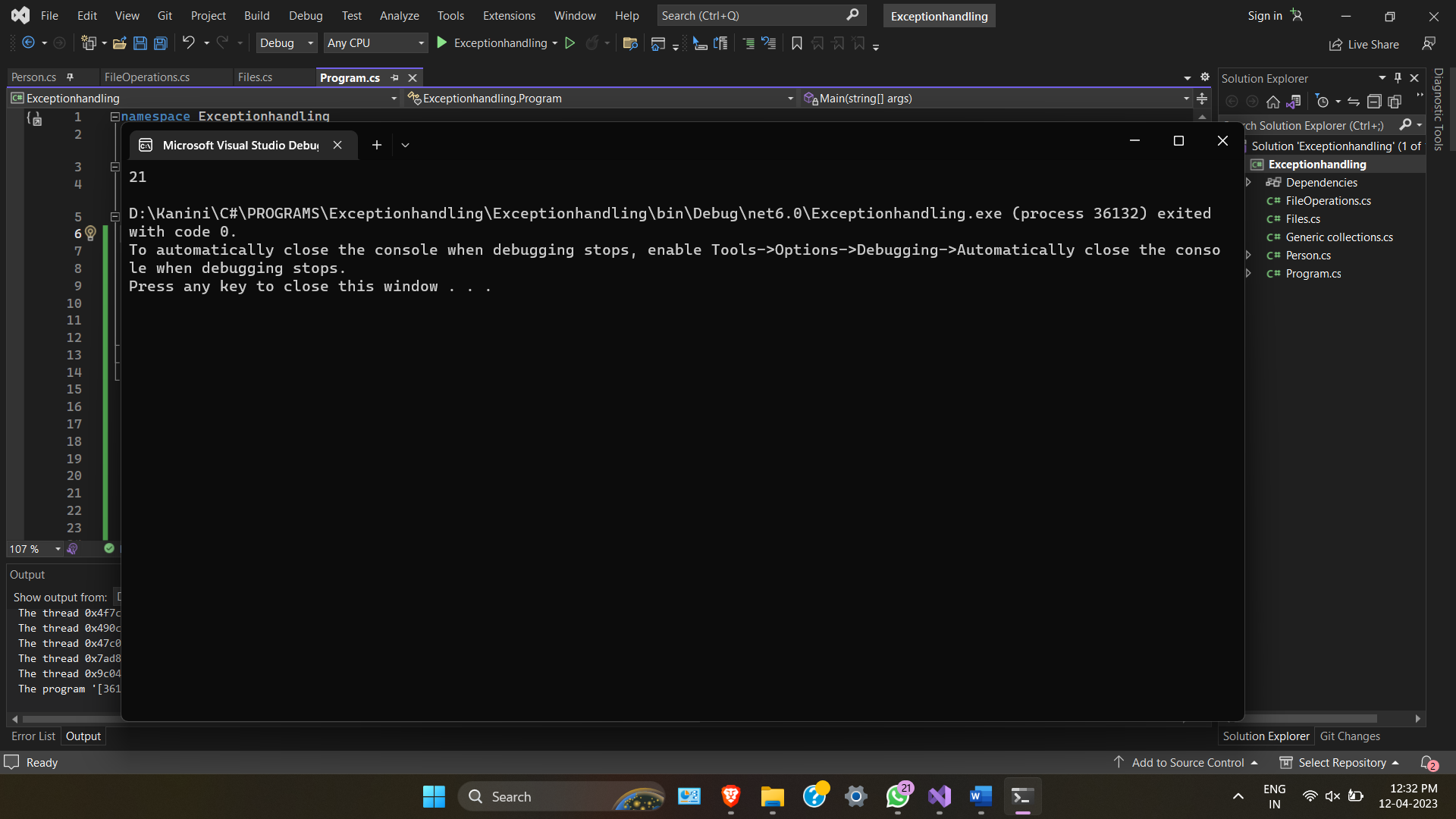
this.Age= age;

}

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

}

}



**b. Implement a user defined exception to handle Age eligibility to vote (18-120 years eligible. Others not eligible)**

namespace Exceptionhandling

{

internal class Program

{

static void Main(string[] args)

{

int age;

age = Convert.ToInt32(Console.ReadLine());

Person po = new Person(age);

po.AgeEligiblity();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Exceptionhandling

{

internal class Person

{

private int age;

public Person(int age)

{

this.Age= age;

}

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

public void AgeEligiblity()

{

if (Age < 18)

{

throw new ArithmeticException("Access denied - You are Not Eligible to Vote");

}

else if (Age > 120)

{

Console.WriteLine("Access denied - Enter the correct Age !");

}

else

{

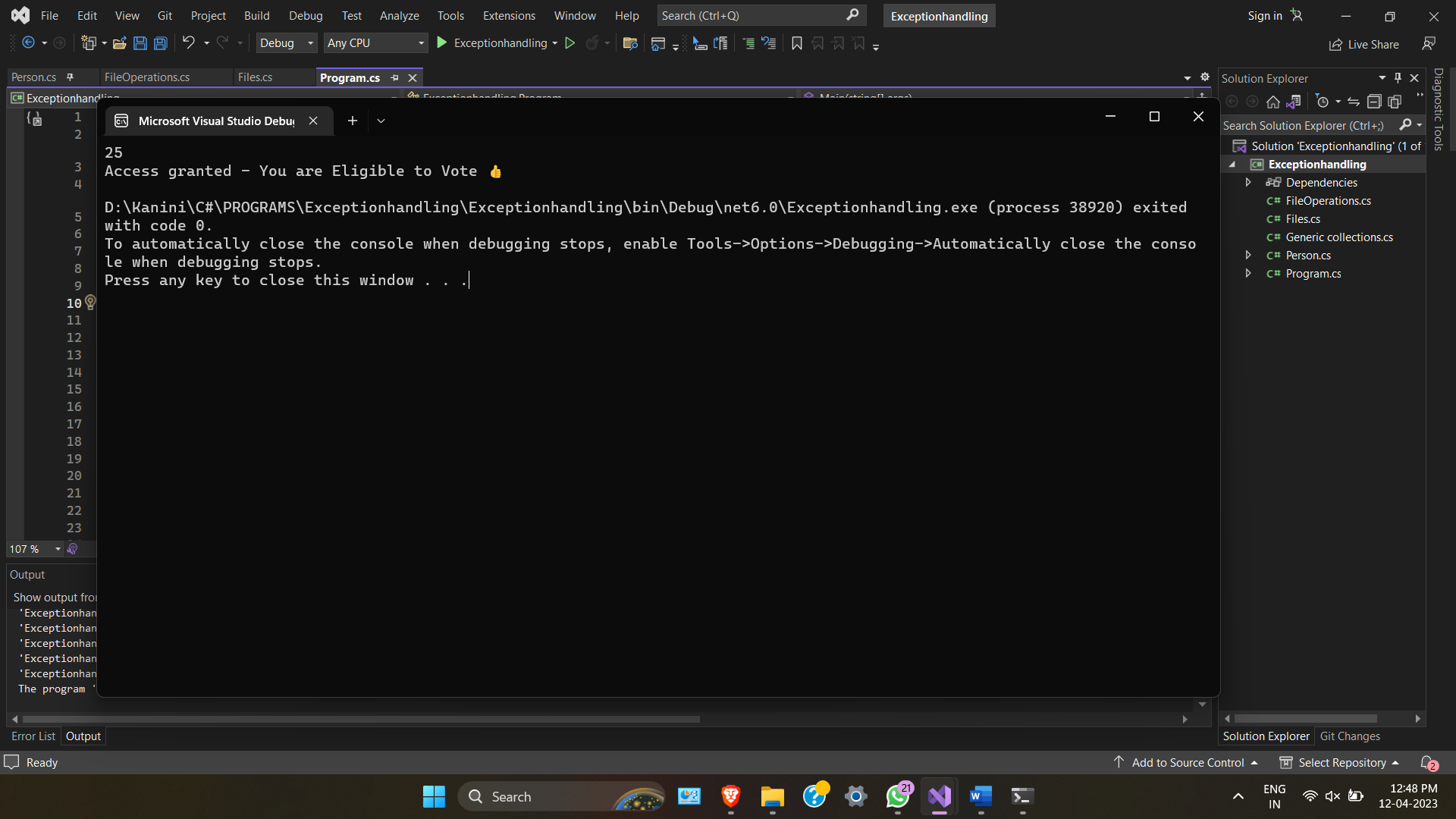
Console.WriteLine("Access granted - You are Eligible to Vote 👍");

}

}

}

}



**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.**

namespace Exceptionhandling

{

internal class Program

{

static void Main(string[] args)

{

CurrencyConverter convert = new CurrencyConverter();

convert.CurrencyConvert();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Exceptionhandling

{

internal class CurrencyConverter

{

public void CurrencyConvert()

{

int choice;

Console.WriteLine("Enter your Choice :\n 1- Rupee to Dollar \n 2 - Rupee to Euro \n 3 - Rupee to Emirati Dirham \n 4 - Rupee to Canadian dollar \n 5 - Rupee to New Zealand Dollar \n \r\n");

choice = int.Parse(Console.ReadLine());

switch (choice)

{

case 1:

Double dollar, rupee;

Console.WriteLine("Enter the Rupee Amount :");

rupee = Double.Parse(Console.ReadLine());

dollar = 0.012 \* rupee;

Console.WriteLine("{0} Dollar Equals {1} Rupees", dollar, rupee);

break;

case 2:

Double Euro, rupe;

Console.WriteLine("Enter the Rupee Amount :");

rupe = Double.Parse(Console.ReadLine());

Euro = rupe \* 0.011;

Console.WriteLine("{0} Euro Equals {1} Rupees", Euro, rupe);

break;

case 3:

Double AED, rup;

Console.WriteLine("Enter the Rupee Amount :");

rup = Double.Parse(Console.ReadLine());

AED = rup \* 0.045;

Console.WriteLine("{0} AED Equals {1} Rupees",

AED, rup);

break;

case 4:

Double CAD, ru;

Console.WriteLine("Enter the Rupee Amount :");

ru = Double.Parse(Console.ReadLine());

CAD = ru \* 0.016;

Console.WriteLine("{0} CAD {1} Rupees",

CAD, ru);

break;

case 5:

Double NZD, rue;

Console.WriteLine("Enter the NZD Amount :");

rue = Double.Parse(Console.ReadLine());

NZD = rue \* 0.020;

Console.WriteLine("{0} NZD {1} Rupees",

NZD, rue);

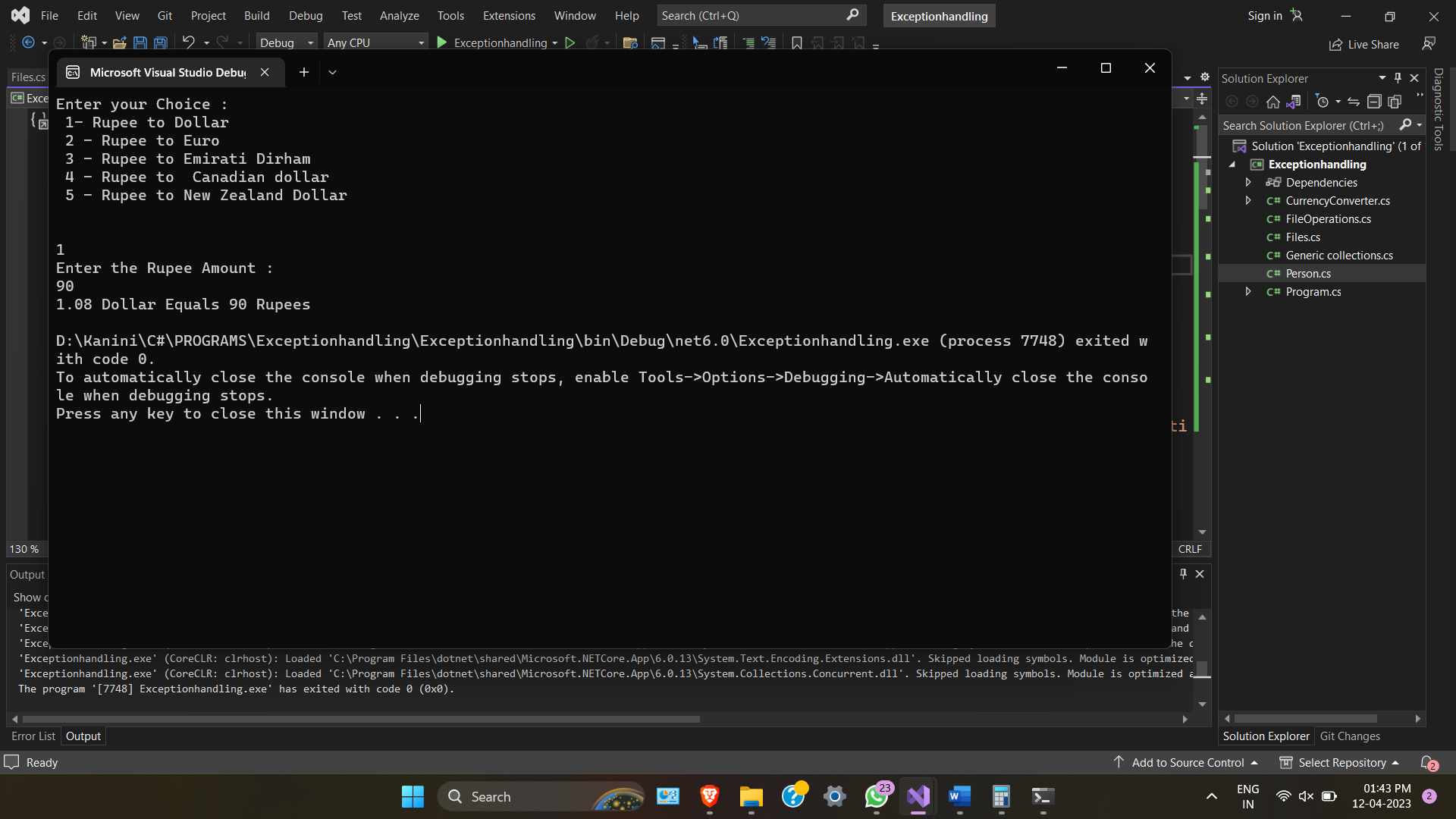
break;

}

}

}

}



**2. Answer the problems**

**a. Implement an EB Calculator class with suitable methods to calculate the bill. The class**

**should hold the methods EnterReading() &amp; CalculateBill(). Use appropriate OOP concepts wherever applicable**

namespace Exceptionhandling

{

internal class Program

{

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);

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Exceptionhandling

{

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;

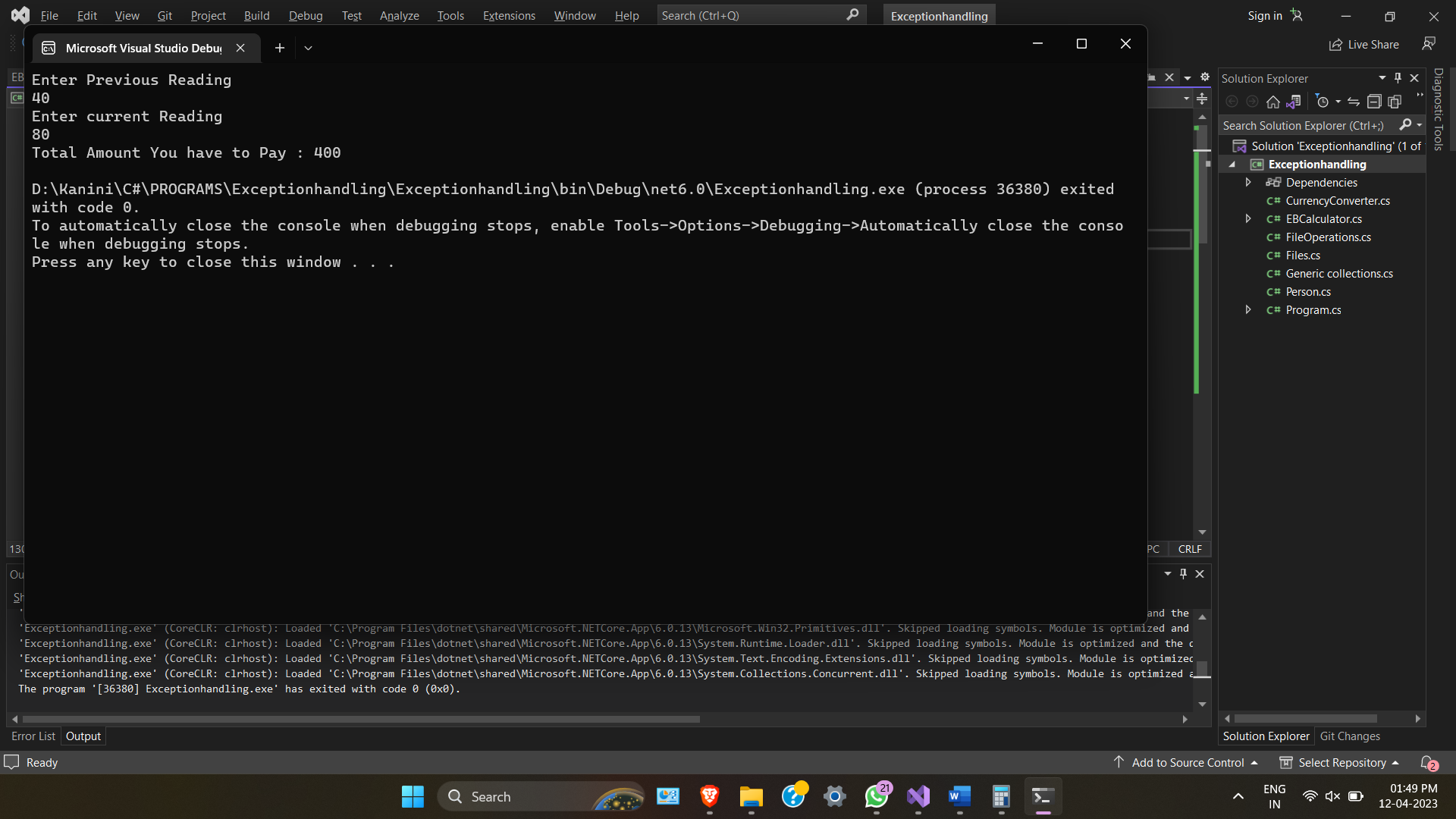
double bill = units \* unit\_charge;

Console.WriteLine("Total Amount You have to Pay : " + bill);

}

}

}



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

namespace Exceptionhandling

{

internal class Program

{

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();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Exceptionhandling

{

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)

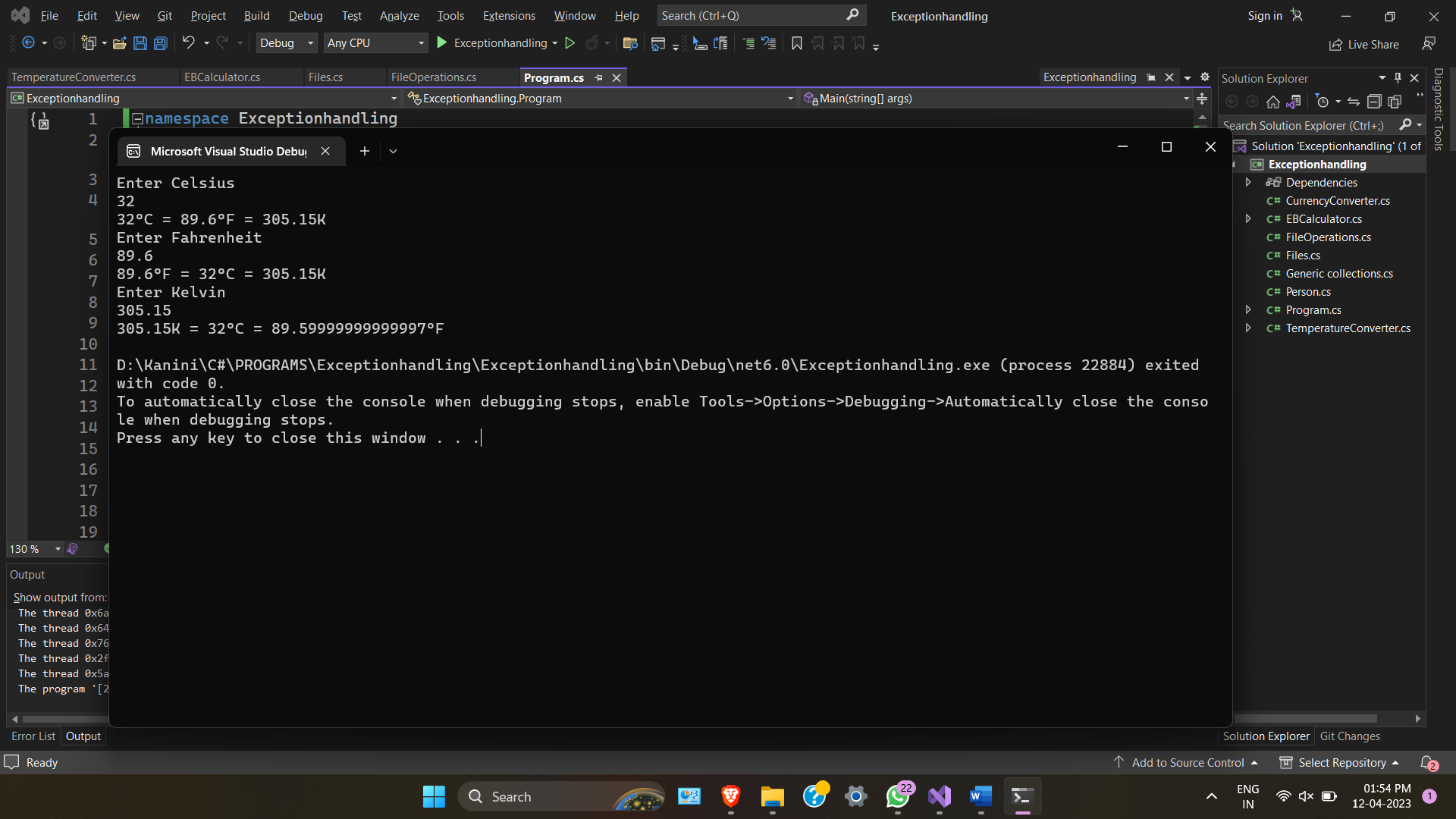
{

return (kelvin \* 9 / 5) - 459.67;

}

}

}



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

**following actions.**

**i. Read the allyears.txt file**

namespace Exceptionhandling

{

internal class Program

{

static void Main(string[] args)

{

Files fo = new Files();

fo.read\_file();

fo.write\_to\_leapyear();

fo.write\_to\_nonleapyear();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Exceptionhandling

{

internal class Files

{

string file\_path = "D:\\Kanini\\C#\\allyears.txt";

string leap\_year\_file\_path = "D:\\Kanini\\C#\\leapyears.txt";

string non\_leapyear\_file\_path = "D:\\Kanini\\C#\\nonleapyears.txt";

public string File\_path { get => file\_path; set => file\_path = value; }

public string Leap\_year\_file\_path { get => leap\_year\_file\_path; set => leap\_year\_file\_path = value; }

public string Non\_leapyear\_file\_path { get => non\_leapyear\_file\_path; set => non\_leapyear\_file\_path = value; }

public void read\_file()

{

try

{

StreamReader sr = new StreamReader(this.File\_path);

string text = File.ReadAllText(this.File\_path);

Console.WriteLine(text);

sr.Close();

}

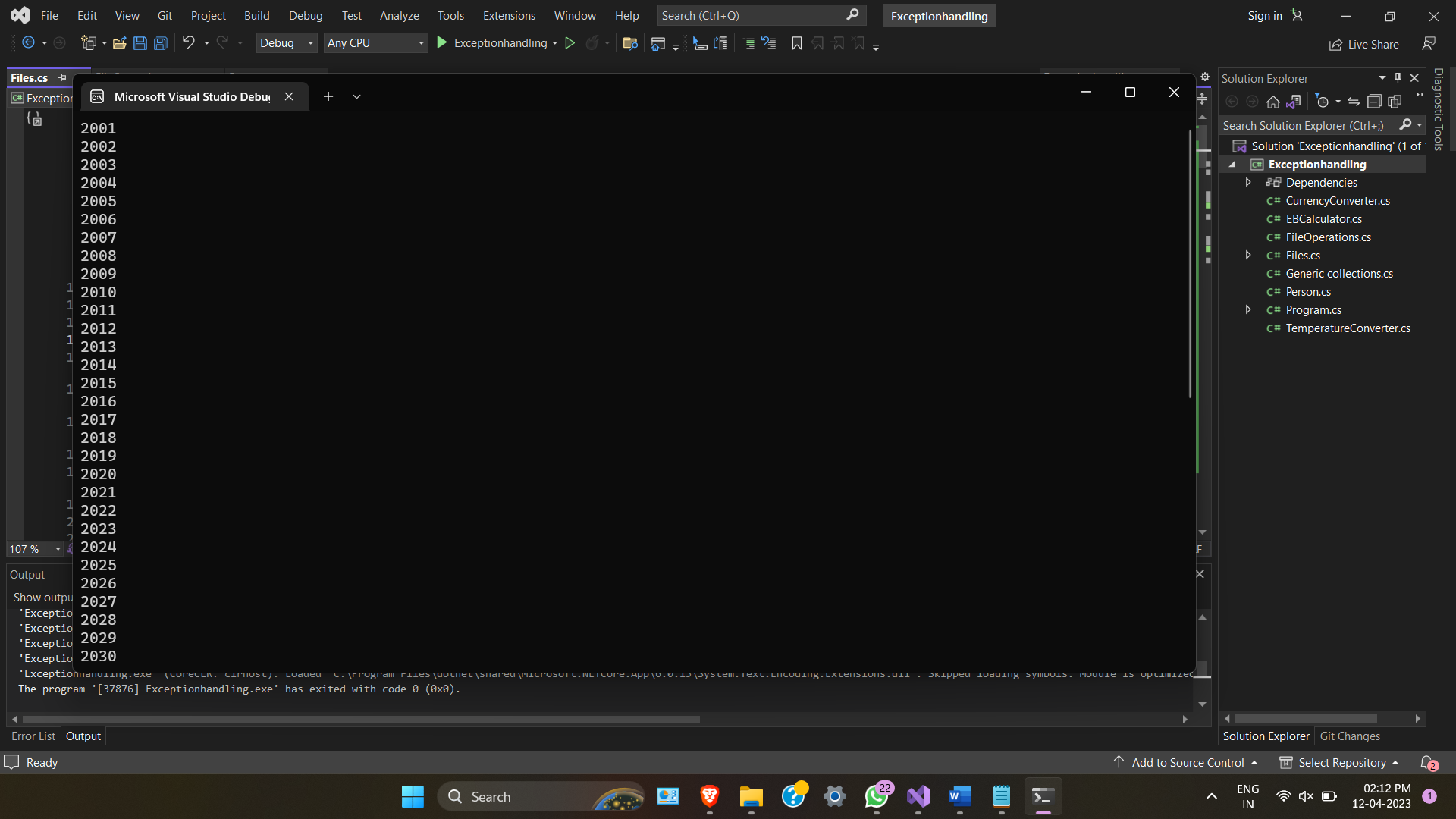
catch (FileNotFoundException e)

{

Console.WriteLine(e);

}

}



**ii. Write the leapyears.txt with leap years based on the input from years.txt**

public void write\_to\_leapyear()

{

using (StreamReader ReaderObject = new StreamReader(File\_path))

{

string line;

while ((line = ReaderObject.ReadLine()) != null)

{

if (DateTime.IsLeapYear(Convert.ToInt32(line)))

{

//File.AppendAllLines(this.Leap\_year\_file\_path, line);

using (StreamWriter leap\_year = File.AppendText(this.Leap\_year\_file\_path))

{

leap\_year.WriteLine(line);

}

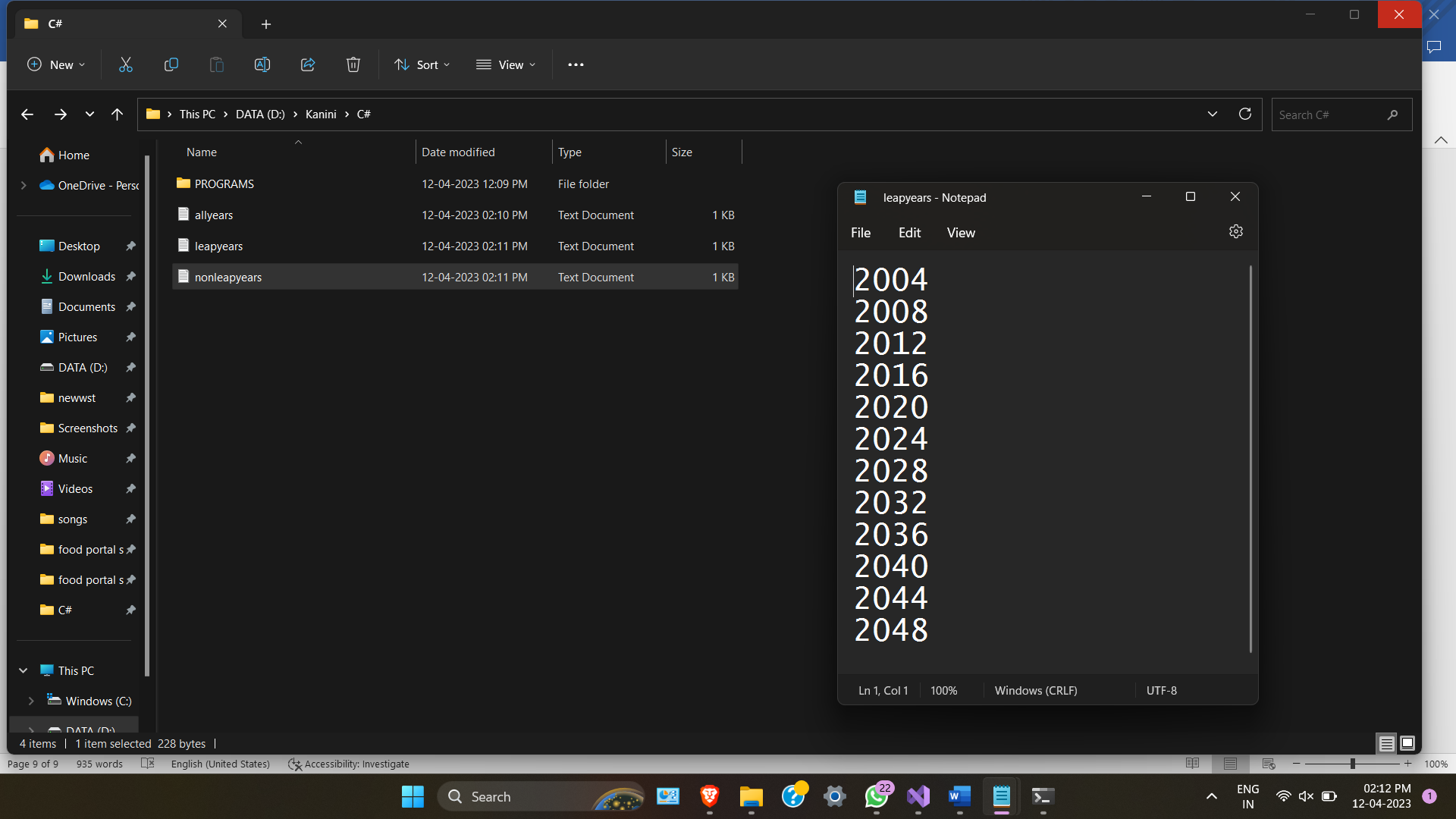
}

}

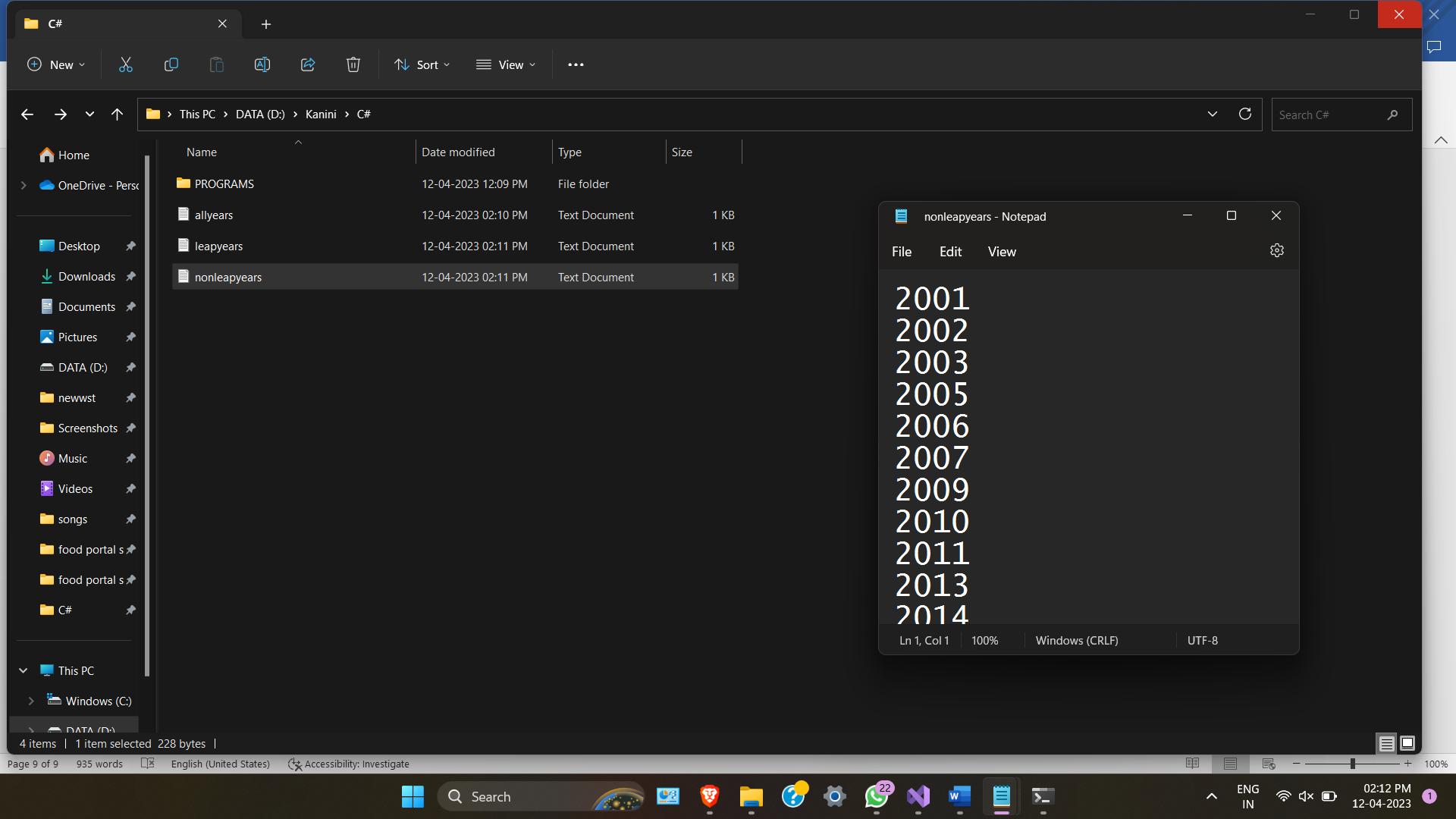
}

Console.WriteLine("Done");

}



**iii. Write the nonleapyears.txt with non-leap years based on the input from years.txt**



public void write\_to\_nonleapyear()

{

using (StreamReader ReaderObject1 = new StreamReader(File\_path))

{

string line;

while ((line = ReaderObject1.ReadLine()) != null)

{

if (!(DateTime.IsLeapYear(Convert.ToInt32(line))))

{

using (StreamWriter non\_leap\_year = File.AppendText(this.Non\_leapyear\_file\_path))

{

non\_leap\_year.WriteLine(line);

}

}

}

}

Console.WriteLine("Done");

}