

1. b.

using BasicPrograms;

using System.Data.SqlClient;

using System.Linq.Expressions;

class Program

{

public static void Main(string[] args)

{

Console.Write("Enter Age: ");

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

Person person = new Person();

Console.WriteLine("Is person eligible for voting? " + Person.CheckAge(age));

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace BasicPrograms

{

internal class Person

{

private int age;

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

public static bool CheckAge(int age)

{

if (age < 18 ||age > 120)

{

throw new ArithmeticException(" Not Eligible for voting.");

}

else

{

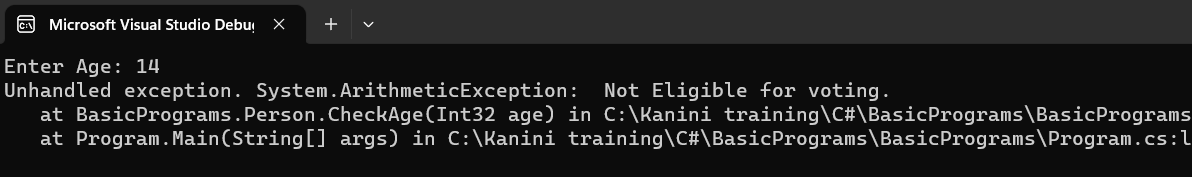
return true;

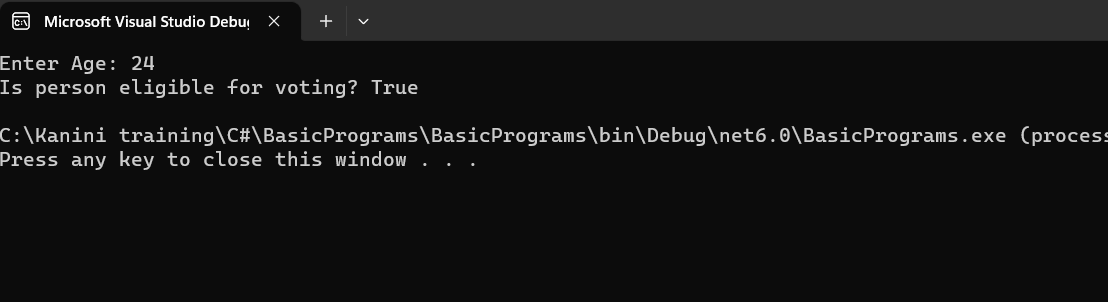
}

}

}

}





c.

using BasicPrograms;

using System.Data.SqlClient;

using System.Linq.Expressions;

class Program

{

public static void Main(string[] args)

{

string fromCurrency="", toCurrency="";

var converter = new CurrencyConverter();

while (true)

{

Console.WriteLine("Choose a conversion type :");

Console.WriteLine("1. USD to INR");

Console.WriteLine("2. INR to USD");

Console.WriteLine("3. USD to EUR");

Console.WriteLine("4. EUR to USD");

Console.WriteLine("5. INR to EUR");

Console.WriteLine("6. EUR to INR");

Console.WriteLine("0. Exit");

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

if (choice == 0)

{

break;

}

Console.WriteLine("Enter amount: ");

double amount = double.Parse(Console.ReadLine());

switch (choice)

{

case 1:

fromCurrency = "USD";

toCurrency = "INR";

break;

case 2: fromCurrency = "INR";

toCurrency = "USD";

break;

case 3: fromCurrency = "USD";

toCurrency = "EUR";

break;

case 4: fromCurrency = "EUR";

toCurrency = "USD";

break;

case 5: fromCurrency = "INR";

toCurrency = "EUR";

break;

case 6: fromCurrency = "EUR";

toCurrency = "INR";

break;

default:

Console.WriteLine("Invalid choice");

break;

}

double result = converter.Convert(fromCurrency, toCurrency, amount);

Console.WriteLine($"{amount} {fromCurrency} = {result} {toCurrency}");

Console.WriteLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace BasicPrograms

{

internal class CurrencyConverter

{

public double Convert(string fromCurrency, string toCurrency, double amount)

{

double rate = GetRate(fromCurrency, toCurrency);

return amount \* rate;

}

private static double GetRate(String fromCurrency, string toCurrency)

{

switch (fromCurrency + toCurrency)

{

case "USDINR":

return 74.35;

case "INRUSD":

return 0.013;

case "USDEUR":

return 0.84;

case "EURUSD":

return 1.19;

case "INREUR":

return 0.008;

case "EURINR":

return 125.06;

default:

Console.WriteLine("Invalid currency pair: {from}/{to}");

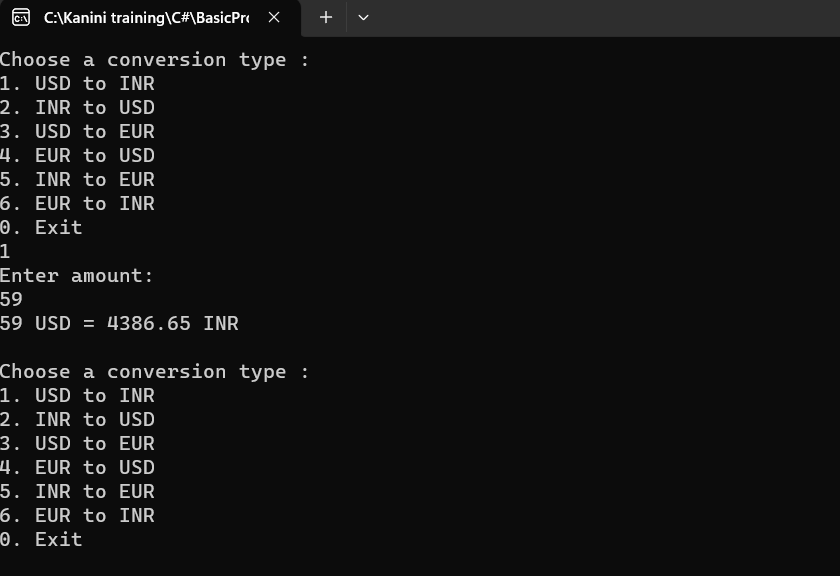
return 0;

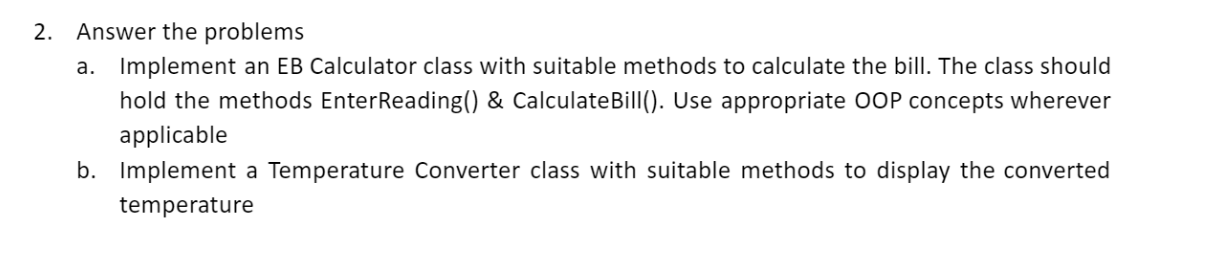
}

}

}

}





a.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace BasicPrograms{

public class EBcalculator

{

private int currentUnit, previousUnit;

private double bill;

public void EnterReading()

{

Console.WriteLine("Enter the previous Unit : ");

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

Console.WriteLine("Enter the current Unit : ");

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

}

public void CalculateBill()

{

int Consumed = currentUnit - previousUnit;

bill = Consumed \* 12.50;

}

public void BillAmount()

{

Console.WriteLine("Total amount is : {0}", bill);

}

}

}

b.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace BasicPrograms

{

internal class TemperatureConverter

{

private double temperature;

public void ReadTemp()

{

Console.WriteLine("Enter the temperature: ");

temperature = double.Parse(Console.ReadLine());

}

public void ConvertTemp()

{

temperature = (temperature \* 1.8) + 32;

}

public void DisplayTemp()

{

Console.WriteLine("The converted temperature is : {0}", temperature);

}

}

}

using BasicPrograms;

using System;

class Program

{

public static void Main(string[] args)

{

EBcalculator calculation = new EBcalculator();

calculation.EnterReading();

calculation.CalculateBill();

calculation.BillAmount();

TemperatureConverter Conversion = new TemperatureConverter();

conversion.ReadTemp();

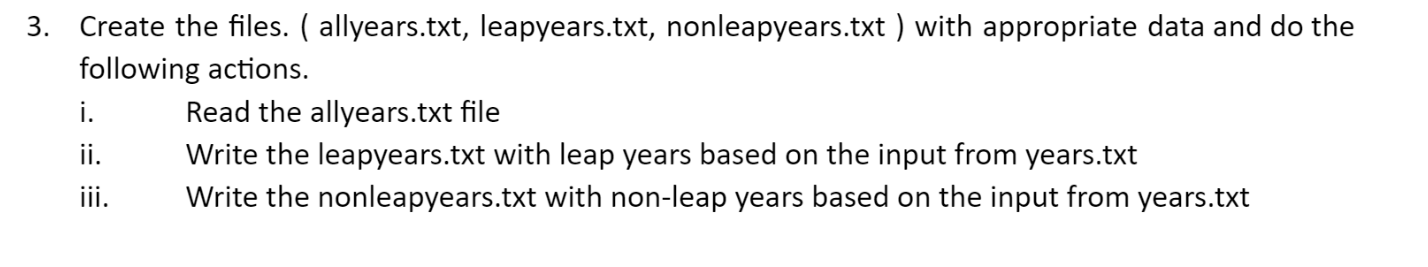
conversion.ConvertTemp();

conversion.DisplayTemp();

Console.ReadLine();

}

}



i)

using System;

using System.IO;

using BasicPrograms;

class Program

{

static void Main(string[] args)

{

string filePath = @"C:\Kanini training\C# \allyears.txt";

string filePathLeapYear = @" C:\Kanini training\C# \ leapyears.txt";

string filePathNonLeapYear = @" C:\Kanini training\C# \nonleapyears.txt";

FileStream fs = new FileStream(filePath, FileMode.Create, FileAccess.Write);

StreamWriter sw = new StreamWriter(fs);

sw.WriteLine("1930");

sw.WriteLine("1933");

sw.WriteLine("1940");

sw.WriteLine("1944");

sw.WriteLine("1948");

sw.WriteLine("1952");

sw.WriteLine("1955");

sw.WriteLine("1960");

sw.WriteLine("1964");

sw.WriteLine("1968");

sw.WriteLine("1972");

sw.WriteLine("1976");

sw.WriteLine("1981");

sw.WriteLine("1984");

sw.WriteLine("1988");

sw.WriteLine("1992");

sw.WriteLine("1996");

sw.WriteLine("2000");

sw.WriteLine("2001");

sw.WriteLine("2005");

sw.WriteLine("2008");

sw.WriteLine("2009");

sw.WriteLine("2016");

sw.WriteLine("2018");

sw.WriteLine("2022");

sw.WriteLine("2023");

sw.Close();

fs.Close();

fs = new FileStream(filePath, FileMode.Open, FileAccess.Read);

StreamReader sr = new StreamReader(fs);

fs = new FileStream(filePathLeapYear, FileMode.Create, FileAccess.Write);

StreamWriter LeapYear = new StreamWriter(fs);

string line;

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

{

int year = int.Parse(line);

if (year % 4 == 0)

{

LeapYear.WriteLine(year);

}

}

LeapYear.Close();

fs = new FileStream(filePathNonLeapYear, FileMode.Create, FileAccess.Write);

StreamWriter swNonLeapYear = new StreamWriter(fs);

fs = new FileStream(filePath, FileMode.Open, FileAccess.Read);

sr = new StreamReader(fs);

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

{

int year = int.Parse(line);

if (year % 4 != 0)

{

swNonLeapYear.WriteLine(year);

}

}

swNonLeapYear.Close();

sr.Close();

fs.Close();

}

}