**C# ASSESSMENT**

1. internal class problem

{

public static void Main(string[] args)

{

Console.WriteLine("Choose Your Choice");

Console.WriteLine("1. Character Pattern");

Console.WriteLine("2. Numeric Pattern");

int choice;

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

switch (choice)

{

case 1:

int num1;

Console.WriteLine("How many rows to be printed ");

num1= Convert.ToInt32(Console.ReadLine());

CharPattern ch = new CharPattern(num1);

ch.CPattern();

break;

case 2:

int num2;

Console.WriteLine("How many rows to be printed ");

num2= Convert.ToInt32(Console.ReadLine());

NumericPattern n = new NumericPattern(5);

n.NPattern();

break;

default: Console.WriteLine("Enter Correct Choice");

break;

}

}

}

//Character Pattern

internal class CharPattern

{

private int i, row,column;

internal CharPattern(int num)

{

this.I = num;

}

public int I { get => i; set => i = value; }

public int Row { get => row; set => row = value; }

public int Column { get => column; set => column = value; }

public void CPattern()

{

for(row=0; row<this.I; row++)

{

for(column=0; column<row; column++)

{

Console.Write("\* ");

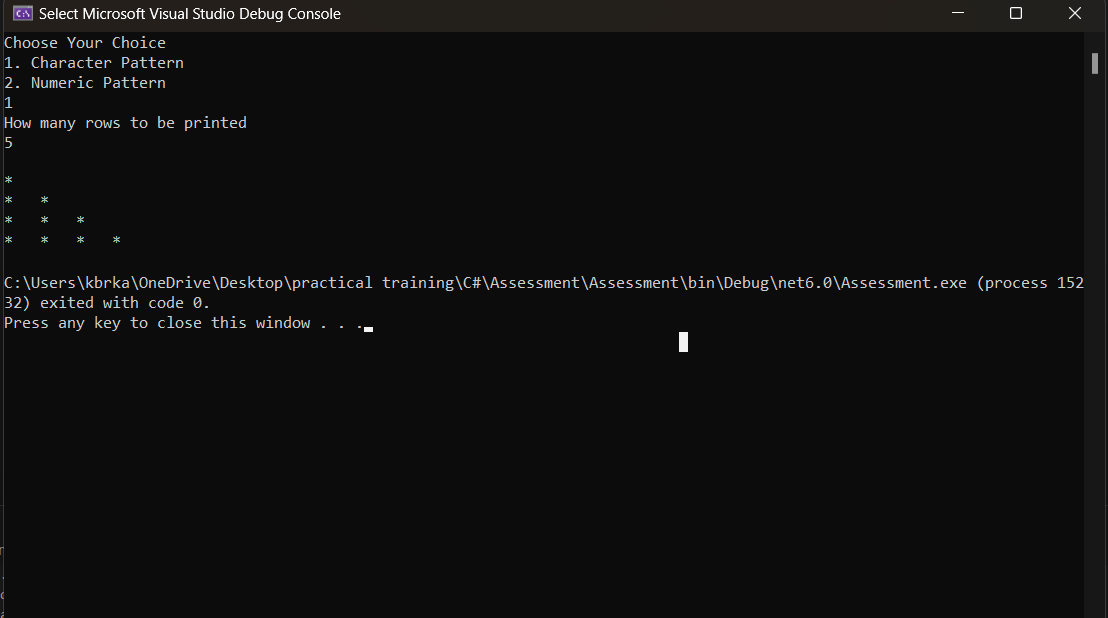
}

Console.WriteLine();

}

}

}



//Numeric Pattern

internal class NumericPattern

{

int i, row, column;

public NumericPattern(int num)

{

this.I = num;

}

public int I { get => i; set => i = value; }

public int Row { get => row; set => row = value; }

public int Column { get => column; set => column = value; }

public void NPattern()

{

int number = 1;

for (int i = 0; i < I ; i++)

{

for (int space = I - 1; space > i; space--)

{

Console.Write(" ");

}

for (int j = 0; j <= i; j++)

{

Console.Write(number++ + " ");

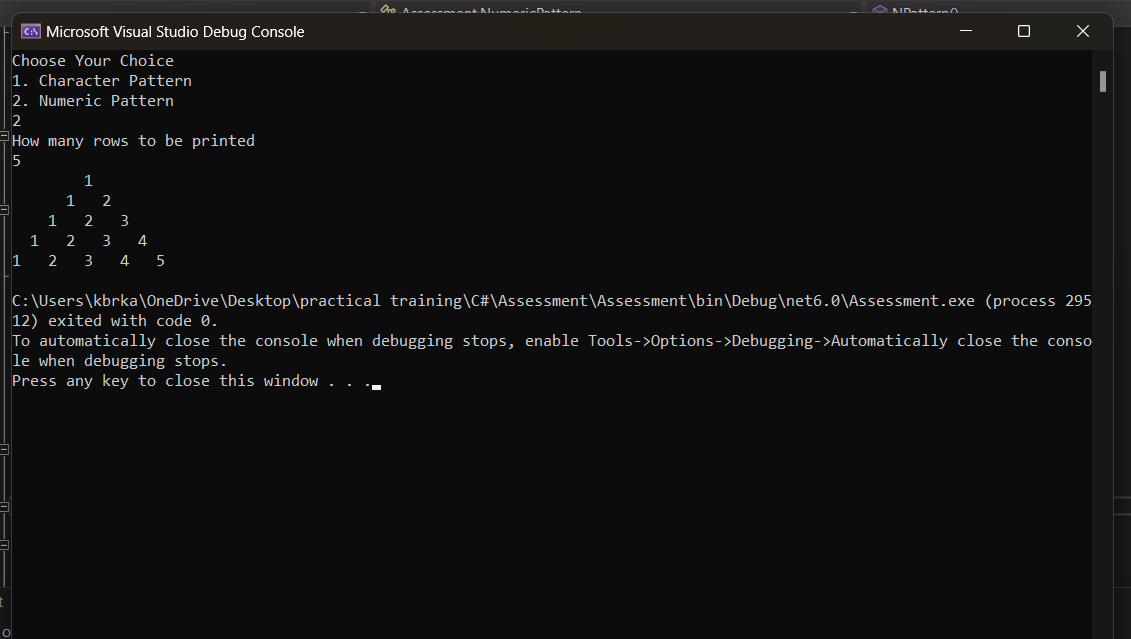
}

Console.WriteLine();

number = 1;

}

}



1. a)int n1, n2, new\_choice;

Console.WriteLine("Enter two Numbers");

n1 = Convert.ToInt32(Console.ReadLine());

n2 = Convert.ToInt32(Console.ReadLine());

Calculator cal = new Calculator(n1, n2);

Console.WriteLine("1. Addition");

Console.WriteLine("2. Subtraction");

Console.WriteLine("3. Multiplication");

Console.WriteLine("4. Division");

Console.WriteLine("5. Modulus");

new\_choice = Convert.ToInt32(Console.ReadLine());

switch (new\_choice)

{

case 1:

Console.WriteLine(cal.add());

break;

case 2:

Console.WriteLine(cal.sub());

break;

case 3:

Console.WriteLine(cal.mul());

break;

case 4:

Console.WriteLine(cal.div());

break;

case 5:

Console.WriteLine(cal.mod());

break;

default:

Console.WriteLine("Enter Correct Choice");

break;

}

internal partial class Calculator

{

private int num1,num2;

public Calculator(int n1,int n2)

{

this.Num1 = n1;

this.Num2 = n2;

}

public int Num1 { get => num1; set => num1 = value; }

public int Num2 { get => num2; set => num2 = value; }

}

internal partial class Calculator

{

public int add()

{

int addition;

addition = Num1 + Num2;

return addition;

}

}

internal partial class Calculator

{

public int sub()

{

int subtract;

subtract = Num1 - Num2;

Console.WriteLine("Subtraction of two numbers " + subtract);

return subtract;

}

}

internal partial class Calculator

{

public int mul()

{

int multiplication;

multiplication=Num1 \* Num2;

return multiplication;

}

}

internal partial class Calculator

{

public int div()

{

int division;

division = Num1 / Num2;

return division;

}

}

internal partial class Calculator

{

public int mod()

{

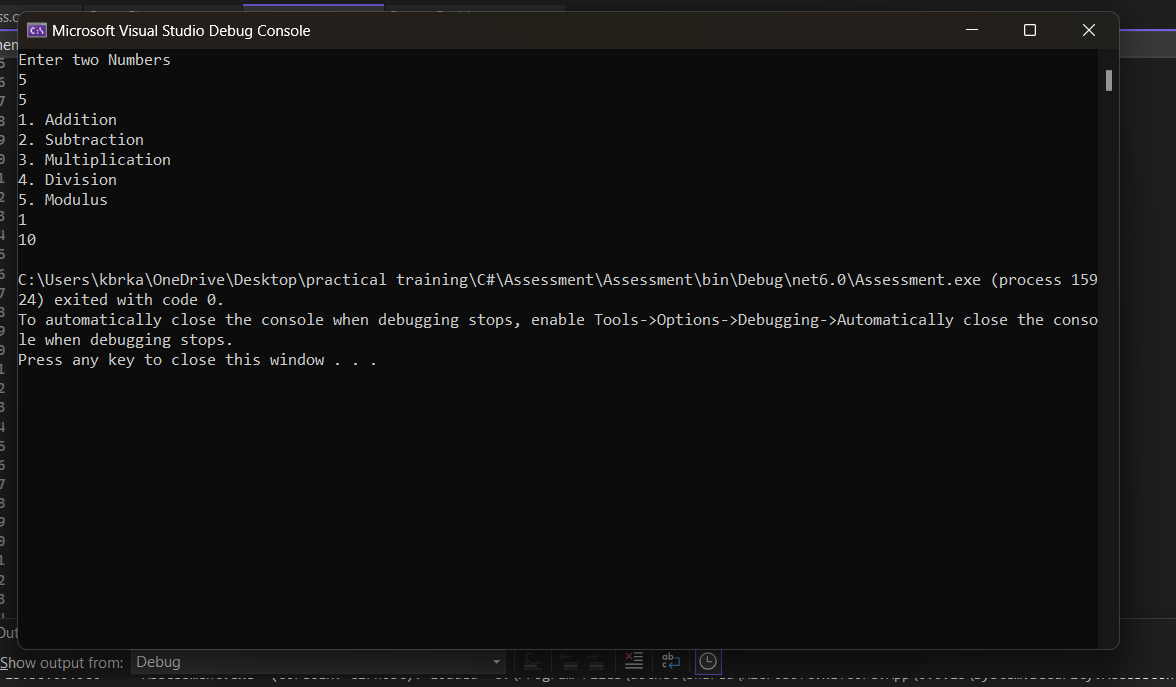
int modulus;

modulus = Num1 % Num2;

return modulus;

}

}



1. b)internal class FileOperations

{

public void CreateFile()

{

FileInfo fi = new FileInfo($@"C:\Users\kbrka\OneDrive\Desktop\practical training\\Employee.txt");

using StreamWriter str = fi.CreateText();

Console.WriteLine("File has been created");

str.WriteLine("Hr Tl Tl Hr Tl ");

Console.WriteLine("Written");

}

public void Readfile()

{

string path = ($@"C:\Users\kbrka\OneDrive\Desktop\practical training\\Employee.txt");

FileStream fileStream = new FileStream(path, FileMode.Open, FileAccess.Read);

StreamReader reader = new StreamReader(fileStream);

reader.BaseStream.Seek(0, SeekOrigin.Begin);

string text = System.IO.File.ReadAllText($@"C:\Users\kbrka\OneDrive\Desktop\practical training\\Employee.txt");

string[] words = text.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries);

int count1 = 0;

int count2 = 0;

for (int i = 0; i < words.Length; i++)

{

if (words[i] == "Hr")

{

count1++;

}

else if (words[i] =="Tl")

{

count2++;

}

}

Console.WriteLine("Count of HR :" + count1);

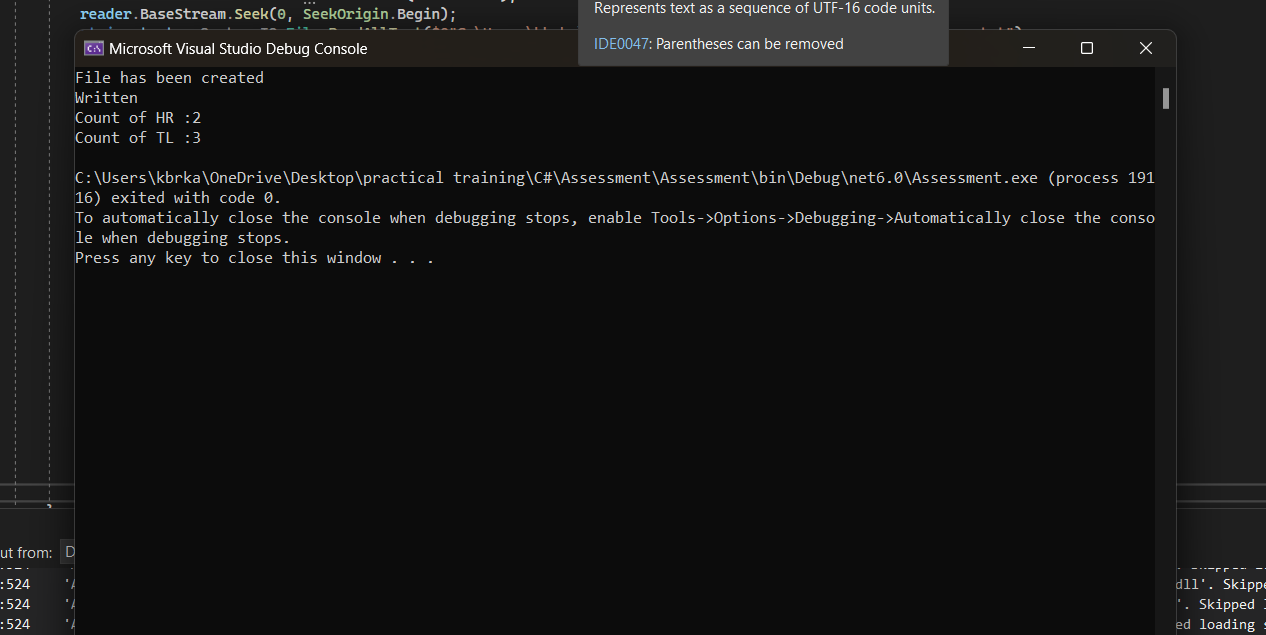
Console.WriteLine("Count of TL :" + count2);

reader.Close();

fileStream.Close();

}

}



1. internal interface SmartPhone

{

public void CallLog();

public void DialLastCall(long s);

}

namespace Assessment

{

internal class derivedclass : SmartPhone

{

public derivedclass() { }

public void CallLog()

{

Stack<long> numbers = new Stack<long>();

numbers.Push(9234567890);

numbers.Push(9545421212);

numbers.Push(8542121542);

numbers.Push(7522454212);

numbers.Push(8542421212);

numbers.Push(8245454512);

numbers.Push(9545421212);

numbers.Push(9545421212);

numbers.Push(9545421212);

numbers.Push(9545421212);

foreach (var number in numbers)

{

Console.WriteLine(number);

}

long n=numbers.Peek();

DialLastCall(n);

}

public void DialLastCall(long s)

{

long s1=s;

Console.WriteLine(s1);

Console.WriteLine();

Console.WriteLine("Dialed to last number");

Console.WriteLine(x);

Console.WriteLine("Call Duration :12:00 minutes");

Console.WriteLine("Call Ended");

}

}

}

derivedclass d = new derivedclass();

double balance = 5;

if (balance <= 0)

{

try

{

Console.WriteLine("Not enough Balance");

}

catch (NotFiniteNumberException ex)

{

Console.WriteLine(ex.Message);

}

}

else

{

d.CallLog();

}

