**Experiment No . 4**

Title: Implementation of Multidimensional and Jagged array.

Name: Vaishnavi Kumar Sutar

Roll No : 54 Batch : T3

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace multidimensionalArr

{

class Program

{

static void Main()

{

int[,] array1 = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

};

int[,] array2 = {

{9, 8, 7},

{6, 5, 4},

{3, 2, 1}

};

int rows = array1.GetLength(0);

int cols = array1.GetLength(1);

int[,] resultArray = new int[rows, cols];

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

{

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

{

resultArray[i, j] = array1[i, j] + array2[i, j];

}

}

Console.WriteLine("Resultant Array after Addition:");

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

{

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

{

Console.Write(resultArray[i, j] + "\t");

}

Console.WriteLine();

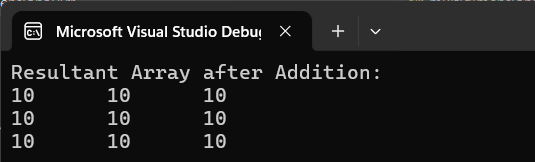
}

}

}

}

**Output :**

****

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace jaggedArray

{

class Program

{

static void Main()

{

// Declaration

int[][] jaggedArray = new int[3][];

// Initialization

jaggedArray[0] = new int[] { 1, 2 };

jaggedArray[1] = new int[] { 3, 4, 5 };

jaggedArray[2] = new int[] { 6, 7, 8, 9 };

// Iterating the elements

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

{

Console.Write("Row " + i + ": ");

foreach (int num in jaggedArray[i])

Console.Write(num + " ");

Console.WriteLine();

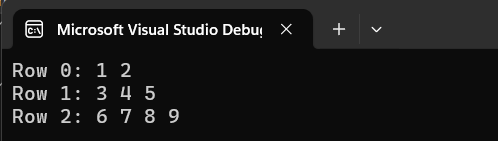
}

}

}

}

**Outout :**

****

**Experiment No . 5**

Title: Implementation of Operator Overloading and String manipulation using String and String builder.

Name: Vaishnavi Kumar Sutar

Roll No : 54 Batch : T3

**//Operator Overloading :**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace OperatorOverloading

{

class Calculator

{

public int number = 0;

public Calculator() { }

public Calculator(int n)

{

number = n;

}

// Overloading of Binary operator ("+")

public static Calculator operator +(Calculator Cal1, Calculator Cal2)

{

Calculator Cal3 = new Calculator(0);

Cal3.number = Cal2.number + Cal1.number;

return Cal3; //returning the result of same type

}

public void Print()

{

Console.WriteLine("{0}", number);

}

}

class TestingInMain

{

static void Main(String[] args)

{

Calculator num1 = new Calculator(510);

Calculator num2 = new Calculator(220);

Calculator num3 = new Calculator();

num3 = num1 + num2;

num1.Print();

num2.Print();

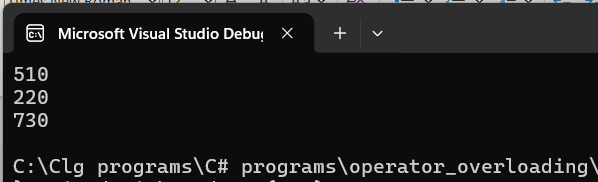
num3.Print();

}

}

}

**Output :**

****

**//String manipulation with StringBuilder :**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace string\_with\_stringBuilder

{

class Program

{

static void Main(string[] args)

{

StringBuilder sb = new StringBuilder("Hello!");

Console.WriteLine(sb);

sb.Insert(5, "World");

Console.WriteLine(sb);

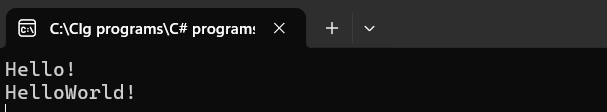
Console.ReadLine();

}

}

}

**Output :**

****