

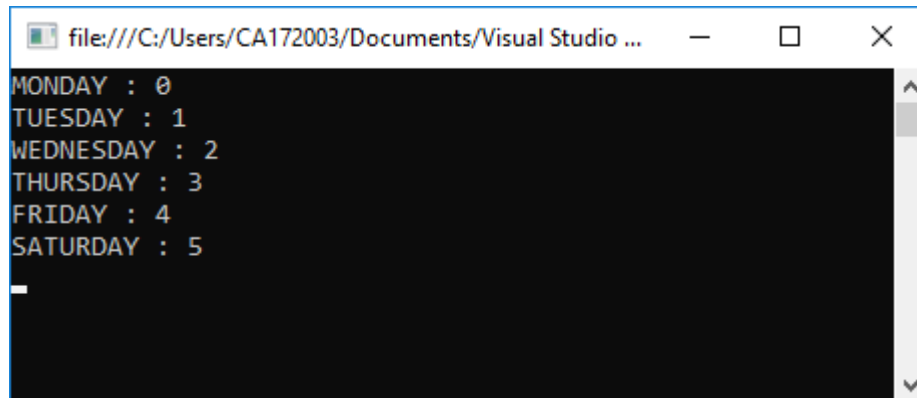
1. Describe the enumerations programming constructs, which provides a human-readable form of a series of related constant values in C#.

```
using System;

namespace ProgramOne
{
    class Program
    {
        enum CollegeDays
        {
            MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY
        }

        static void Main(string[] args)
        {
            foreach (var day in Enum.GetValues(typeof(CollegeDays)))
            {
                Console.WriteLine("{0} : {1}", day, (int) day);
            }

            Console.Read();
        }
    }
}
```

OUTPUT:

```
file:///C:/Users/CA172003/Documents/Visual Studio ...  
MONDAY : 0  
TUESDAY : 1  
WEDNESDAY : 2  
THURSDAY : 3  
FRIDAY : 4  
SATURDAY : 5  
  
_
```

2. Check Whether the Entered Year is a Leap Year or Not.

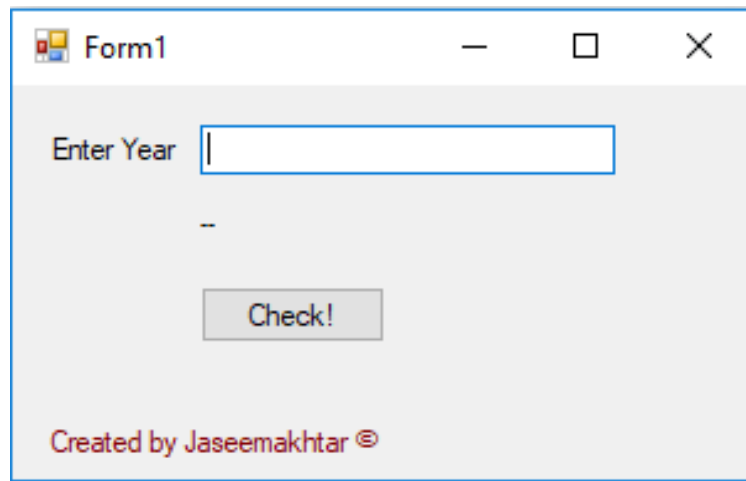
```
using System;
using System.Windows.Forms;

namespace ProgramThreeGUI
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            label2.Text = "";
            long year;
            try
            {
                year = Int64.Parse(textBox1.Text);

                if (checkYear(year))
                    label2.Text = year + " is a Leap year";
                else
                    label2.Text = year + " is not a Leap year";
            }
            catch (FormatException ex) {
                label2.Text = "Please enter only numbers";
            }
        }

        static bool checkYear(long year)
        {
            if (year % 400 == 0)
                return true;
            else if (year % 100 == 0)
                return false;
            else if (year % 4 == 0)
                return true;
            else
                return false;
        }
    }
}
```

OUTPUT:

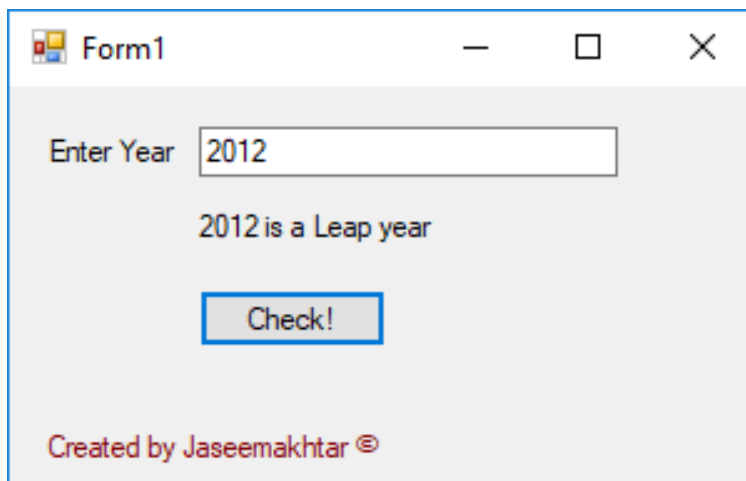
Form1

Enter Year

--

Check!

Created by Jaseemakhtar ©



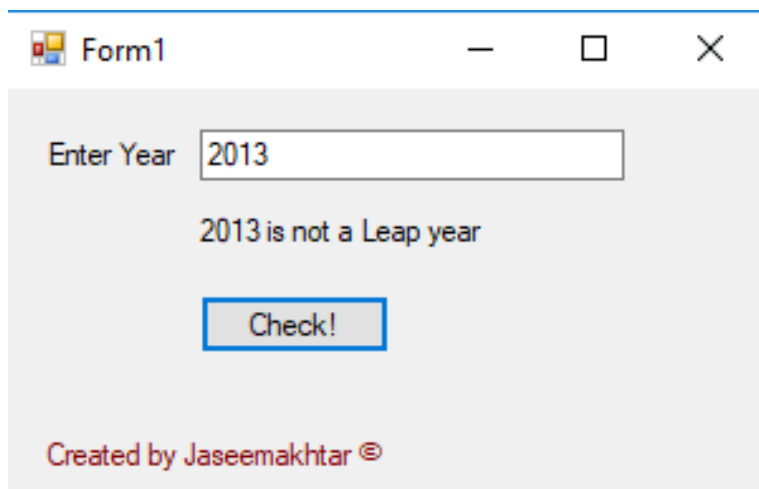
Form1

Enter Year

2012 is a Leap year

Check!

Created by Jaseemakhtar ©



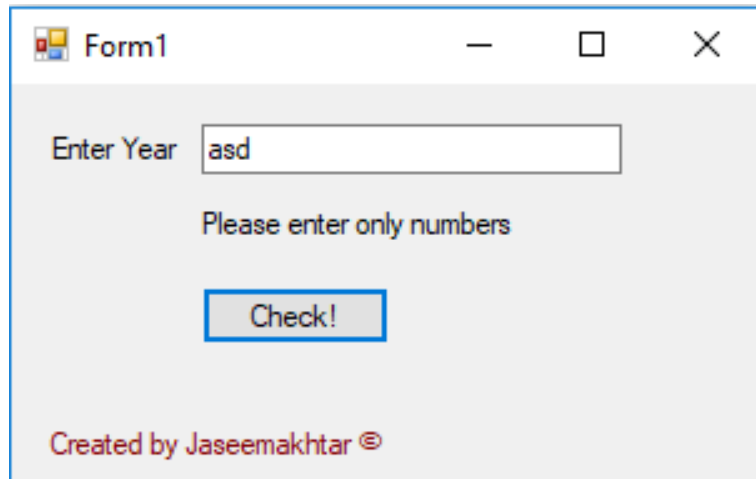
Form1

Enter Year

2013 is not a Leap year

Check!

Created by Jaseemakhtar ©

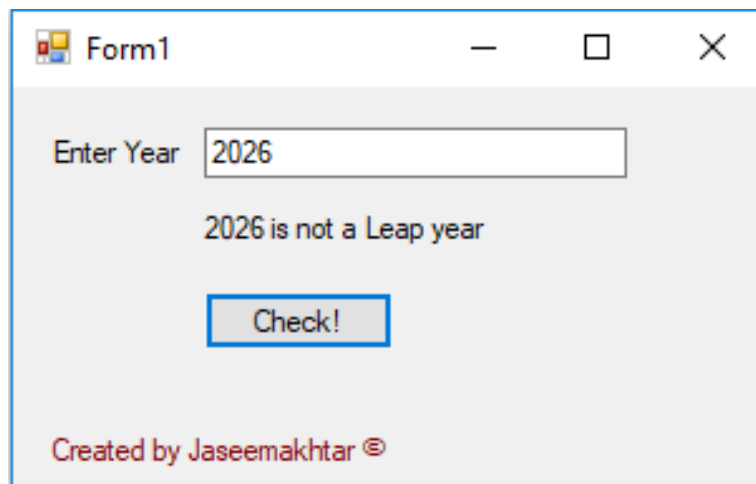


Form1

Enter Year

Please enter only numbers

Created by Jaseemakhtar ©



Form1

Enter Year

2026 is not a Leap year

Created by Jaseemakhtar ©

3. Program to display the addition, subtraction, multiplication and division of two number using console applications.

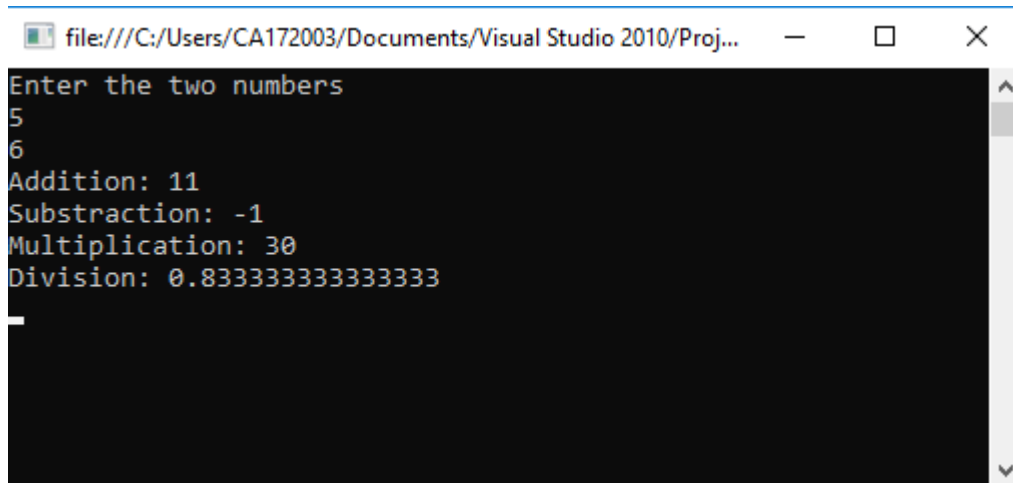
```
using System;

namespace ProgramFive
{
    class Program
    {
        static void Main(string[] args)
        {
            double num1, num2;
            double sum, sub, mul, div;

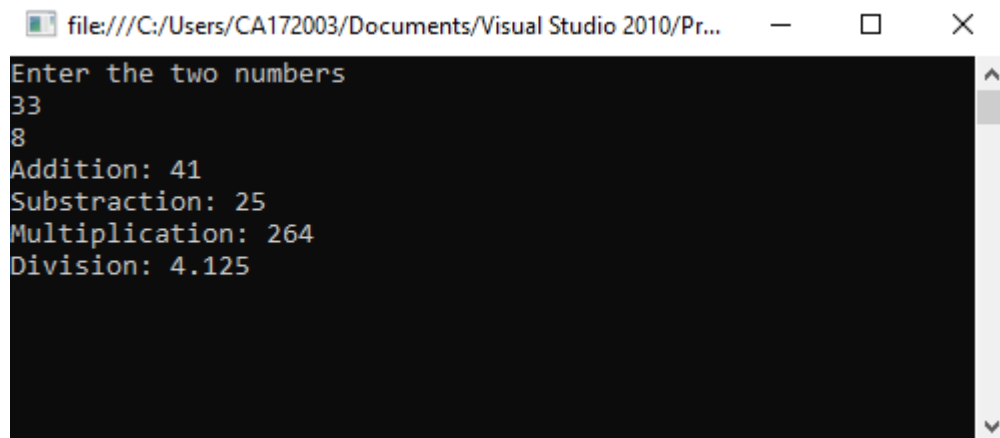
            Console.WriteLine("Enter the two numbers");
            num1 = Double.Parse(Console.ReadLine());
            num2 = Double.Parse(Console.ReadLine());

            sum = num1 + num2;
            sub = num1 - num2;
            mul = num1 * num2;
            div = num1 / num2;

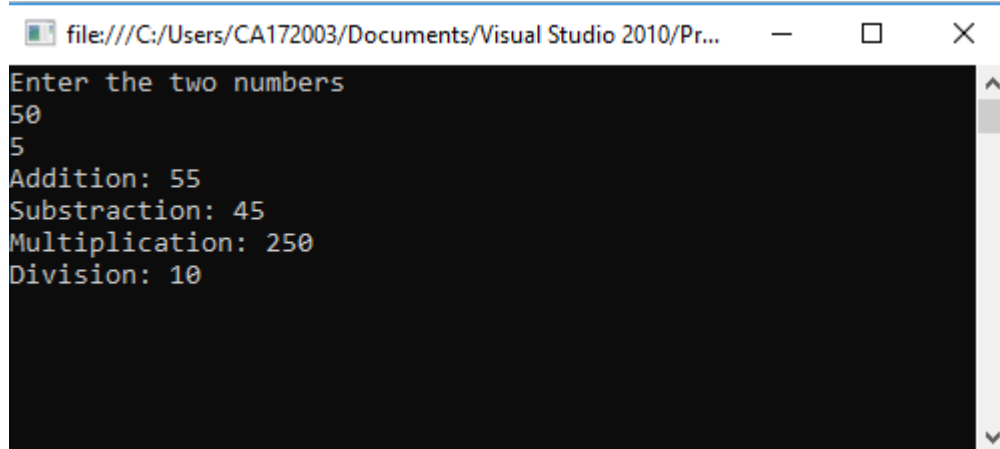
            Console.WriteLine("Addition: {0}", sum);
            Console.WriteLine("Substraction: {0}", sub);
            Console.WriteLine("Multiplication: {0}", mul);
            Console.WriteLine("Division: {0}", div);
            Console.ReadLine();
        }
    }
}
```

OUTPUT:

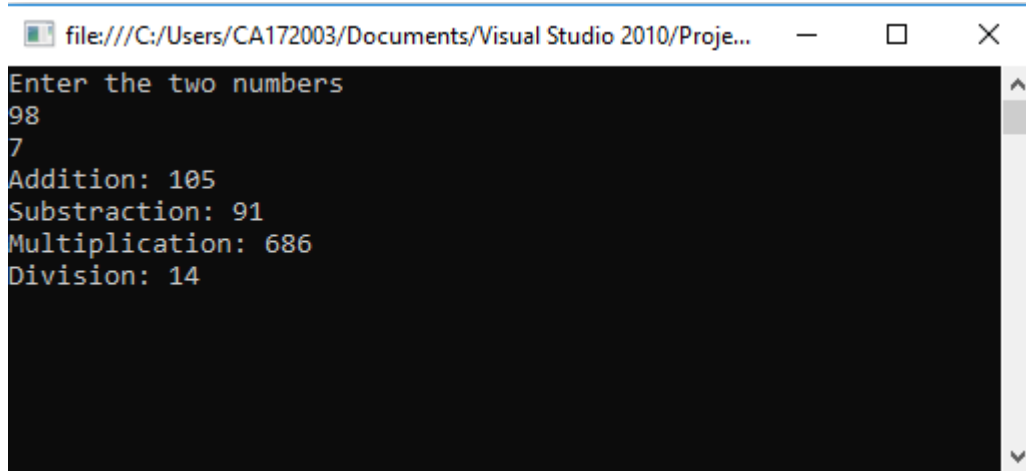
```
file:///C:/Users/CA172003/Documents/Visual Studio 2010/Proj...  
Enter the two numbers  
5  
6  
Addition: 11  
Substraction: -1  
Multiplication: 30  
Division: 0.8333333333333333
```



```
file:///C:/Users/CA172003/Documents/Visual Studio 2010/Pr...  
Enter the two numbers  
33  
8  
Addition: 41  
Substraction: 25  
Multiplication: 264  
Division: 4.125
```

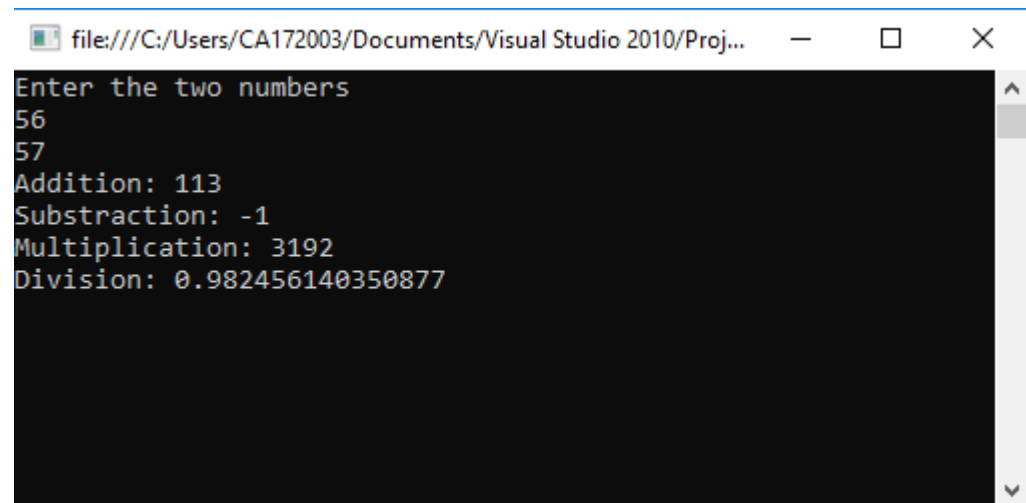


```
file:///C:/Users/CA172003/Documents/Visual Studio 2010/Pr...  
Enter the two numbers  
50  
5  
Addition: 55  
Substraction: 45  
Multiplication: 250  
Division: 10
```



A screenshot of a Windows console window titled "file:///C:/Users/CA172003/Documents/Visual Studio 2010/Proje...". The window contains the following text:

```
Enter the two numbers
98
7
Addition: 105
Substraction: 91
Multiplication: 686
Division: 14
```



A screenshot of a Windows console window titled "file:///C:/Users/CA172003/Documents/Visual Studio 2010/Proj...". The window contains the following text:

```
Enter the two numbers
56
57
Addition: 113
Substraction: -1
Multiplication: 3192
Division: 0.982456140350877
```

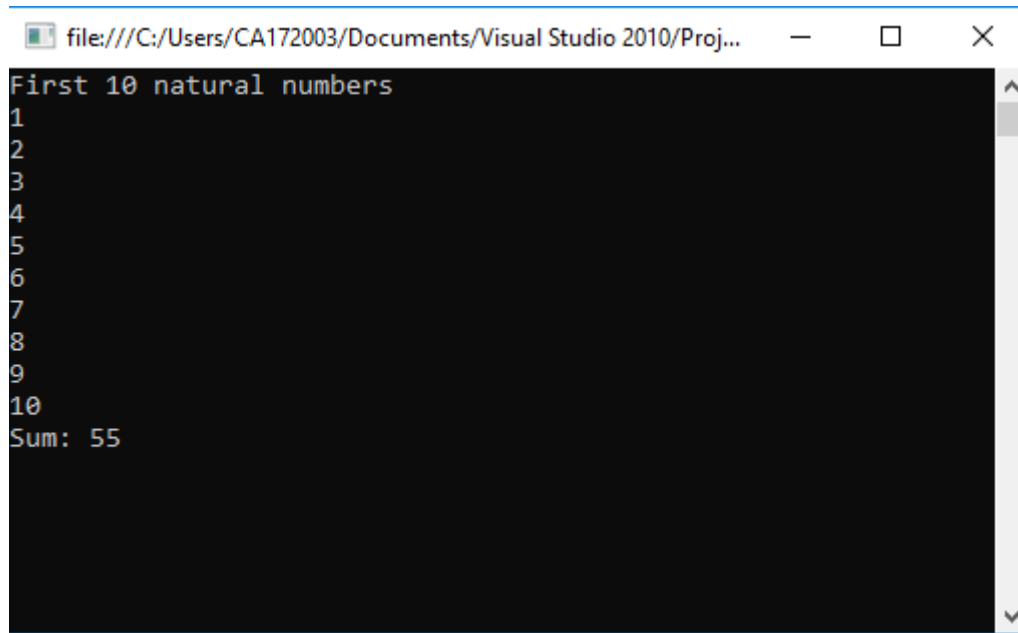

4. Program to display the first 10 natural numbers and their sum using console application.

```
using System;

namespace ProgramSix
{
    class Program
    {
        static void Main(string[] args)
        {
            int sum = 0;
            Console.WriteLine("First 10 natural numbers");

            for (int i = 1; i <= 10; i++)
            {
                sum += i;
                Console.WriteLine(i);
            }

            Console.WriteLine("Sum: {0}", sum);
            Console.ReadLine();
        }
    }
}
```

OUTPUT:

The screenshot shows a console window with the following output:

```
file:///C:/Users/CA172003/Documents/Visual Studio 2010/Proj...  
First 10 natural numbers  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
Sum: 55
```

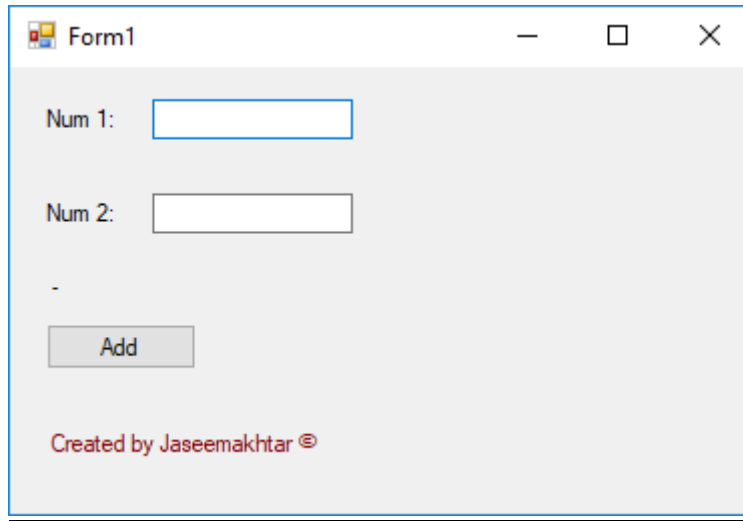
5. Program to display the addition using the windows application.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace ProgramSeven
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            int num1 = Int16.Parse(textBox1.Text);
            int num2 = Int16.Parse(textBox2.Text);

            int sum = num1 + num2;
            label1.Text = "Sum: " + sum;
        }
    }
}
```

OUTPUT:

Form1

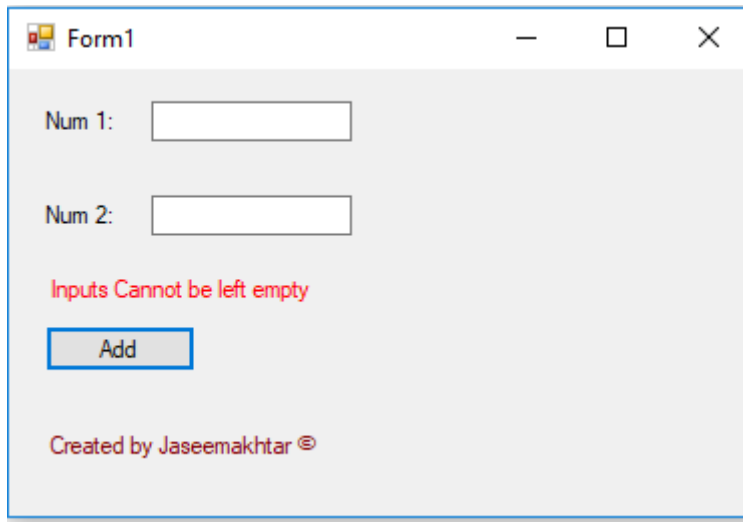
Num 1:

Num 2:

-

Add

Created by Jaseemakhtar ©



Form1

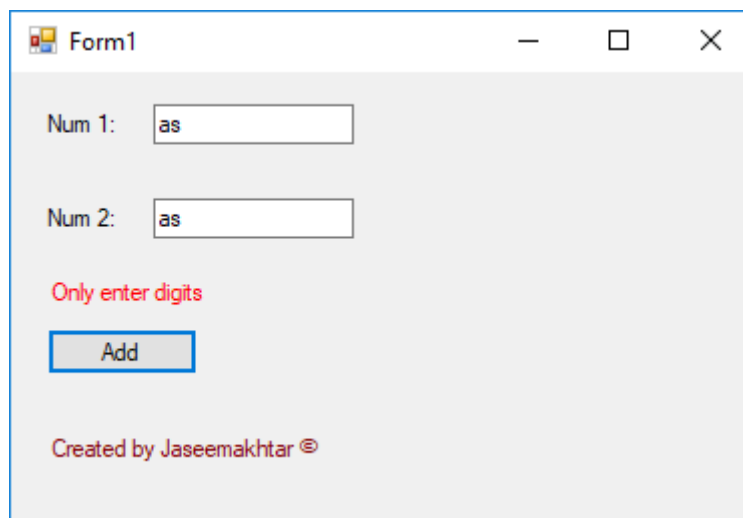
Num 1:

Num 2:

Inputs Cannot be left empty

Add

Created by Jaseemakhtar ©



Form1

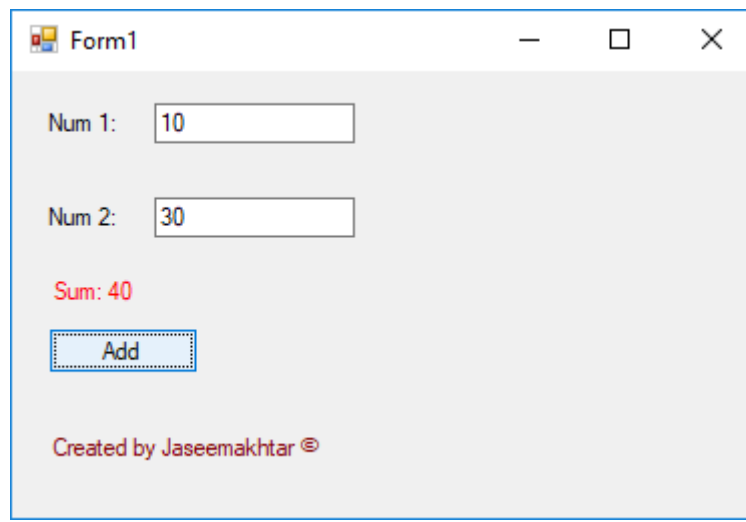
Num 1:

Num 2:

Only enter digits

Add

Created by Jaseemakhtar ©



Form1

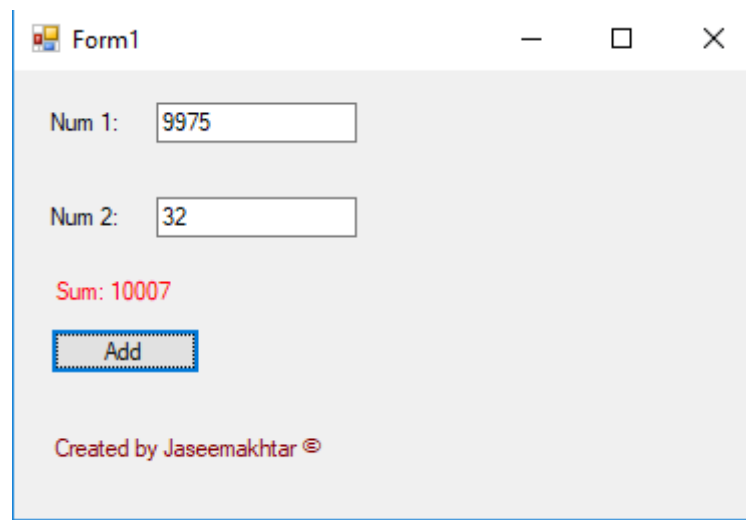
Num 1: 10

Num 2: 30

Sum: 40

Add

Created by Jaseemakhtar ©



Form1

Num 1: 9975

Num 2: 32

Sum: 10007

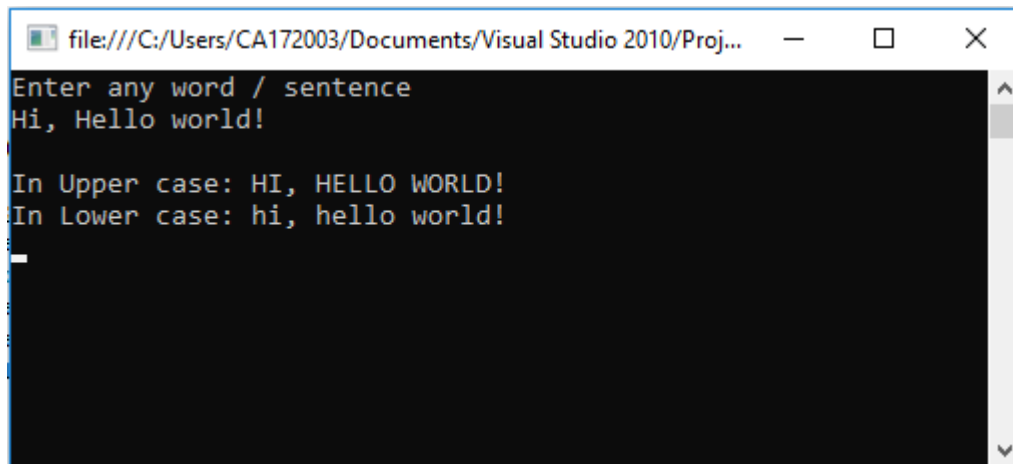
Add

Created by Jaseemakhtar ©

6. Write a program to convert input string from lower to upper and upper to lower case.

```
using System;

namespace ProgramNine
{
    class Program
    {
        static void Main(string[] args)
        {
            string input;
            Console.WriteLine("Enter any word / sentence");
            input = Console.ReadLine();
            Console.WriteLine("Upper case: {0}", input.ToUpper());
            Console.WriteLine("Lower case: {0}", input.ToLower());
            Console.ReadLine();
        }
    }
}
```

OUTPUT:

```
file:///C:/Users/CA172003/Documents/Visual Studio 2010/Proj...
Enter any word / sentence
Hi, Hello world!

In Upper case: HI, HELLO WORLD!
In Lower case: hi, hello world!
```

7. Find the second largest element in a single dimensional array.

using System;

namespace ProgramFourteen

{

class Program

{

static void Main(string[] args)

{

int n;

Console.WriteLine("Enter the size of the array");

n = Int16.Parse(Console.ReadLine());

int[] array = new int[n];

Console.WriteLine("Enter {0} elements into array", n);

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

array[i] = Int16.Parse(Console.ReadLine());

for (int i = 0; i < n; i++) {

int max = array[i];

for (int j = 0; j < n; j++) {

if (array[j] > max) {

int t = array[j];

array[j] = array[i];

array[i] = t;

}

}

}

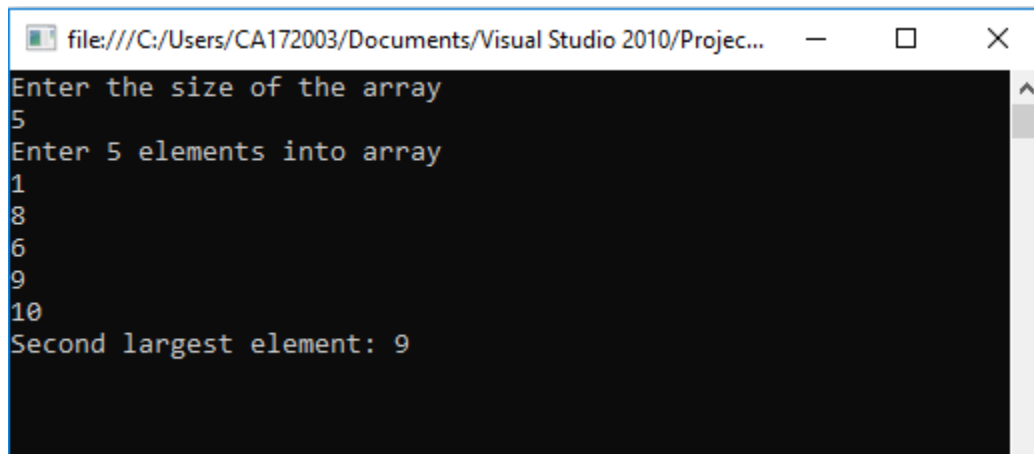
Console.WriteLine("Second largest element: {0}", array[n - 2]);

Console.ReadLine();

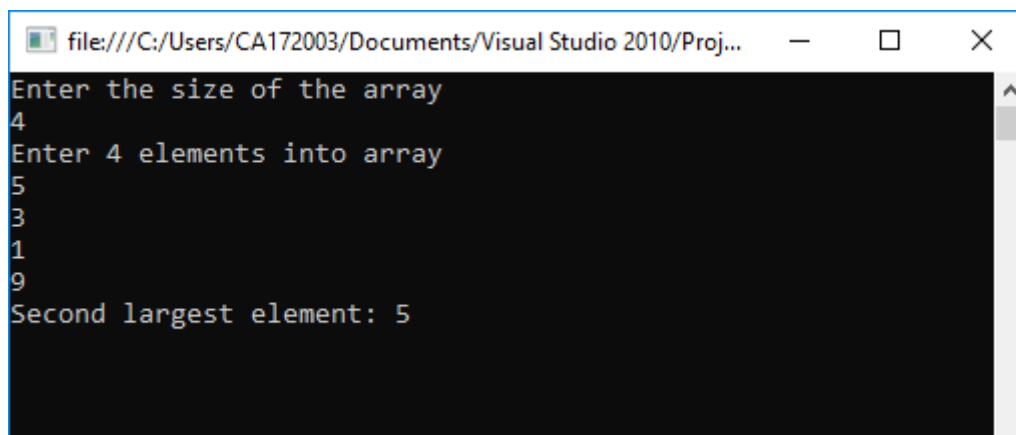
}

}

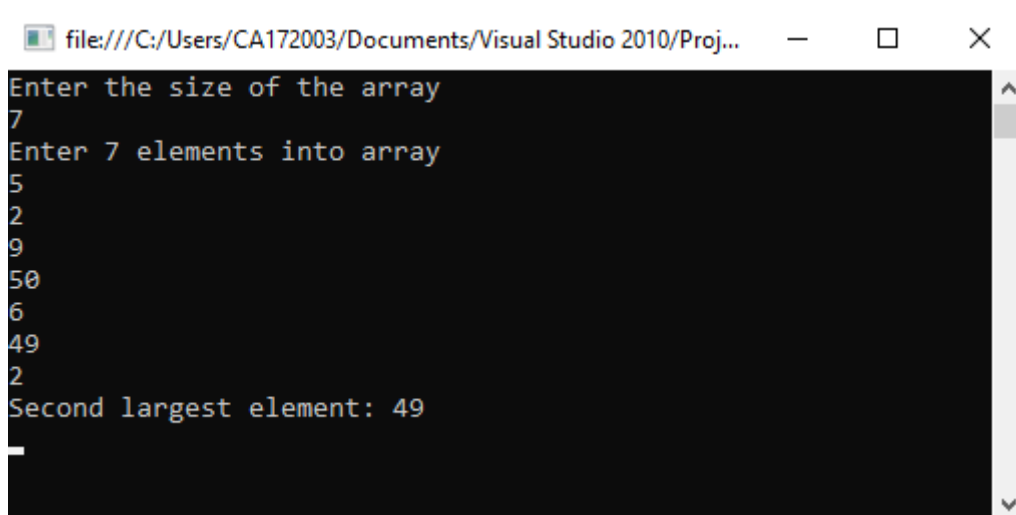
}

OUTPUT:

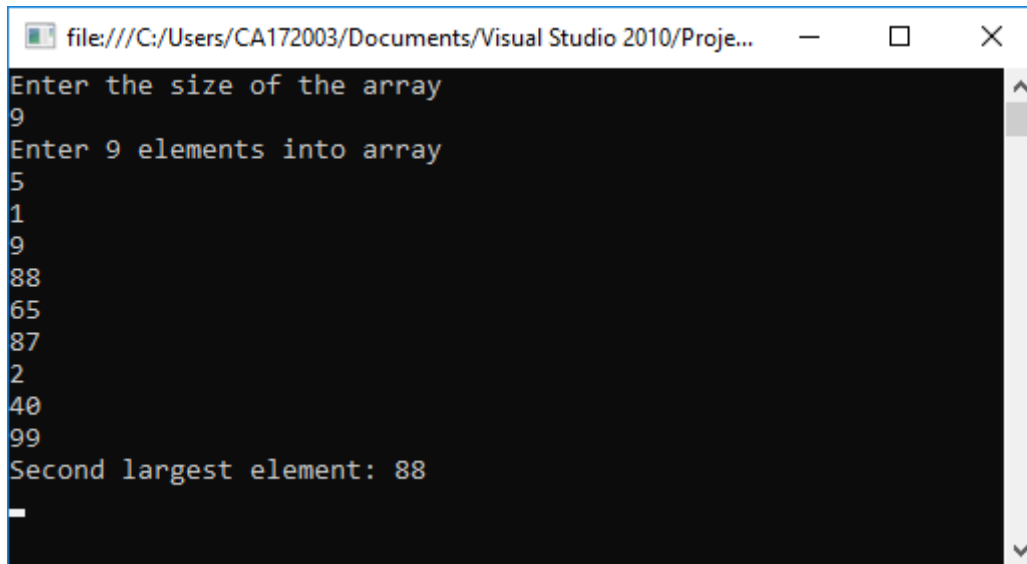
```
file:///C:/Users/CA172003/Documents/Visual Studio 2010/Projec...
Enter the size of the array
5
Enter 5 elements into array
1
8
6
9
10
Second largest element: 9
```



```
file:///C:/Users/CA172003/Documents/Visual Studio 2010/Projec...
Enter the size of the array
4
Enter 4 elements into array
5
3
1
9
Second largest element: 5
```

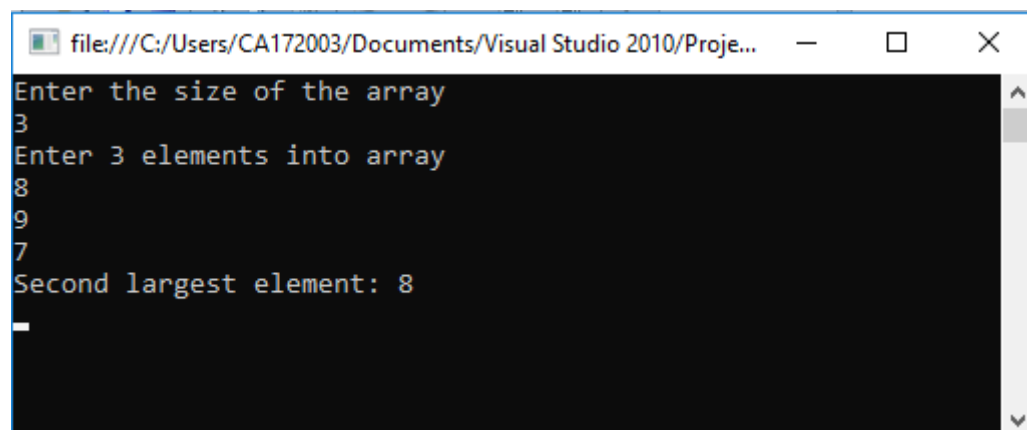


```
file:///C:/Users/CA172003/Documents/Visual Studio 2010/Projec...
Enter the size of the array
7
Enter 7 elements into array
5
2
9
50
6
49
2
Second largest element: 49
```



A screenshot of a Windows console window titled "file:///C:/Users/CA172003/Documents/Visual Studio 2010/Proje...". The window has standard Windows window controls (minimize, maximize, close). The console text is as follows:

```
Enter the size of the array
9
Enter 9 elements into array
5
1
9
88
65
87
2
40
99
Second largest element: 88
_
```



A screenshot of a Windows console window titled "file:///C:/Users/CA172003/Documents/Visual Studio 2010/Proje...". The window has standard Windows window controls (minimize, maximize, close). The console text is as follows:

```
Enter the size of the array
3
Enter 3 elements into array
8
9
7
Second largest element: 8
_
```

8. Program to illustrate the use of different properties in C#.

```
using System;

namespace ProgramFifteen
{
    class PropertiesDemo
    {
        private string name;
        private int age;

        public string Name{
            set {
                name = value;
            }
            get {
                return name;
            }
        }

        public int Age {
            set {
                if (value > 0)
                    age = value;
            }

            get {
                return age;
            }
        }

        static void Main(string[] args)
        {
            PropertiesDemo p = new PropertiesDemo();
            p.Name = "Jaseem";
            p.Age = 23;

            PropertiesDemo d = new PropertiesDemo();
            d.Name = "Mr.X";
            d.Age = -1;

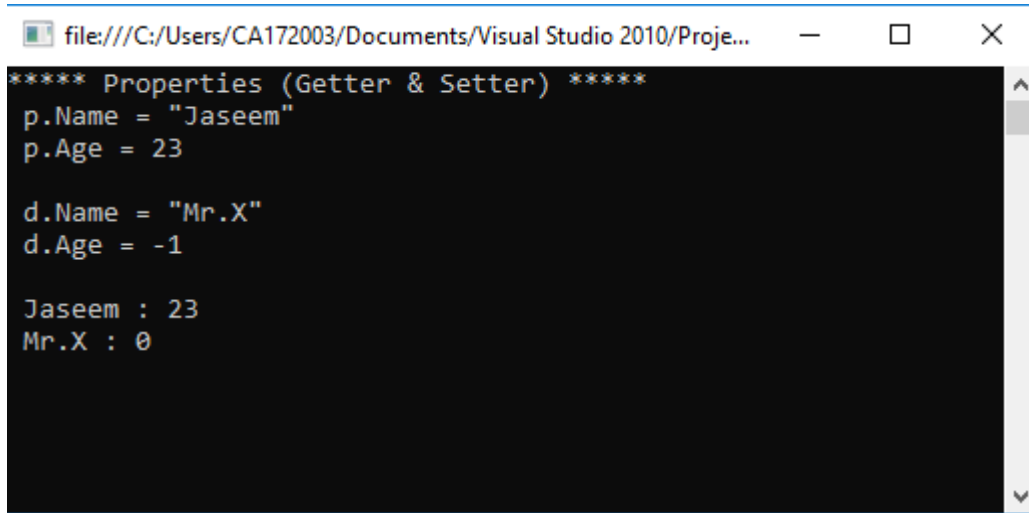
            Console.WriteLine("***** Properties (Getter & Setter) *****");
            Console.WriteLine(" p.Name = \"Jaseem\" ");
            Console.WriteLine(" p.Age = 23 ");

            Console.WriteLine();
        }
    }
}
```

```
    Console.WriteLine(" d.Name = \"Mr.X\" ");
    Console.WriteLine(" d.Age = -1 ");

    Console.WriteLine();

    Console.WriteLine(" {0} : {1}", p.Name, p.Age);
    Console.WriteLine(" {0} : {1}", d.Name, d.Age);
    Console.ReadLine();
}
}
}
```

OUTPUT:

```
file:///C:/Users/CA172003/Documents/Visual Studio 2010/Proje...  
***** Properties (Getter & Setter) *****  
p.Name = "Jaseem"  
p.Age = 23  
  
d.Name = "Mr.X"  
d.Age = -1  
  
Jaseem : 23  
Mr.X : 0
```

9. Demonstrate Command line arguments processing.

using System;

namespace ProgramSixteen

{

class Program

{

static void Main(string[] args)

{

if (args.Length >= 2)

{

int num1 = Int32.Parse(args[0]);

int num2 = Int32.Parse(args[1]);

Console.WriteLine("Addition of Command Line Arguments ({0}, {1})", num1,
num2);

int sum = num1 + num2;

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

}

else

{

Console.WriteLine("No command line arguments to process");

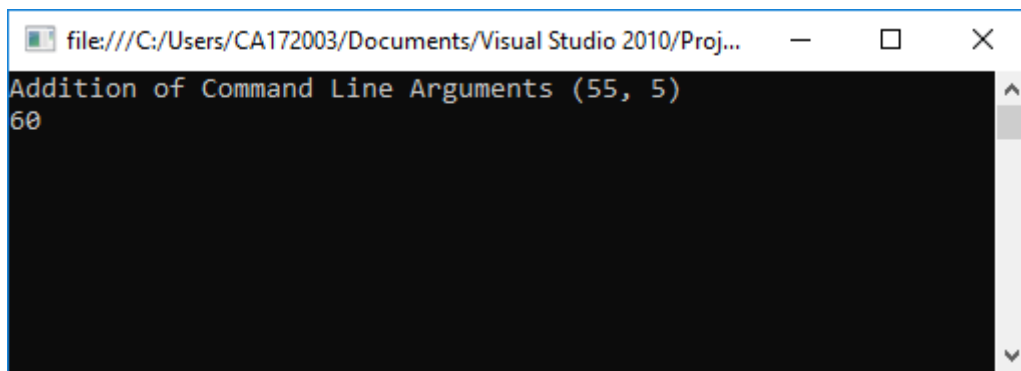
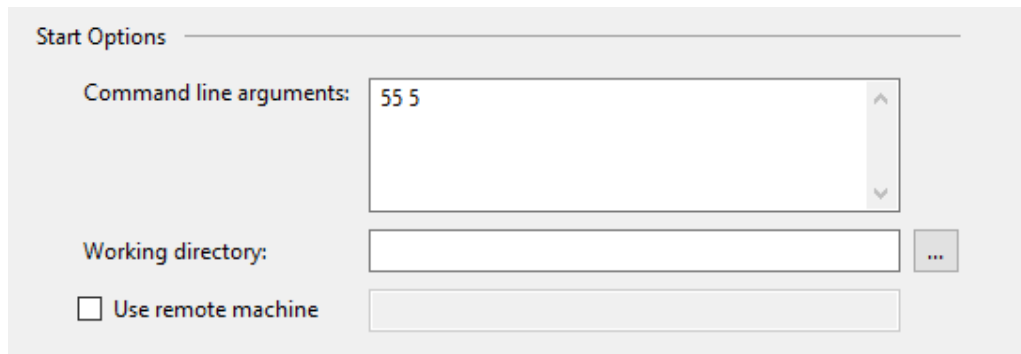
}

Console.ReadLine();

}

}

}

OUTPUT:

10. Create classes, they are reference types in C# and hence are allocated on the heap. Classes provide object-oriented constructs such as encapsulation, polymorphism, and inheritance. For instance, the program should print John. Doe twice, illustrating that objects are reference types, allocated on the heap implement the same using C#.

```
using System;

namespace ProgramTwo
{
    class Program
    {
        static void Main(string[] args)
        {
            User user1 = new User("Tony");
            Admin user2 = new Admin("Jaseem", "jaseem@gmail.com", "akhtar");

            Console.WriteLine("User 1:");
            Console.WriteLine("Name: {0}", user1.getName());
            Console.WriteLine("Email: {0}", user1.getEmail());

            Console.WriteLine();

            Console.WriteLine("User 2 (Admin):");
            Console.WriteLine("Name: {0}", user2.getName());
            Console.WriteLine("Email: {0}", user2.getEmail());
            Console.WriteLine("Password: {0}", user2.getPassword());

            Console.Read();
        }
    }

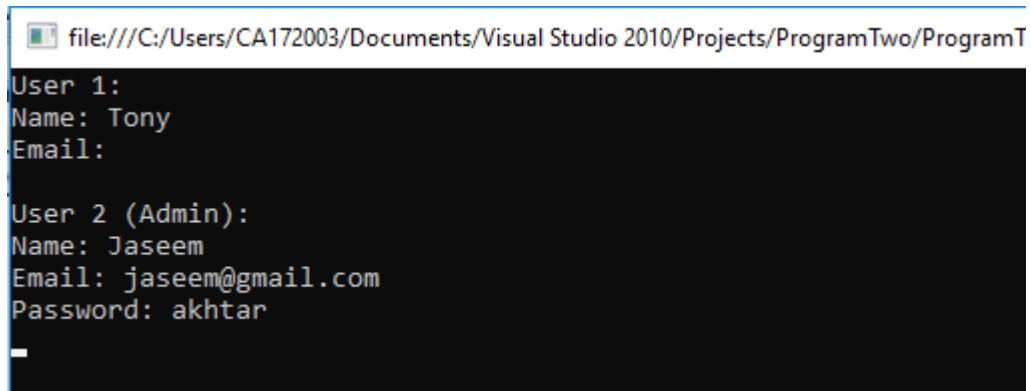
    class User {
        private string name;
        private string email;
```



```
public User(String name) {
    this.name = name;
}
public User(String name, String email)
{
    this.name = name;
    this.email = email;
}
public string getName() {
    return name;
}
public string getEmail()
{
    return email;
}
public void setName(string name)
{
    this.name = name;
}
public void setEmail(string email)
{
    this.email = email;
}
}

class Admin : User {
    private string password;
    public Admin(string name, string email, string password): base(name, email)
    {
        this.password = password;
    }
    public void setPassword(string password) {
        this.password = password;
    }

    public string getPassword() {
        return password;
    }
}
}
```

OUTPUT:

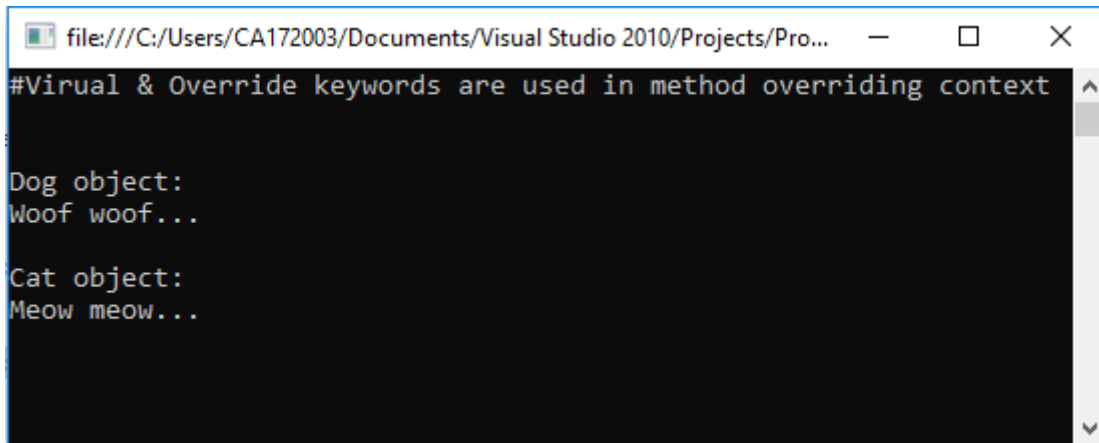
```
file:///C:/Users/CA172003/Documents/Visual Studio 2010/Projects/ProgramTwo/ProgramT
User 1:
Name: Tony
Email:

User 2 (Admin):
Name: Jaseem
Email: jaseem@gmail.com
Password: akhtar
_
```

11. Demonstrate Use of Virtual and override keyword in C# with a simple Program.

using System;

```
class Animal
{
    public virtual void speak() { }
}
class Dog : Animal
{
    public override void speak()
    {
        Console.WriteLine("Woof woof...");
    }
}
class Cat : Animal
{
    public override void speak()
    {
        Console.WriteLine("Meow meow...");
    }
}
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("#Virtual & Override keywords are used in method overriding context\n");
        Animal dog = new Dog();
        Animal cat = new Cat();
        Console.WriteLine("\nDog object:");
        dog.speak();
        Console.WriteLine("\nCat object:");
        cat.speak();
        Console.ReadLine();
    }
}
```

OUTPUT:

```
file:///C:/Users/CA172003/Documents/Visual Studio 2010/Projects/Pro...  
#Virtual & Override keywords are used in method overriding context  
Dog object:  
Woof woof...  
Cat object:  
Meow meow...
```

12. Perform operator overloading.

using System;

```
namespace ProgramTwelve
{
    class Rectangle
    {
        int width;
        int height;

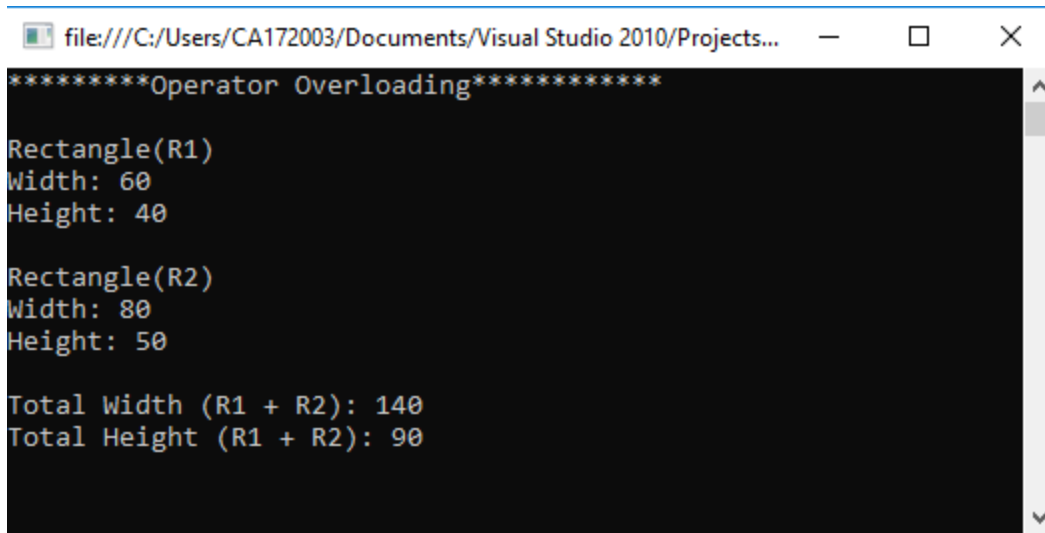
        Rectangle(int width, int height) {
            this.width = width;
            this.height = height;
        }

        public static Rectangle operator +(Rectangle a, Rectangle b)
        {
            int totalWidth = a.width + b.width;
            int totalHeight = a.height + b.height;
            return new Rectangle(totalWidth, totalHeight);
        }

        static void Main(string[] args)
        {
            Rectangle r1 = new Rectangle(60, 40);
            Rectangle r2 = new Rectangle(80, 50);
            Console.WriteLine("*****Operator Overloading*****\n");
            Console.WriteLine("Rectangle(R1)");
            Console.WriteLine("Width: {0}", r1.width);
            Console.WriteLine("Height: {0}", r1.height);

            Console.WriteLine();
            Console.WriteLine("Rectangle(R2)");
            Console.WriteLine("Width: {0}", r2.width);
            Console.WriteLine("Height: {0}", r2.height);

            Console.WriteLine();
            Rectangle r3 = r1 + r2;
            Console.WriteLine("Total Width (R1 + R2): {0}", r3.width);
            Console.WriteLine("Total Height (R1 + R2): {0}", r3.height);
            Console.Read();
        }
    }
}
```

OUTPUT:

```
file:///C:/Users/CA172003/Documents/Visual Studio 2010/Projects... — □ ×
*****Operator Overloading*****
Rectangle(R1)
Width: 60
Height: 40
Rectangle(R2)
Width: 80
Height: 50
Total Width (R1 + R2): 140
Total Height (R1 + R2): 90
```

13. Describe Arrays and Strings methods with suitable C# program.

using System;

namespace ProgramFour

```
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] array = { 1, 4, 6, 2, 8, 9, 7 };
            Console.WriteLine("Properties & Methods of an Array: ");
            displayArray(array);
            Console.WriteLine();
            Console.WriteLine("Length: {0}", array.Length);
            Console.WriteLine("Rank: {0}", array.Rank);
            Console.WriteLine("Max(): {0}", array.Max());
            Console.WriteLine("Min(): {0}", array.Min());
            Console.WriteLine("Sum(): {0}", array.Sum());

            Console.WriteLine("Array.Reverse()");
            Array.Reverse(array);
            displayArray(array);

            Console.WriteLine("Array.Sort()");
            Array.Sort(array);
            displayArray(array);

            Console.WriteLine();
            Console.WriteLine("_____");
            Console.WriteLine();

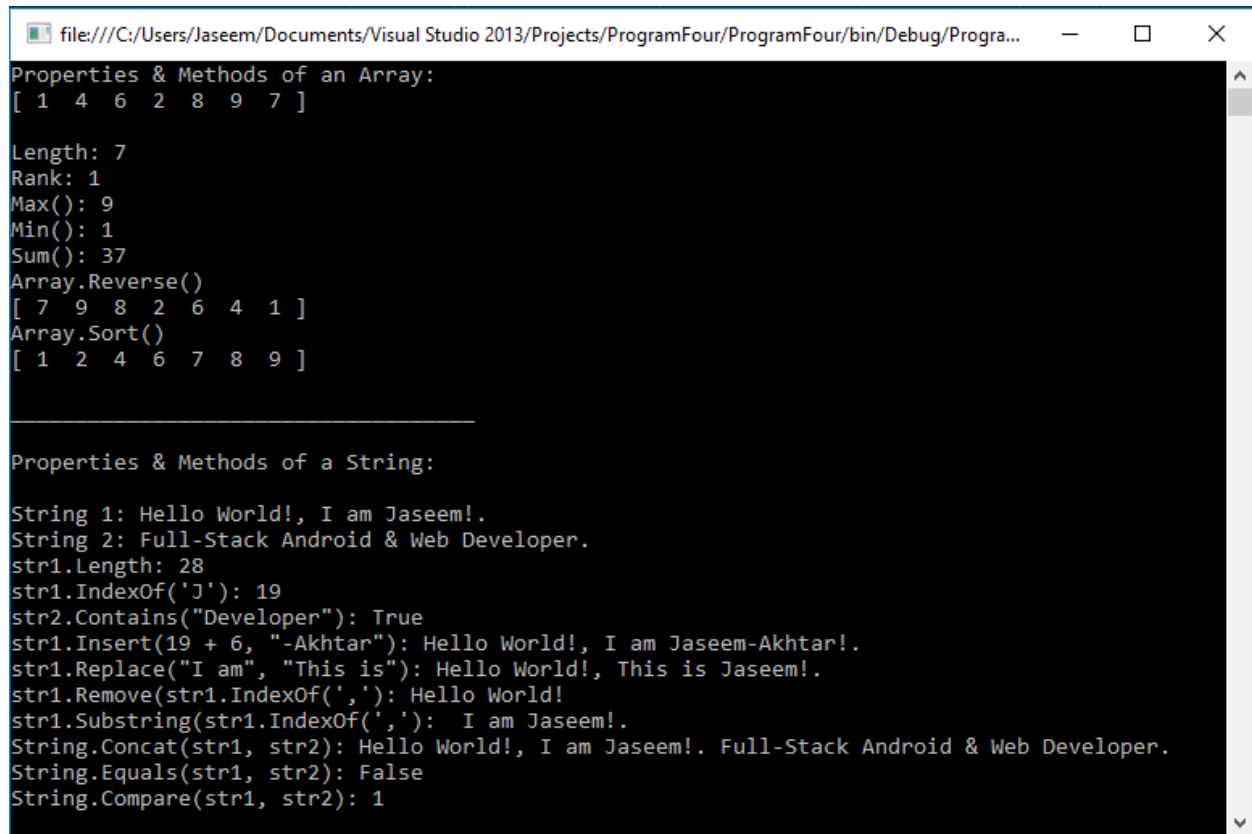
            Console.WriteLine("Properties & Methods of a String: ");
            String str1 = "Hello World!, I am Jaseem!. ";
            Console.WriteLine();
            String str2 = "Full-Stack Android & Web Developer.";
            Console.WriteLine("String 1: {0}", str1);
            Console.WriteLine("String 2: {0}", str2);
            Console.WriteLine("str1.Length: {0}", str1.Length);
            Console.WriteLine("str1.IndexOf('J'): {0}", str1.IndexOf('J'));
```

```
Console.WriteLine("str2.Contains(\"Developer\"): {0}", str2.Contains("Developer"));
    Console.WriteLine("str1.Insert(19 + 6, \"-Akhtar\"): {0}",
str1.Insert(str1.IndexOf('J') + 6, "-Akhtar"));
Console.WriteLine("str1.Replace(\"I am\", \"This is\"): {0}", str1.Replace("I am", "This
is"));
Console.WriteLine("str1.Remove(str1.IndexOf(','): {0}", str1.Remove(str1.IndexOf(',')));
    Console.WriteLine("str1.Substring(str1.IndexOf(','): {0}",
str1.Substring(str1.IndexOf(',') + 1));

Console.WriteLine("String.Concat(str1, str2): {0}", String.Concat(str1, str2));
Console.WriteLine("String.Equals(str1, str2): {0}", String.Equals(str1, str2));
Console.WriteLine("String.Compare(str1, str2): {0}", String.Compare(str1, str2));

Console.ReadLine();
}

static void displayArray(int[] a)
{
    Console.Write("[");
    for (int i = 0; i < a.Length; i++)
    {
        Console.Write(" {0} ", a[i]);
    }
    Console.WriteLine("]");
}
}
```


OUTPUT:

```
file:///C:/Users/Jaseem/Documents/Visual Studio 2013/Projects/ProgramFour/ProgramFour/bin/Debug/Progra...
Properties & Methods of an Array:
[ 1 4 6 2 8 9 7 ]

Length: 7
Rank: 1
Max(): 9
Min(): 1
Sum(): 37
Array.Reverse()
[ 7 9 8 2 6 4 1 ]
Array.Sort()
[ 1 2 4 6 7 8 9 ]

Properties & Methods of a String:

String 1: Hello World!, I am Jaseem!.
String 2: Full-Stack Android & Web Developer.
str1.Length: 28
str1.IndexOf('J'): 19
str2.Contains("Developer"): True
str1.Insert(19 + 6, "-Akhtar"): Hello World!, I am Jaseem-Akhtar!.
str1.Replace("I am", "This is"): Hello World!, This is Jaseem!.
str1.Remove(str1.IndexOf(',')): Hello World!
str1.Substring(str1.IndexOf(',')): I am Jaseem!.
String.Concat(str1, str2): Hello World!, I am Jaseem!. Full-Stack Android & Web Developer.
String.Equals(str1, str2): False
String.Compare(str1, str2): 1
```

14. Describe delegates, events, errors and exceptions.Program.cs

```
using System;

namespace ProgramThirteen
{
    class Program
    {
        static void Main(string[] args)
        {
            Car car = new Car("Tesla");
            car.aboutToBlowListener += new Car.EventHandler(aboutToBlow);
            car.explodedListener += new Car.EventHandler(exploded);

            Console.WriteLine("*****Speeding Up*****");

            try
            {
                for (int i = 0; i < 20; i++)
                {
                    car.accelerate(20);
                }
            }
            catch (Exception e)
            {
                Console.WriteLine("Exception: Car is dead already!");
            }

            Console.ReadLine();
        }

        public static void aboutToBlow(string msg)
        {
            Console.WriteLine("-> Reporting: {0}", msg);
        }

        public static void exploded(string msg)
        {
            Console.WriteLine("-> Reporting: {0}", msg);
            throw new Exception("Car dead");
        }
    }
}
```

Car.cs

using System;

namespace ProgramThirteen

```
{
    class Car
    {
        public delegate void EventHandler(string msg);

        public event EventHandler explodeListener;
        public event EventHandler aboutToBlowListener;

        private string name;
        private bool isExhausted;
        private int currentSpeed;
        private const int maxSpeed = 140;

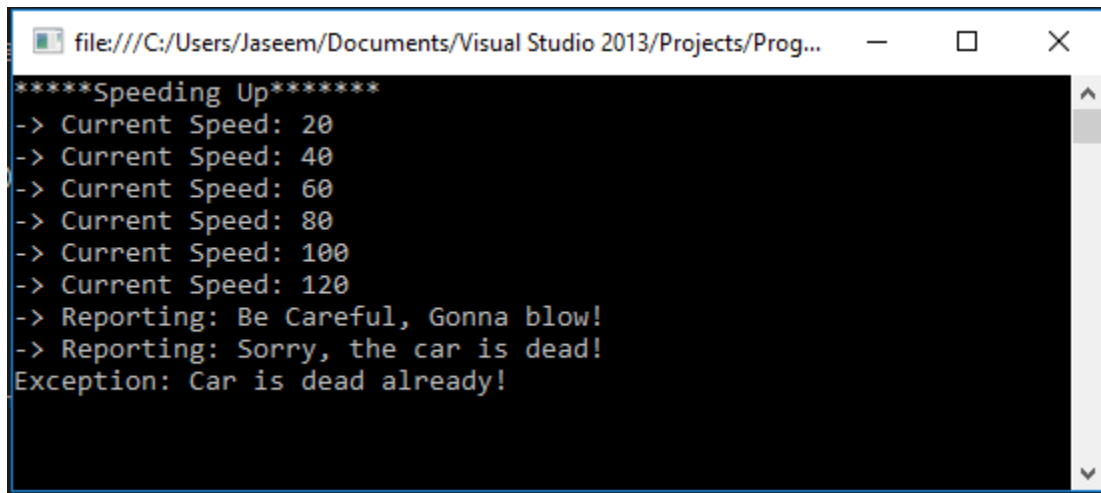
        public Car(String name)
        {
            this.name = name;
        }

        public void accelerate(int delta)
        {
            if (isExhausted)
            {
                if (explodeListener != null)
                    explodeListener("Sorry, the car is dead!");
            }
            else
            {
                currentSpeed += delta;

                if (10 >= maxSpeed - currentSpeed && aboutToBlowListener != null)
                {
                    aboutToBlowListener("Be Careful, Gonna blow!");
                }

                if (currentSpeed >= maxSpeed)
                    isExhausted = true;
                else
                    Console.WriteLine("-> Current Speed: {0}", currentSpeed);
            }
        }
    }
}
```

```
}  
}
```

OUTPUT:

```
file:///C:/Users/Jaseem/Documents/Visual Studio 2013/Projects/Prog...  
*****Speeding Up*****  
-> Current Speed: 20  
-> Current Speed: 40  
-> Current Speed: 60  
-> Current Speed: 80  
-> Current Speed: 100  
-> Current Speed: 120  
-> Reporting: Be Careful, Gonna blow!  
-> Reporting: Sorry, the car is dead!  
Exception: Car is dead already!
```

15. Program to multiply to matrices using Rectangular arrays.

```
using System;
```

```
namespace MatrixMultiplication
```

```
{
```

```
    class Program
```

```
    {
```

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

```
        {
```

```
            int n1, n2, m1, m2;
```

```
            int[,] matrixA, matrixB;
```

```
            m1 = readInt("Enter the value of m, for first M x N matrix\n");
```

```
            n1 = readInt("Enter the value of n, for first M x N matrix\n");
```

```
            m2 = readInt("Enter the value of m, for second M x N matrix\n");
```

```
            n2 = readInt("Enter the value of n, for second M x N matrix\n");
```

```
            if (n1 != m2)
```

```
            {
```

```
                Console.WriteLine("Sorry! Matrices cannot be multiplied\n");
```

```
            }
```

```
            else
```

```
            {
```

```
                matrixA = new int[m1, n1];
```

```
                matrixB = new int[m2, n2];
```

```
                matrixA = readMatrix(m1, n1);
```

```
                matrixB = readMatrix(m2, n2);
```

```
                displayMatrix(matrixA, m1, n1);
```

```
                displayMatrix(matrixB, m2, n2);
```

```
                int[,] matrixC = new int[m1, n2];
```

```
                for (int m = 0; m < m1; m++)
```

```
                {
```

```
                    for (int n = 0; n < n2; n++)
```

```
                    {
```

```
                        int sum = 0;
```

```
                        for (int k = 0; k < m2; k++)
```

```
                            sum += matrixA[m, k] * matrixB[k, n];
```

```
                        matrixC[m, n] = sum;
```

```
        }
    }

    Console.WriteLine("Resultant Matrix: ");
    displayMatrix(matrixC, m1, n2);
}
Console.ReadLine();

}

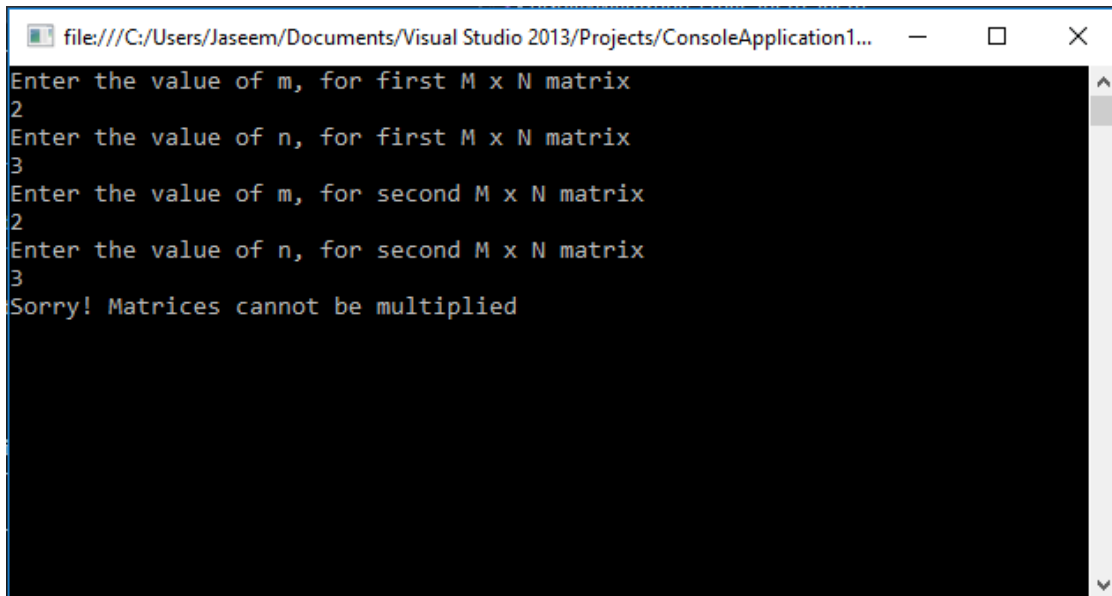
static int readInt(String message)
{
    Console.Write(message);
    return Convert.ToInt16(Console.ReadLine());
}

static int[,] readMatrix(int m, int n)
{
    int[,] mat = new int[m, n];

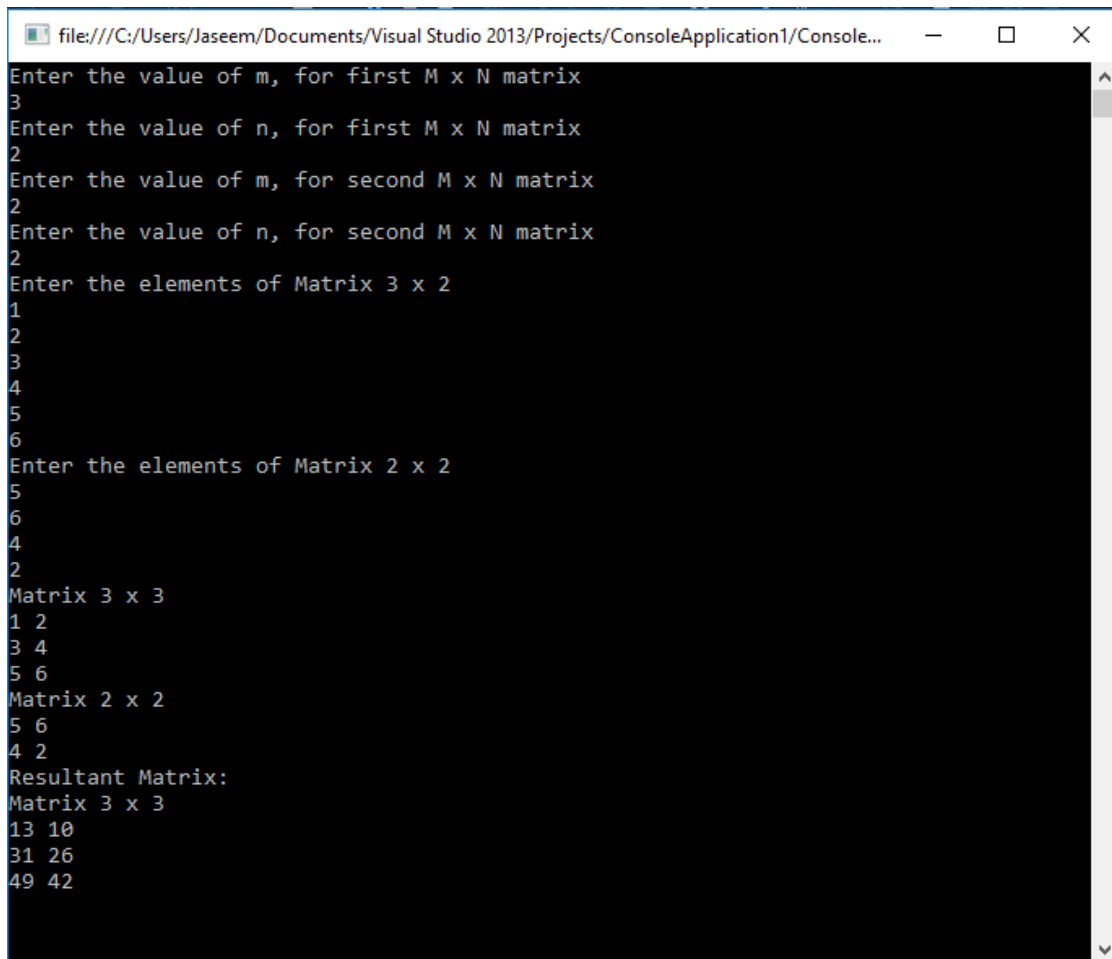
    Console.WriteLine("Enter the elements of Matrix {0} x {1}", m, n);
    for (int i = 0; i < m; i++)
        for (int j = 0; j < n; j++)
            mat[i, j] = readInt("");

    return mat;
}

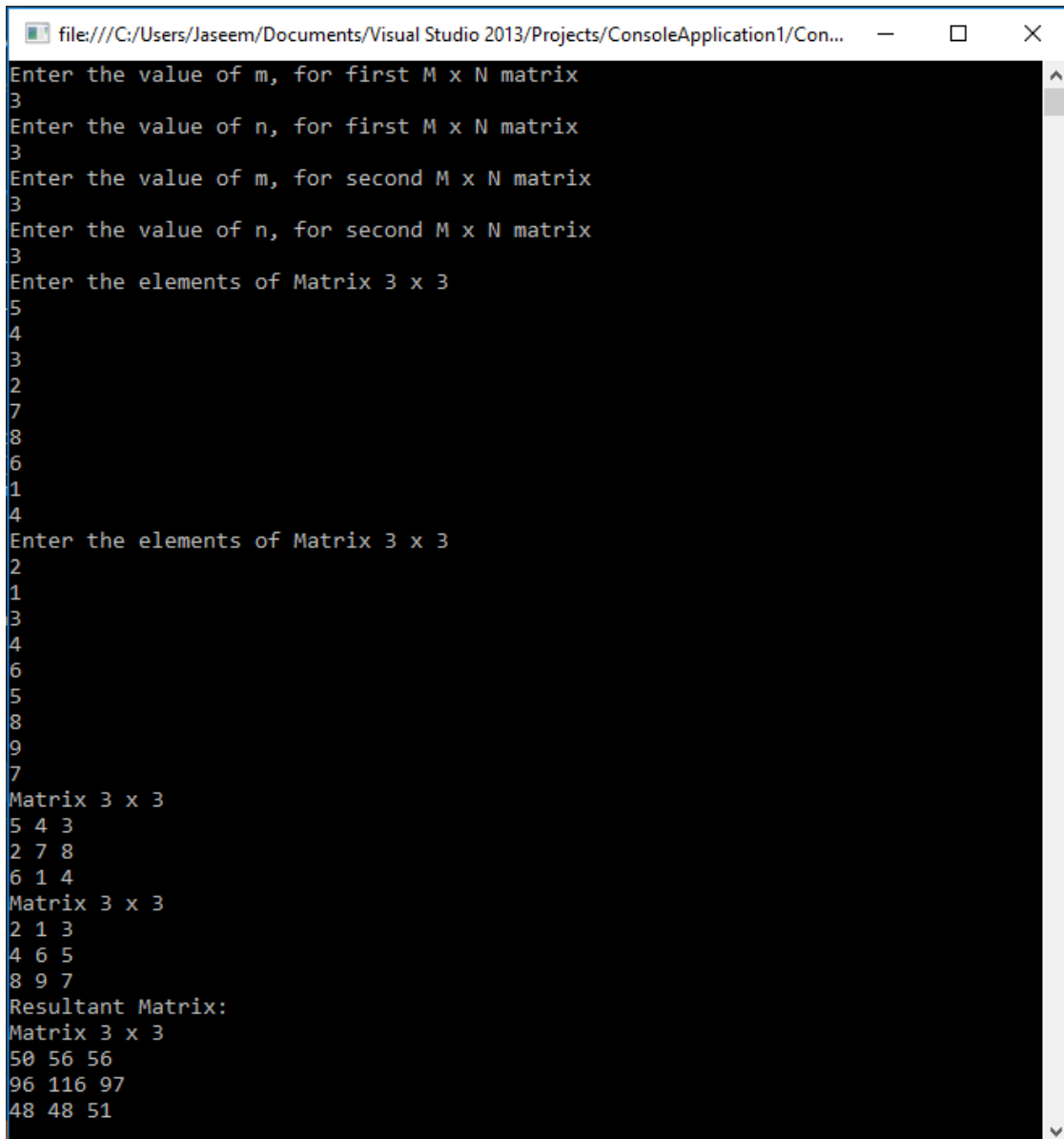
static void displayMatrix(int[,] mat, int m, int n)
{
    Console.WriteLine("Matrix {0} x {1}", m, m);
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
            Console.Write("{0} ", mat[i, j]);
        Console.WriteLine("");
    }
}
}
```

OUTPUT:

```
file:///C:/Users/Jaseem/Documents/Visual Studio 2013/Projects/ConsoleApplication1...
Enter the value of m, for first M x N matrix
2
Enter the value of n, for first M x N matrix
3
Enter the value of m, for second M x N matrix
2
Enter the value of n, for second M x N matrix
3
Sorry! Matrices cannot be multiplied
```

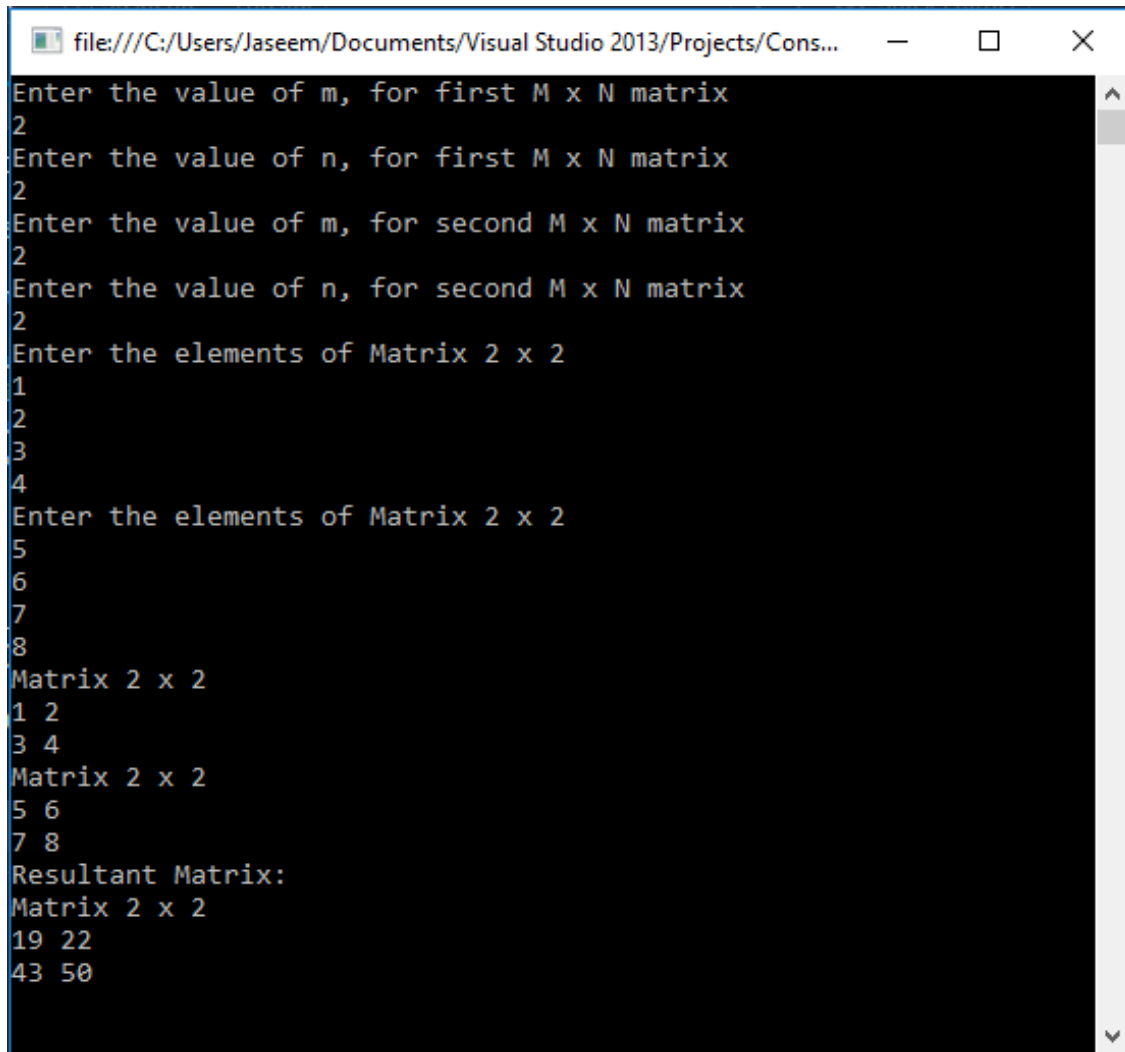


```
file:///C:/Users/Jaseem/Documents/Visual Studio 2013/Projects/ConsoleApplication1/Console...
Enter the value of m, for first M x N matrix
3
Enter the value of n, for first M x N matrix
2
Enter the value of m, for second M x N matrix
2
Enter the value of n, for second M x N matrix
2
Enter the elements of Matrix 3 x 2
1
2
3
4
5
6
Enter the elements of Matrix 2 x 2
5
6
4
2
Matrix 3 x 3
1 2
3 4
5 6
Matrix 2 x 2
5 6
4 2
Resultant Matrix:
Matrix 3 x 3
13 10
31 26
49 42
```



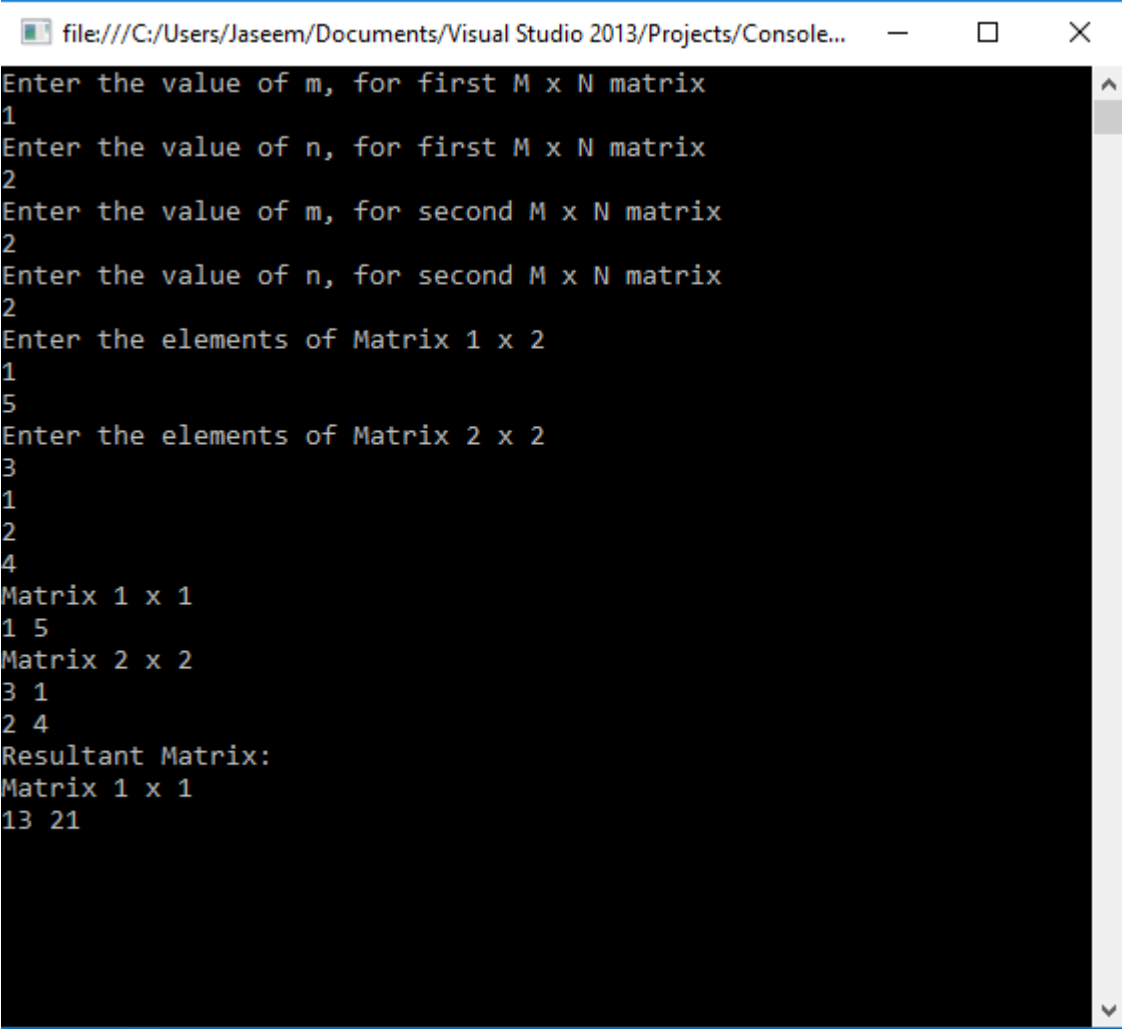
The screenshot shows a console window titled "file:///C:/Users/Jaseem/Documents/Visual Studio 2013/Projects/ConsoleApplication1/Con...". The application prompts the user to enter dimensions and elements for two 3x3 matrices. The first matrix is entered with dimensions 3x3 and elements 5, 4, 3; 2, 7, 8; 6, 1, 4. The second matrix is entered with dimensions 3x3 and elements 2, 1, 3; 4, 6, 5; 8, 9, 7. The resultant matrix is then displayed as 50, 56, 56; 96, 116, 97; 48, 48, 51.

```
file:///C:/Users/Jaseem/Documents/Visual Studio 2013/Projects/ConsoleApplication1/Con...
Enter the value of m, for first M x N matrix
3
Enter the value of n, for first M x N matrix
3
Enter the value of m, for second M x N matrix
3
Enter the value of n, for second M x N matrix
3
Enter the elements of Matrix 3 x 3
5
4
3
2
7
8
6
1
4
Enter the elements of Matrix 3 x 3
2
1
3
4
6
5
8
9
7
Matrix 3 x 3
5 4 3
2 7 8
6 1 4
Matrix 3 x 3
2 1 3
4 6 5
8 9 7
Resultant Matrix:
Matrix 3 x 3
50 56 56
96 116 97
48 48 51
```

The screenshot shows a console window titled "file:///C:/Users/Jaseem/Documents/Visual Studio 2013/Projects/Cons...". The window contains the following text:

```
Enter the value of m, for first M x N matrix
2
Enter the value of n, for first M x N matrix
2
Enter the value of m, for second M x N matrix
2
Enter the value of n, for second M x N matrix
2
Enter the elements of Matrix 2 x 2
1
2
3
4
Enter the elements of Matrix 2 x 2
5
6
7
8
Matrix 2 x 2
1 2
3 4
Matrix 2 x 2
5 6
7 8
Resultant Matrix:
Matrix 2 x 2
19 22
43 50
```



The screenshot shows a console window titled "file:///C:/Users/Jaseem/Documents/Visual Studio 2013/Projects/Console...". The window contains the following text:

```
Enter the value of m, for first M x N matrix
1
Enter the value of n, for first M x N matrix
2
Enter the value of m, for second M x N matrix
2
Enter the value of n, for second M x N matrix
2
Enter the elements of Matrix 1 x 2
1
5
Enter the elements of Matrix 2 x 2
3
1
2
4
Matrix 1 x 1
1 5
Matrix 2 x 2
3 1
2 4
Resultant Matrix:
Matrix 1 x 1
13 21
```

16. Work with Page using ASP.Net.

Default.aspx

```
<% @ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %>
```

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

```
  <title>Game - Hit the button</title>
```

```
  <style>
```

```
    html{
```

```
      height: 100%;
```

```
    }
```

```
    body{
```

```
      min-height: 100%;
```

```
      padding: 0;
```

```
      margin: 0;
```

```
    }
```

```
    *{
```

```
      font-family: Arial;
```

```
    }
```

```
    .container{
```

```
      height: 100vh;
```

```
      display: flex;
```

```
      flex-direction: column;
```

```
      align-items:center;
```

```
    }
```

```
    .space{
```

```
      flex-grow: 1;
```

```
    }
```

```
    footer{
```

```
      font-size: 0.8em;
```

```
      padding: .4em;
```

```
      color: gray;
```

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <form id="form1" runat="server">
```

```
    <div class="container">
```

```
      <h1>Welcome to Game - Hit the button!</h1>
```

```
      <asp:Label ID="lblOutput" Text="Your score is 0" runat="server" />
```

```
<asp:button id="clickMeButton" runat="server" text="Hit!"
onClick="clickMeButton_Click" />
<div class="space"></div>
<footer>&copy; 2019 Jaseemakhtar. All Rights Reserved.</footer>
</div>
</form>
</body>
</html>
```

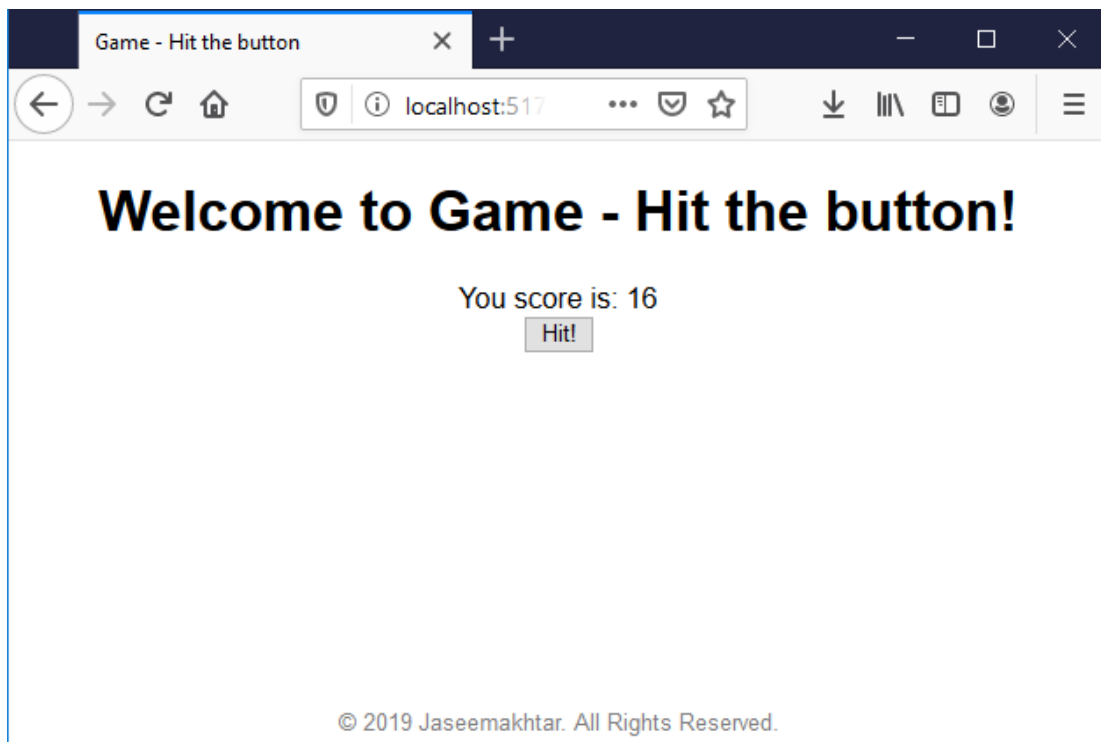
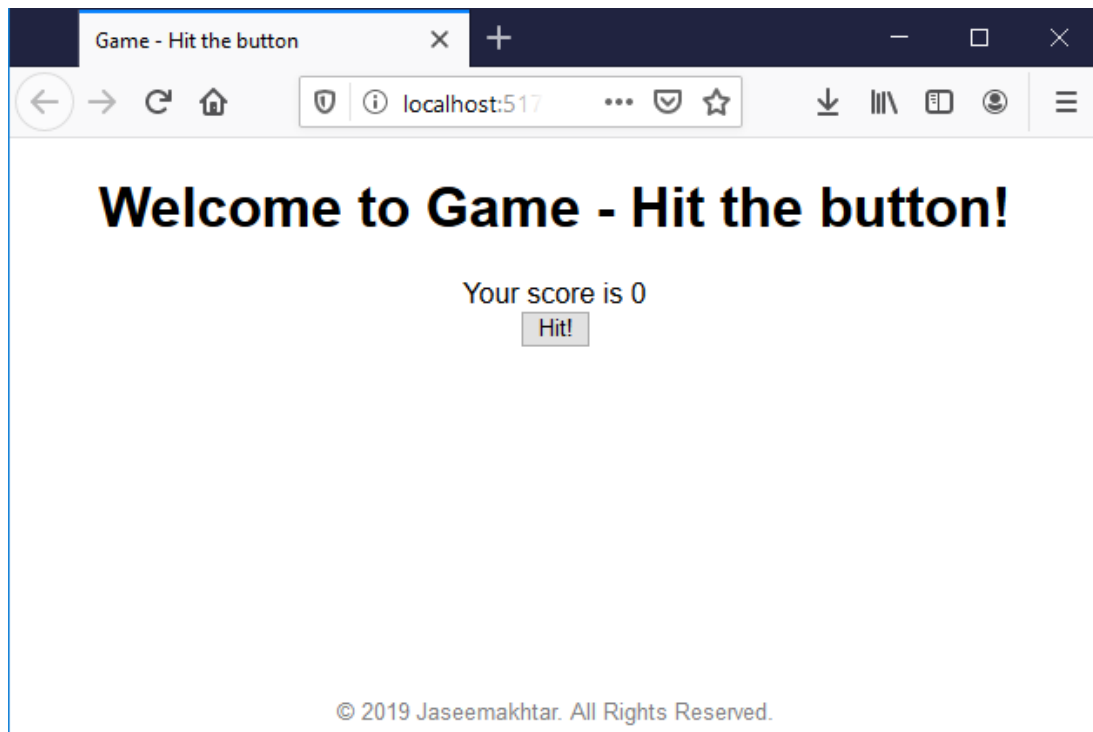
Default.aspx.cs

```
using System;
using System.Web;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void clickMeButton_Click(object sender, EventArgs e)
    {
        object value = ViewState["HitCount"];
        int i = (value == null) ? 1 : (int)value + 1;
        lblOutput.Text = string.Format("You score is: {0}", i);
        ViewState["HitCount"] = i;
    }
}
```

OUTPUT:

17. Work with forms using ASP.NET.

```
using System;
using System.Windows.Forms;

namespace ProgramTen
{
    public partial class Form1 : Form
    {
        string[] names;
        string[] passs;
        int rows;

        public Form1()
        {
            InitializeComponent();
            names = new string[10];
            passs = new string[10];

            names[0] = "Jaseem";
            names[1] = "Tony";
            names[2] = "Morgan";

            passs[0] = "sugarcane";
            passs[1] = "pepper";
            passs[2] = "burger";

            rows = 3;
        }

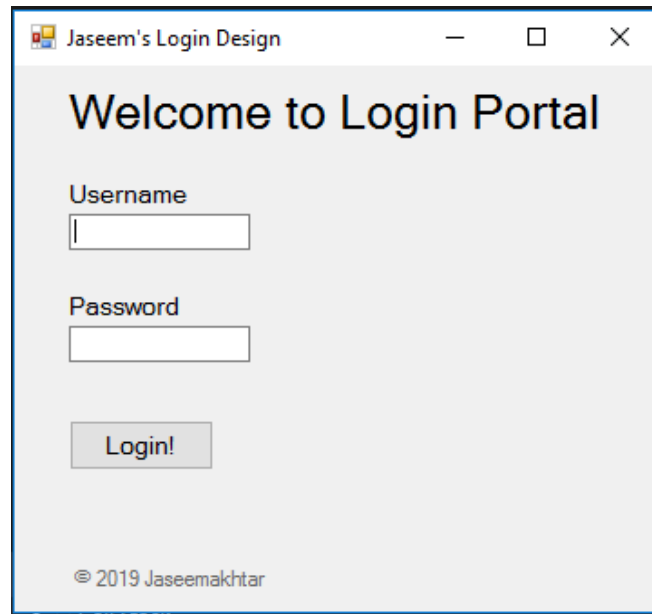
        private void btnLogin_Click(object sender, EventArgs e)
        {
            string username = txtName.Text.Trim();
            string password = txtPass.Text.Trim();

            if (username.Equals("") || password.Equals(""))
            {
                MessageBox.Show("Fields cannot be empty!");
                return;
            }

            for (int i = 0; i < rows; i++)
            {
                if (names[i].Equals(username) && passs[i].Equals(password))
                {

```

```
        MessageBox.Show("Login Successfull!");  
        return;  
    }  
}  
  
    MessageBox.Show("Incorrect username/password!");  
}  
  
}  
}
```

OUTPUT:

Jaseem's Login Design

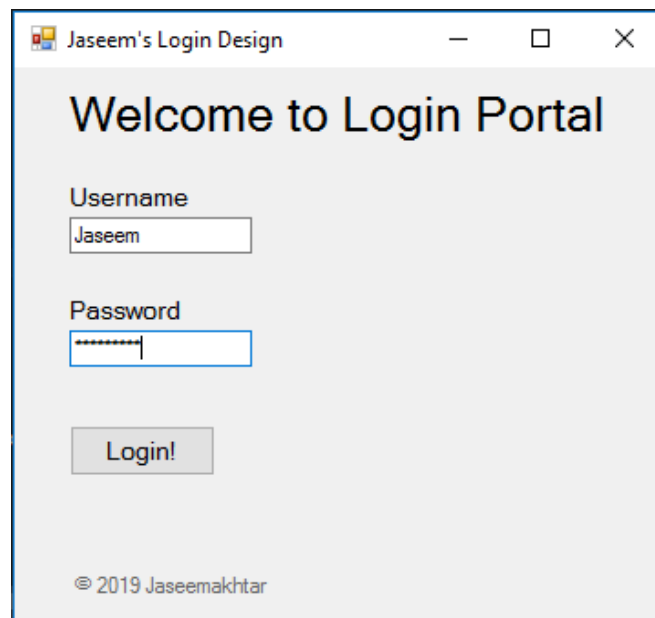
Welcome to Login Portal

Username

Password

Login!

© 2019 Jaseemakhtar



Jaseem's Login Design

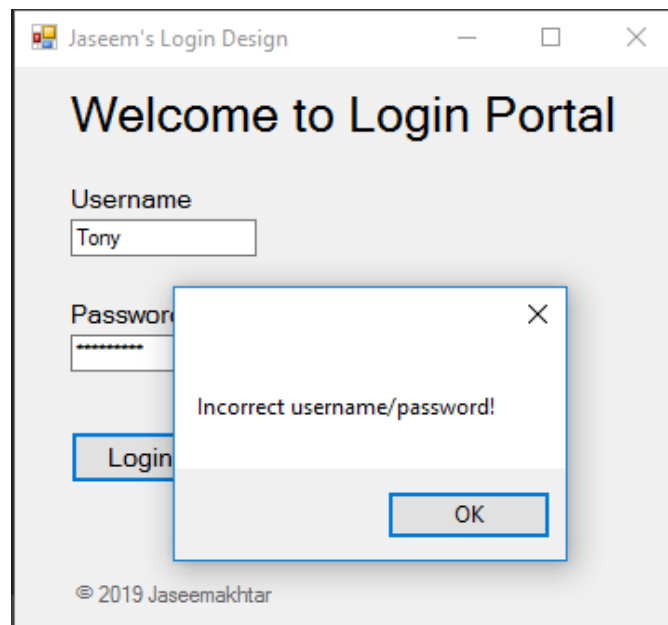
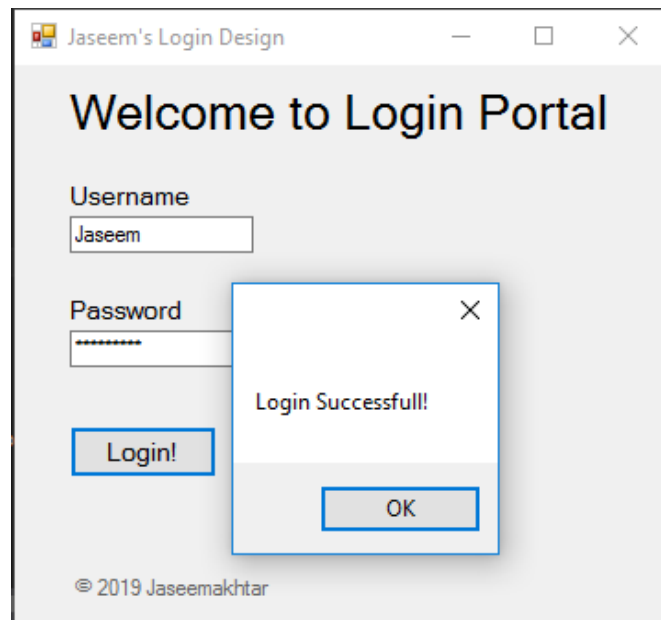
Welcome to Login Portal

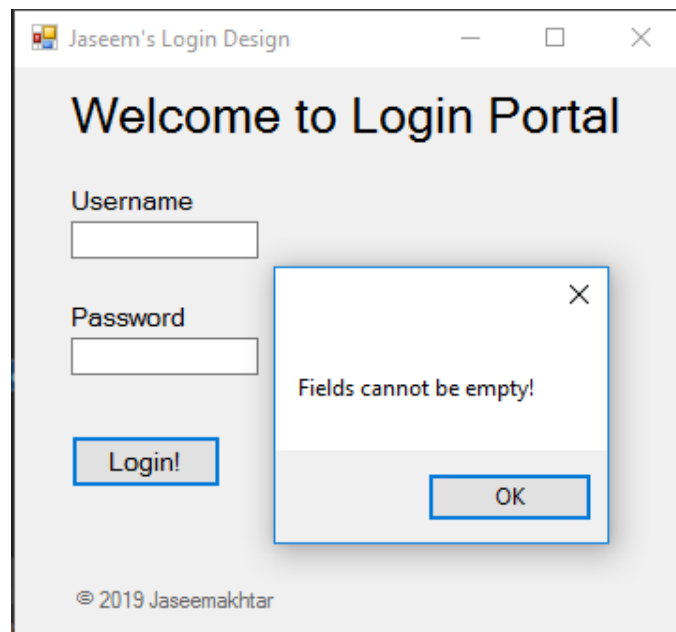
Username

Password

Login!

© 2019 Jaseemakhtar





18. Describe access data source through ADO.NET.User.cs

```
using System;

namespace ProgramEleven
{
    class User
    {
        public int Id
        {
            get;
            set;
        }
        public string UserName
        {
            get;
            set;
        }
        public string RollNumber
        {
            get;
            set;
        }
        public string Email
        {
            get;
            set;
        }
    }
}
```

UserAccessLayer.cs

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;

namespace ProgramEleven
{
    class UserAccessLayer
    {
        private List<User> users;
```

```
private string connectionString = "Data
Source=(LocalDB)\\v11.0;AttachDbFilename=\\\"C:\\Users\\Jaseem\\Documents\\Visual Studio
2013\\Projects\\ProgramEleven\\ProgramEleven\\myDatabase.mdf\\\";Integrated Security=True";
private SqlConnection connection;
private SqlCommand command;
private string query;

public List<User> getAllUsers()
{
    users = new List<User>();
    try
    {
        connection = new SqlConnection(connectionString);
        connection.Open();
        query = "SELECT * FROM _user";
        command = new SqlCommand(query, connection);
        SqlDataReader reader = command.ExecuteReader();
        while (reader.Read())
        {
            User user = new User();
            user.Id = Convert.ToInt16(reader.GetValue(0));
            user.UserName = reader.GetValue(1).ToString();
            user.Email = reader.GetValue(2).ToString();
            user.RollNumber = reader.GetValue(3).ToString();

            users.Add(user);
        }
    }
    catch (SqlException ex)
    {
        Console.WriteLine("Error in fetching database!: " + ex.Message);
    }

    return users;
}
}
```

Form1.cs

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Windows.Forms;
```

```
namespace ProgramEleven
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnFetch_Click(object sender, EventArgs e)
        {
            UserAccessLayer uAL = new UserAccessLayer();
            List<User> users = uAL.getAllUsers();
            if(users.Count == 0)
                lblStatus.Text = "No data!";
            else
                lblStatus.Text = "Data Fetched!";

            dGV.DataSource = users;
        }
    }
}
```

OUTPUT: