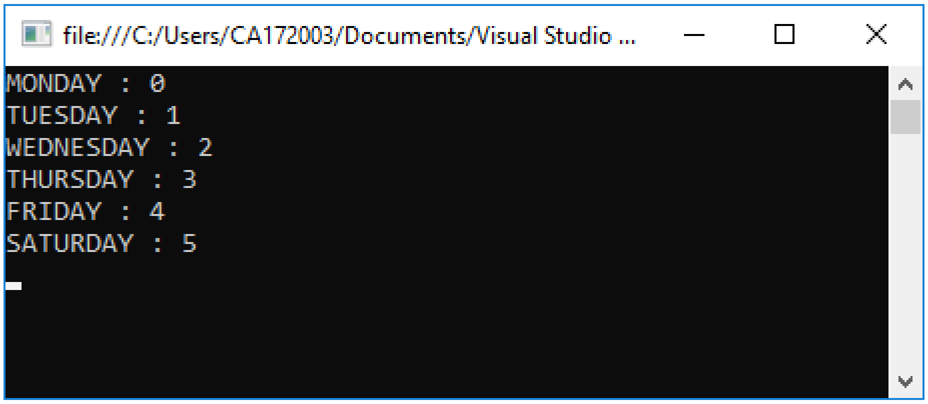
1. Describe the enumerations programming constructs, which provides a human-readable form of a series of related constant values in C#.

```
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:

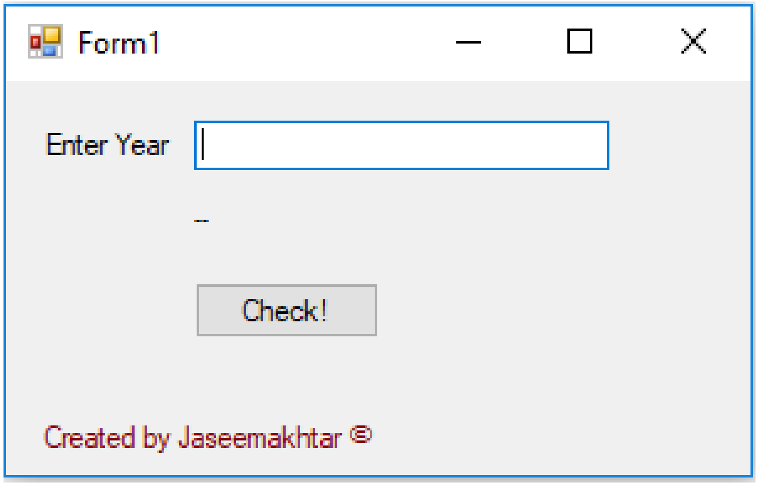


```
MONDAY : 0
TUESDAY : 1
WEDNESDAY : 2
THURSDAY : 3
FRIDAY : 4
SATURDAY : 5
```

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;
        }
    }
}
```

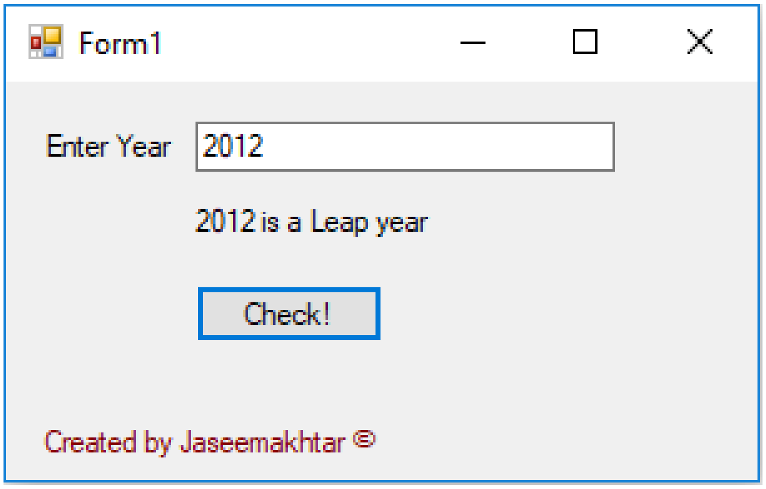
OUTPUT:



Enter Year

--

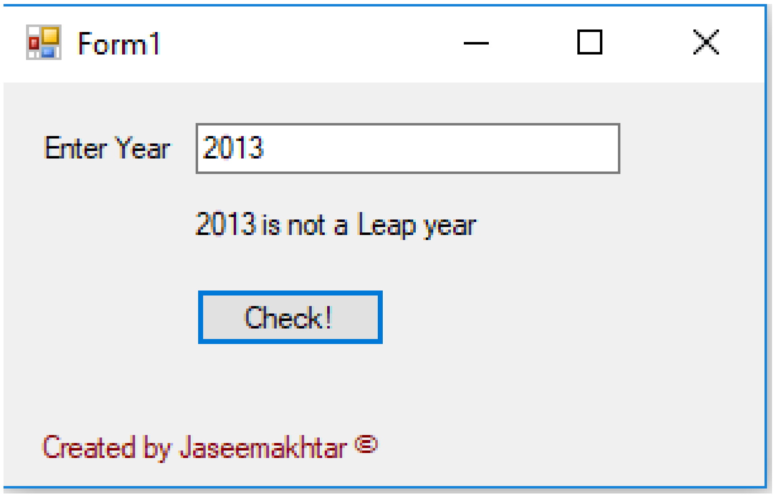
Created by Jaseemakhtar ©



Enter Year 2012

2012 is a Leap year

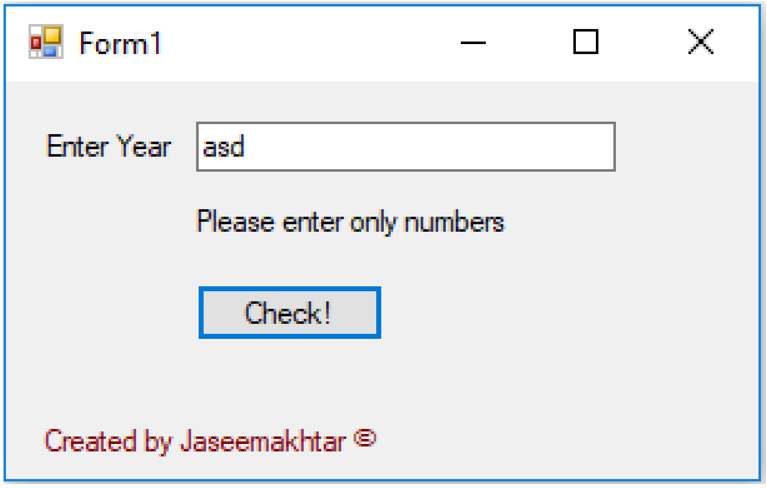
Created by Jaseemakhtar ©



Enter Year 2013

2013 is not a Leap year

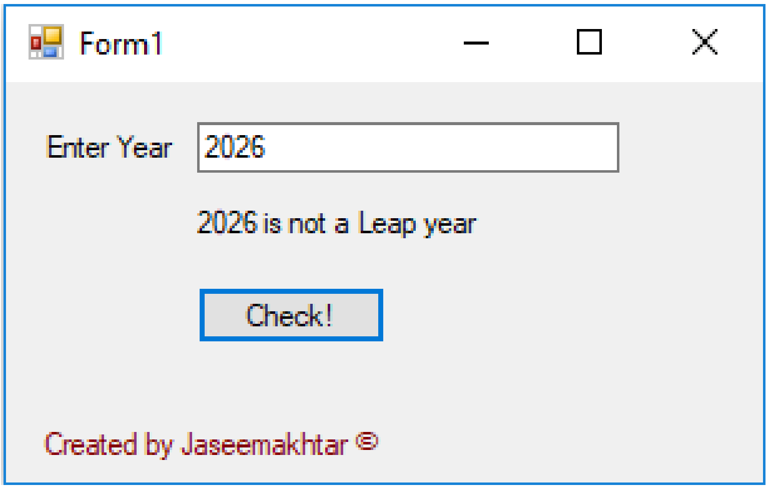
Created by Jaseemakhtar ©



Enter Year asd

Please enter only numbers

Created by Jaseemakhtar ©



Enter Year 2026

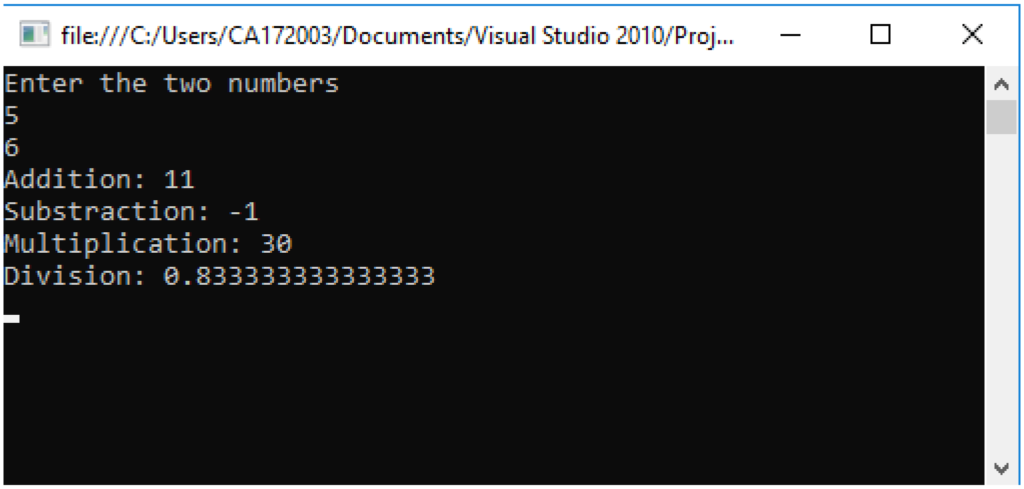
2026 is not a Leap year

Created by Jaseemakhtar ©

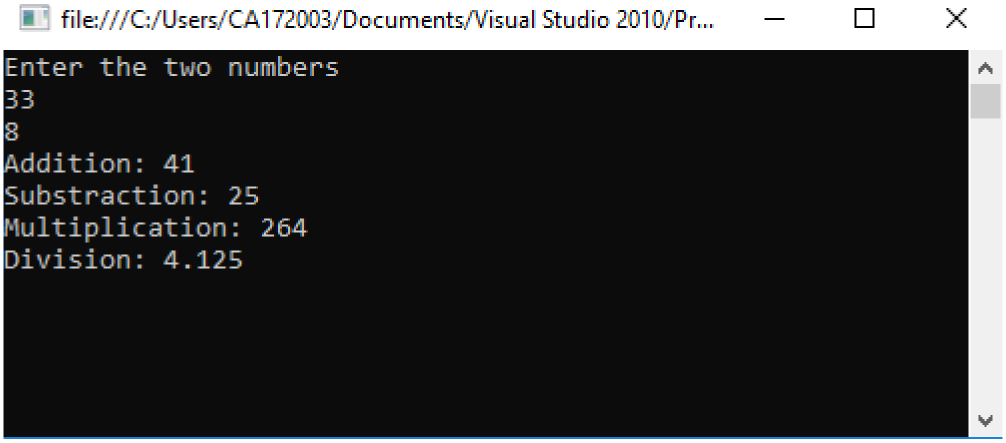
3. Program to display the addition, subtraction, multiplication and division of two number using console applications.

```
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();
        }
    }
}
```

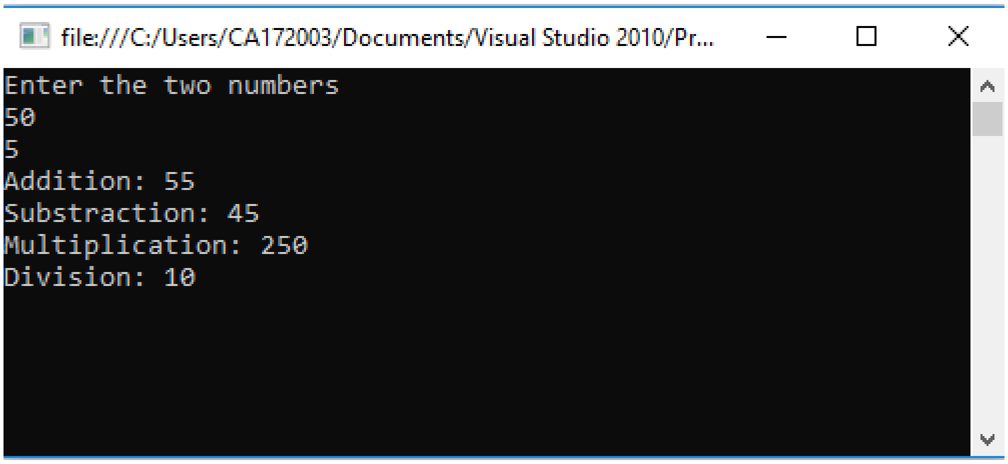
OUTPUT:


file:///C:/Users/CA172003/Documents/Visual Studio 2010/Proj...

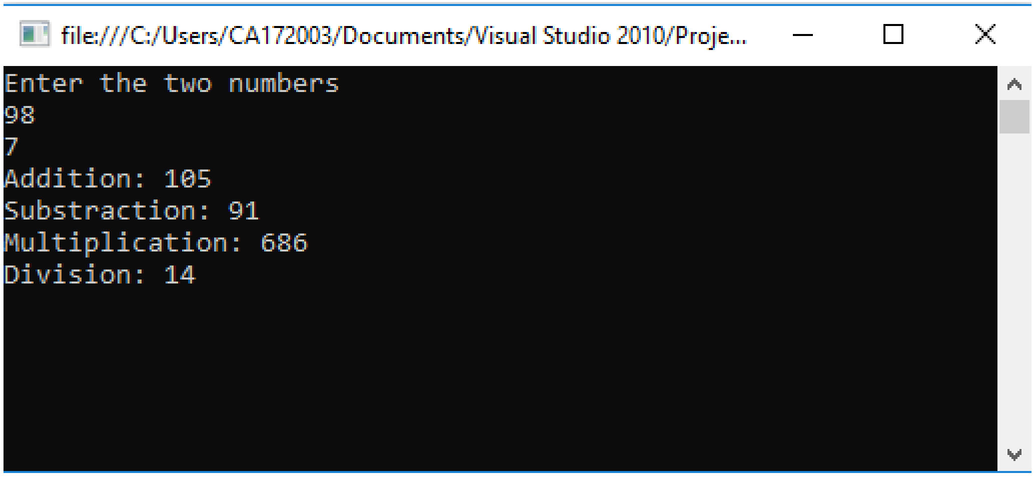
```
Enter the two numbers
5
6
Addition: 11
Substraction: -1
Multiplication: 30
Division: 0.833333333333333
```


file:///C:/Users/CA172003/Documents/Visual Studio 2010/Pr...

```
Enter the two numbers
33
8
Addition: 41
Substraction: 25
Multiplication: 264
Division: 4.125
```

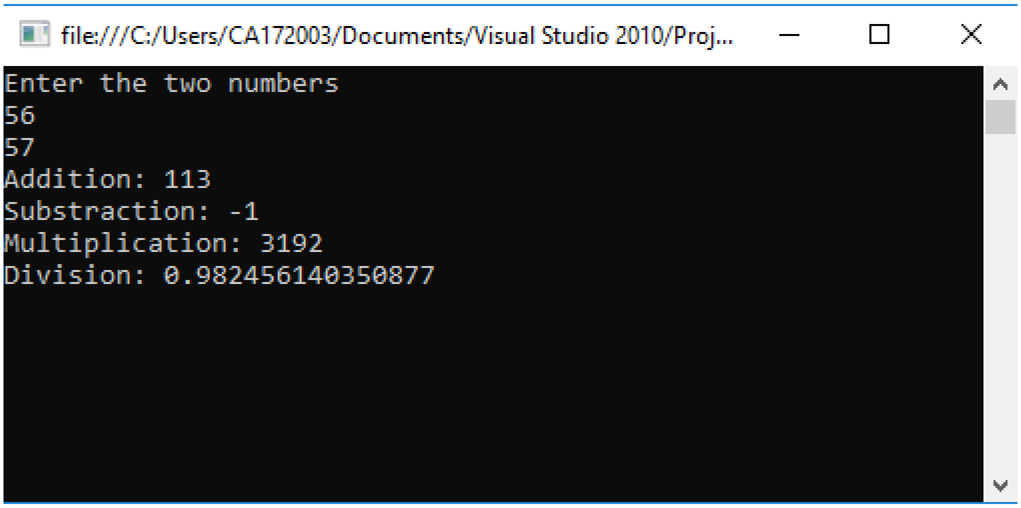

file:///C:/Users/CA172003/Documents/Visual Studio 2010/Pr...

```
Enter the two numbers
50
5
Addition: 55
Substraction: 45
Multiplication: 250
Division: 10
```



file:///C:/Users/CA172003/Documents/Visual Studio 2010/Proje...

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



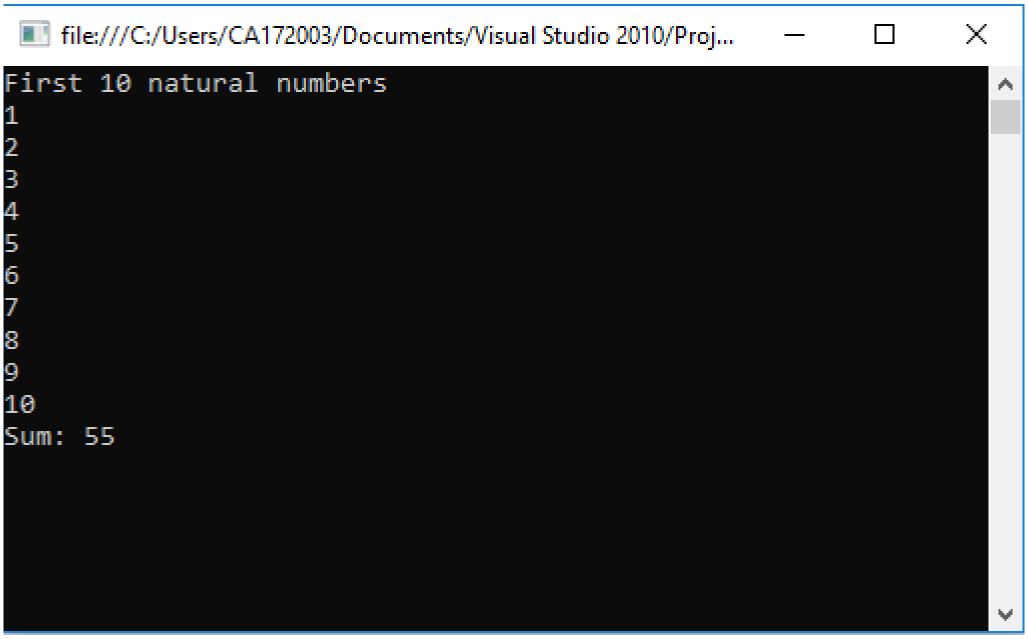
file:///C:/Users/CA172003/Documents/Visual Studio 2010/Proj...

```
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:

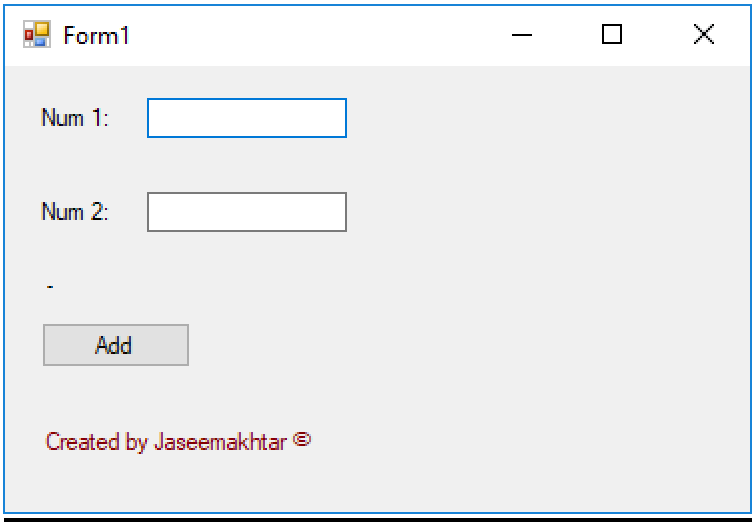

file:///C:/Users/CA172003/Documents/Visual Studio 2010/Proj...

```
First 10 natural numbers
1
2
3
4
5
6
7
8
9
10
Sum: 55
```

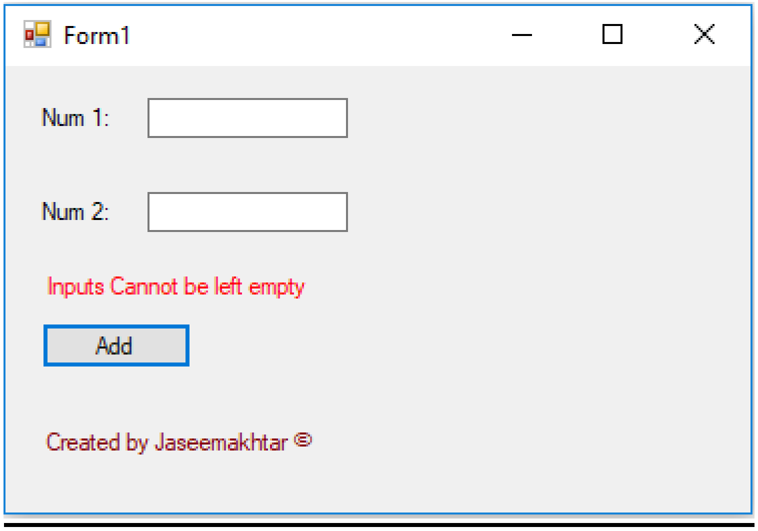
5. Program to display the addition using the windows application.

```
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;
        }
    }
}
```

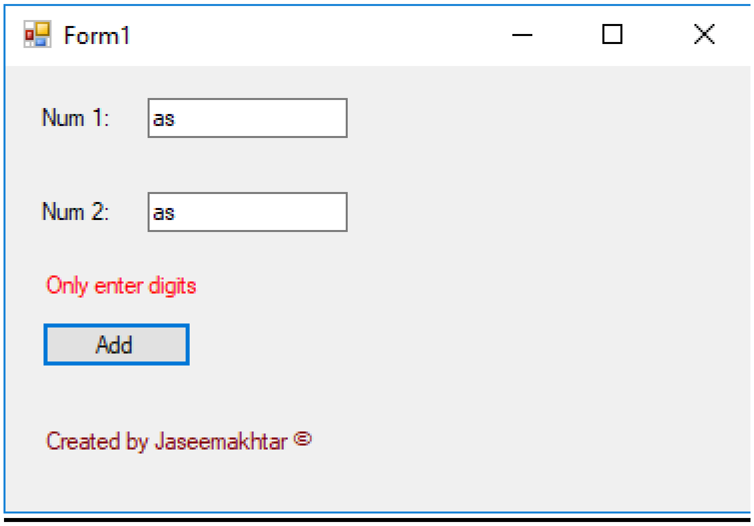
OUTPUT:



-



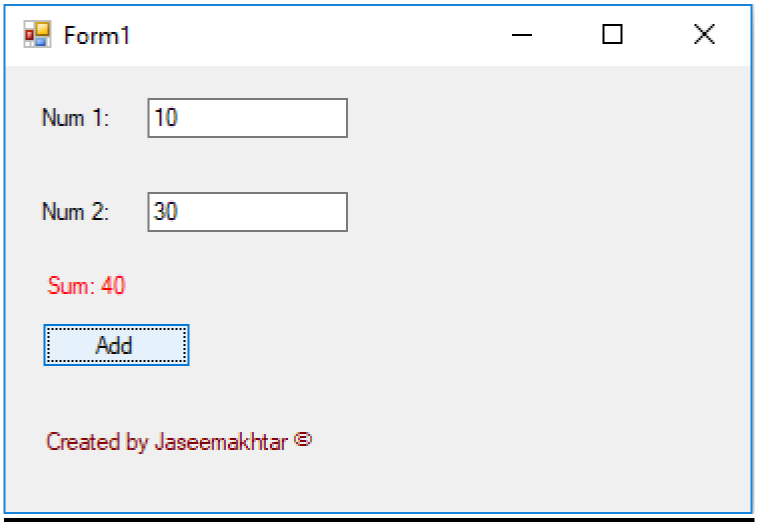
Inputs Cannot be left empty



Num 1: as

Num 2: as

Only enter digits

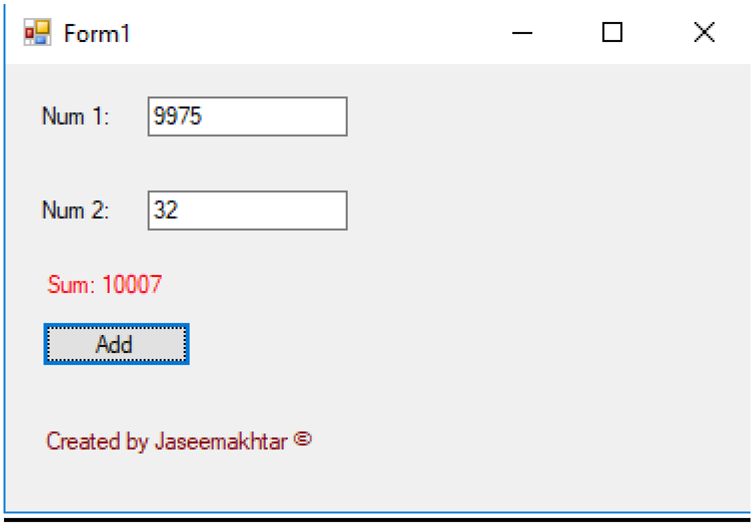


Num 1: 10

Num 2: 30

Sum: 40

Created by Jaseemakhtar ©



Num 1: 9975

Num 2: 32

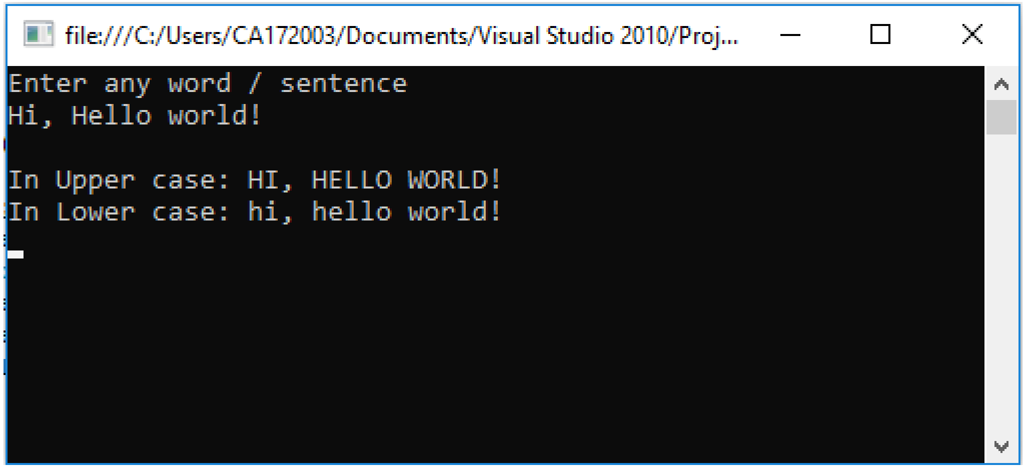
Sum: 10007

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:

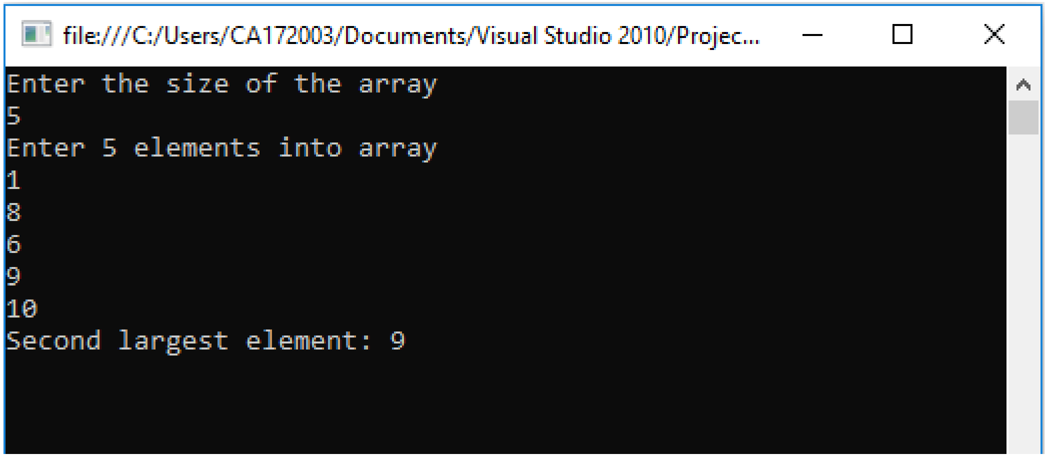

file:///C:/Users/CA172003/Documents/Visual Studio 2010/Proj...

```
Enter any word / sentence
Hi, Hello world!

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

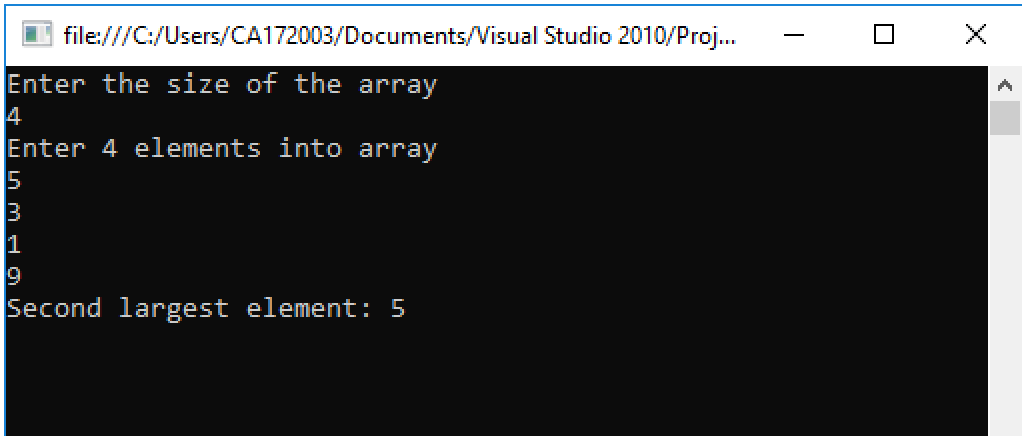
7. Find the second largest element in a single dimensional array.

```
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();
        }
    }
}
```

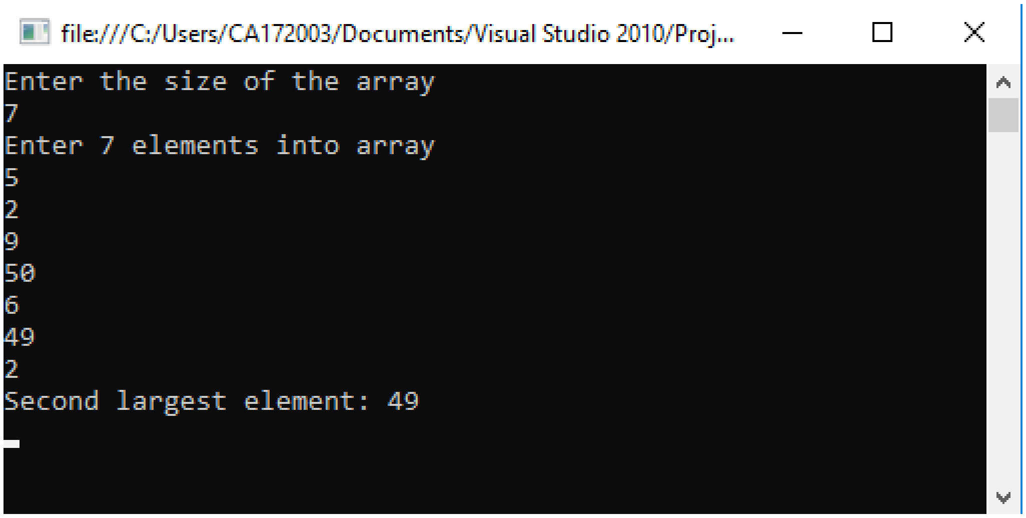
OUTPUT:


file:///C:/Users/CA172003/Documents/Visual Studio 2010/Projec...

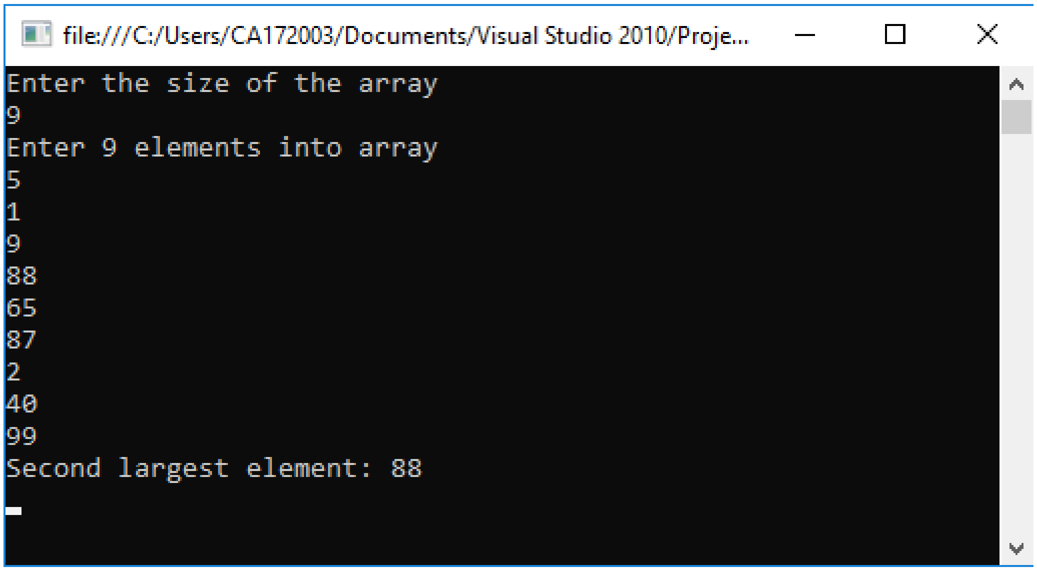
```
Enter the size of the array
5
Enter 5 elements into array
1
8
6
9
10
Second largest element: 9
```


file:///C:/Users/CA172003/Documents/Visual Studio 2010/Proj...

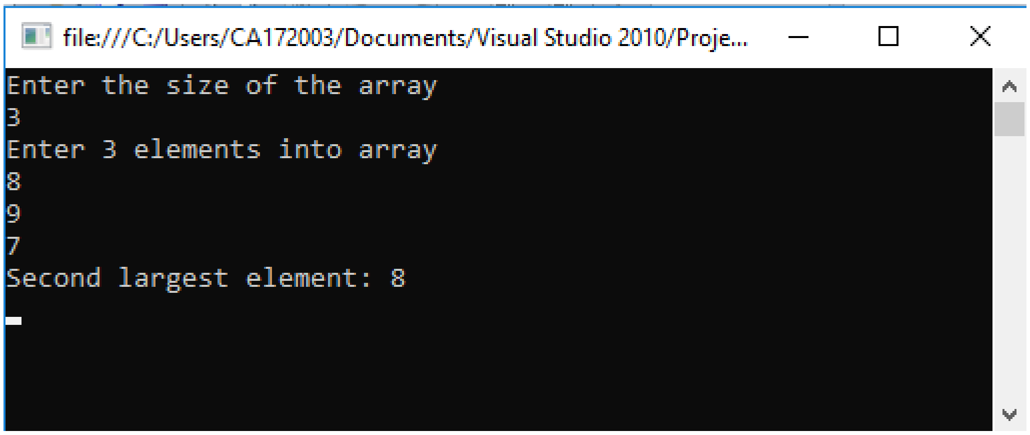
```
Enter the size of the array
4
Enter 4 elements into array
5
3
1
9
Second largest element: 5
```


file:///C:/Users/CA172003/Documents/Visual Studio 2010/Proj...

```
Enter the size of the array
7
Enter 7 elements into array
5
2
9
50
6
49
2
Second largest element: 49
```


file:///C:/Users/CA172003/Documents/Visual Studio 2010/Proje...

```
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
```

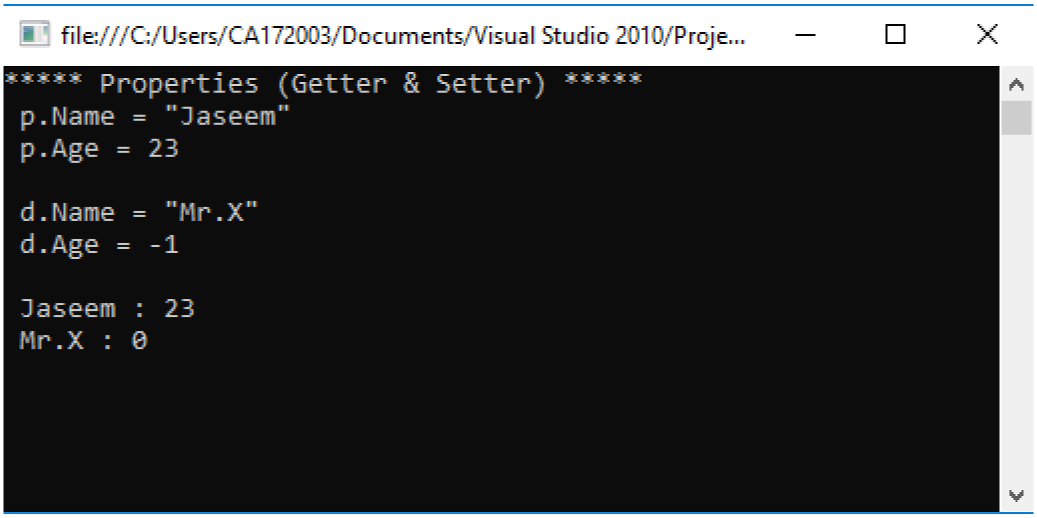

file:///C:/Users/CA172003/Documents/Visual Studio 2010/Proje...

```
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:



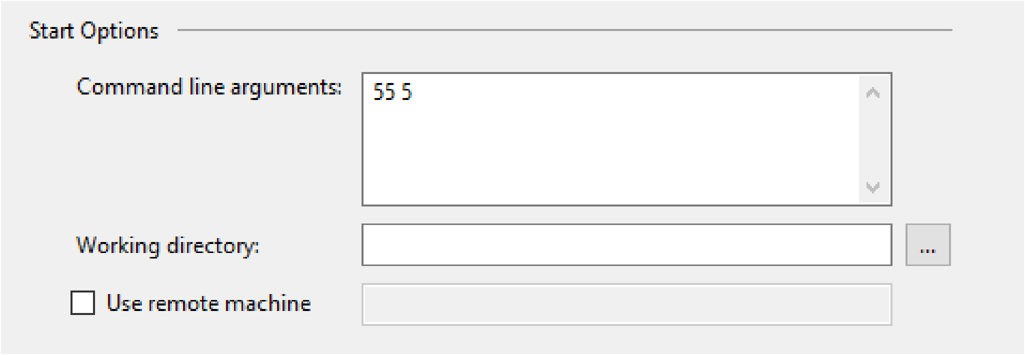
file:///C:/Users/CA172003/Documents/Visual Studio 2010/Proje...

```
***** 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:

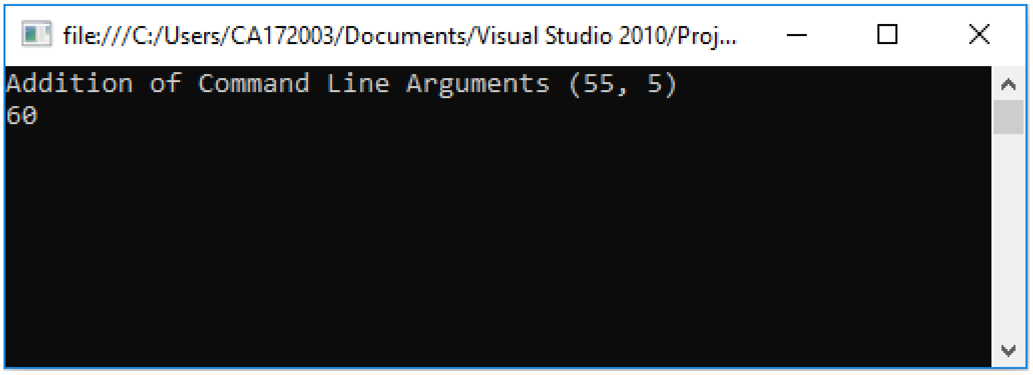


Start Options

Command line arguments: 55 5

Working directory:

□ Use remote machine



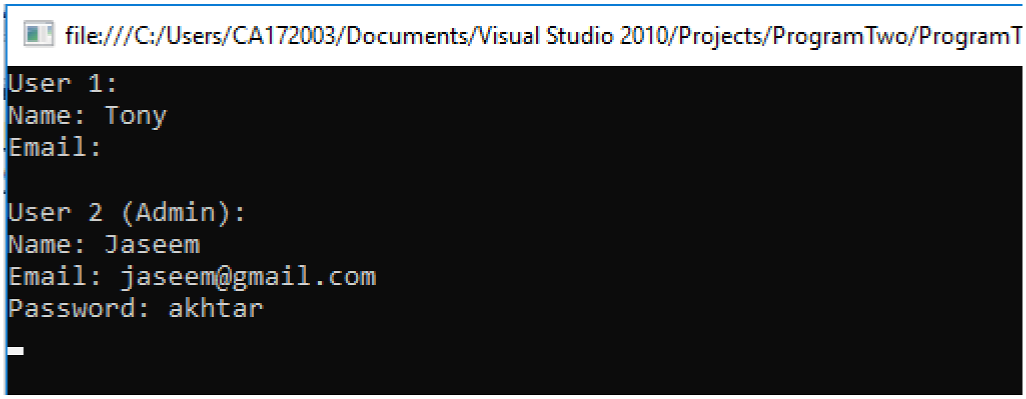
file:///C:/Users/CA172003/Documents/Visual Studio 2010/Proj...

```
Addition of Command Line Arguments (55, 5)
60
```

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:



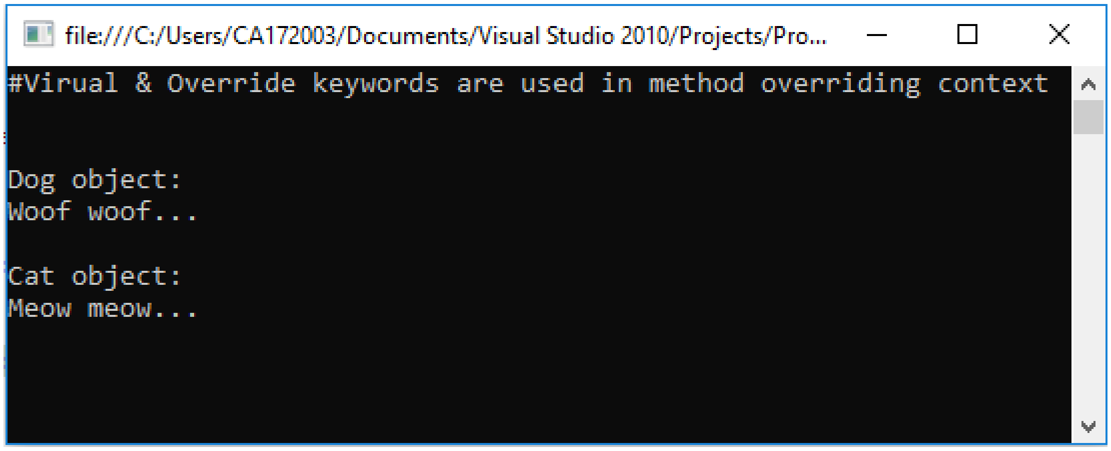
file:///C:/Users/CA172003/Documents/Visual Studio 2010/Projects/ProgramTwo/ProgramT

```
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("#Virual & 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:

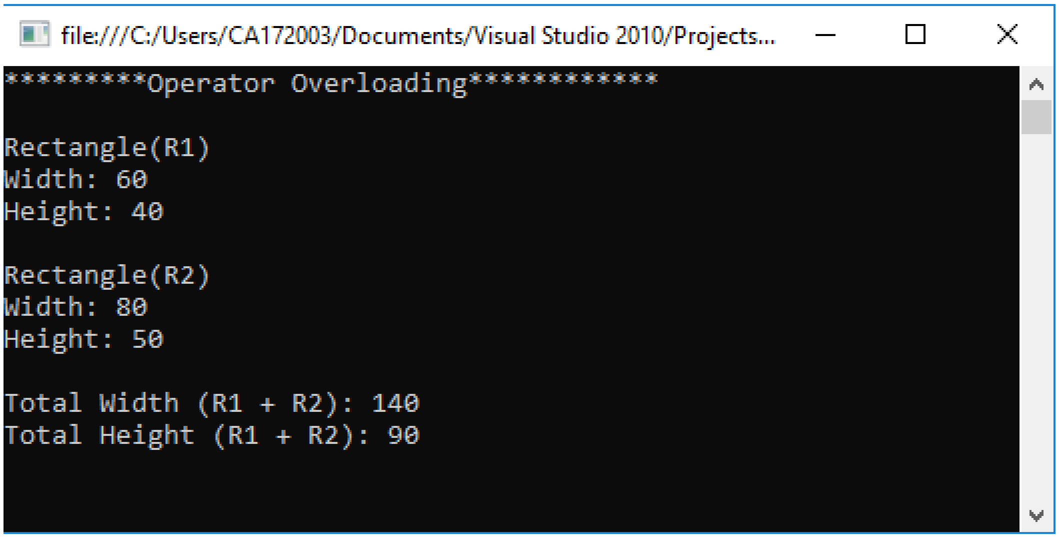

file:///C:/Users/CA172003/Documents/Visual Studio 2010/Projects/Pro...

```
#Virual & 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
```