

70-483 : Programming in C#

- Arrays and Lists -



Zouhair Rimale, Ph.D.

Expert technique .NET | SharePoint | ASP.NET MVC | WEB API | Angular

Arrays



Zouhair Rimale, Ph.D.

Expert technique .NET | SharePoint | ASP.NET MVC | WEB API | Angular

In this lecture...

- Quick review of arrays
- Types of Arrays in C#
- Array methods



Zouhair Rimale, Ph.D.

Expert technique .NET | SharePoint | ASP.NET MVC | WEB API | Angular

Array

Represents a fixed number of variables of a particular type.



Zouhair Rimale, Ph.D.

Expert technique .NET | SharePoint | ASP.NET MVC | WEB API | Angular

Types of Arrays

Single Dimension

0	1	2	3	4
---	---	---	---	---

Multi Dimension

0	1	2	3	4
0	1	2	3	4
0	1	2	3	4



Zouhair Rimale, Ph.D.

Expert technique .NET | SharePoint | ASP.NET MVC | WEB API | Angular

Single Dimension Arrays

```
var numbers = new int[5];
```

```
var numbers = new int[5]{ 1, 2, 3, 4, 5 };
```



Zouhair Rimale, Ph.D.

Expert technique .NET | SharePoint | ASP.NET MVC | WEB API | Angular

Multi Dimension Arrays

Rectangular

3x5

0	1	2	3	4
0	1	2	3	4
0	1	2	3	4

Jagged

0	1	2	3	
0	1	2	3	4
0	1	2		



Zouhair Rimale, Ph.D.

Expert technique .NET | SharePoint | ASP.NET MVC | WEB API | Angular

Syntax (Rectangular 2D)

```
var matrix = new int[3, 5];  
var matrix = new int[3, 5]  
{  
    { 1, 2, 3, 4, 5 },  
    { 6, 7, 8, 9, 10 },  
    { 11, 12, 13, 14, 15 }  
};  
  
var element = matrix[0, 0];
```



Zouhair Rimale, Ph.D.

Expert technique .NET | SharePoint | ASP.NET MVC | WEB API | Angular

Syntax (Rectangular 3D)

```
var colors = new int[3, 5, 4];
```



Zouhair Rimale, Ph.D.

Expert technique .NET | SharePoint | ASP.NET MVC | WEB API | Angular

Syntax (Jagged)

Jagged

0	1	2	3	
0	1	2	3	4
0	1	2		

```
var array = new int[3][];
```

```
▶ array[0] = new int[4];  
array[1] = new int[5];  
array[2] = new int[3];
```

```
array[0][0] = 1;
```



Zouhair Rimale, Ph.D.

Expert technique .NET | SharePoint | ASP.NET MVC | WEB API | Angular

Jagged

```
var array = new int[3][];
```

Rectangular

```
var array = new int[3, 5];
```



Zouhair Rimale, Ph.D.

Expert technique .NET | SharePoint | ASP.NET MVC | WEB API | Angular

Array
Length
Clear() Copy() IndexOf() Reverse() Sort()



Zouhair Rimale, Ph.D.

Expert technique .NET | SharePoint | ASP.NET MVC | WEB API | Angular

Demo Arrays



Zouhair Rimale, Ph.D.

Expert technique .NET | SharePoint | ASP.NET MVC | WEB API | Angular

Program.cs

CSharpFundamentals.Program

Main(string[] args)

```

using System;

namespace CSharpFundamentals
{
    class Program
    {
        static void Main(string[] args)
        {
            var numbers = new[] { 3, 7, 9, 2, 14, 6 };

            // Length
            Console.WriteLine("Length: " + numbers.Length);

            // IndexOf()
            var index = Array.IndexOf(numbers, 9);
            Console.WriteLine("Index of 9: " + index);

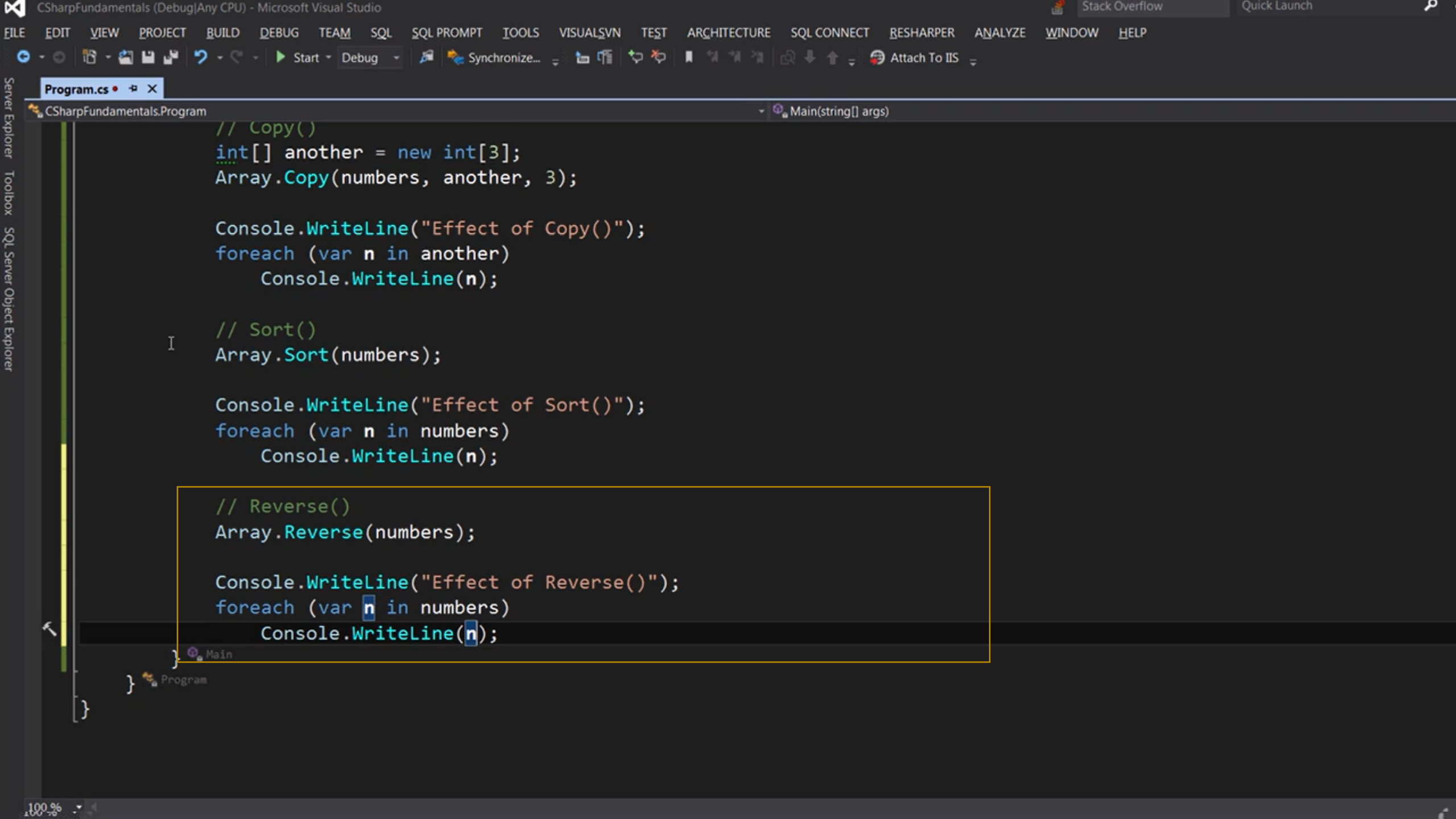
            // Clear()
            Array.Clear(numbers, 0, 2);

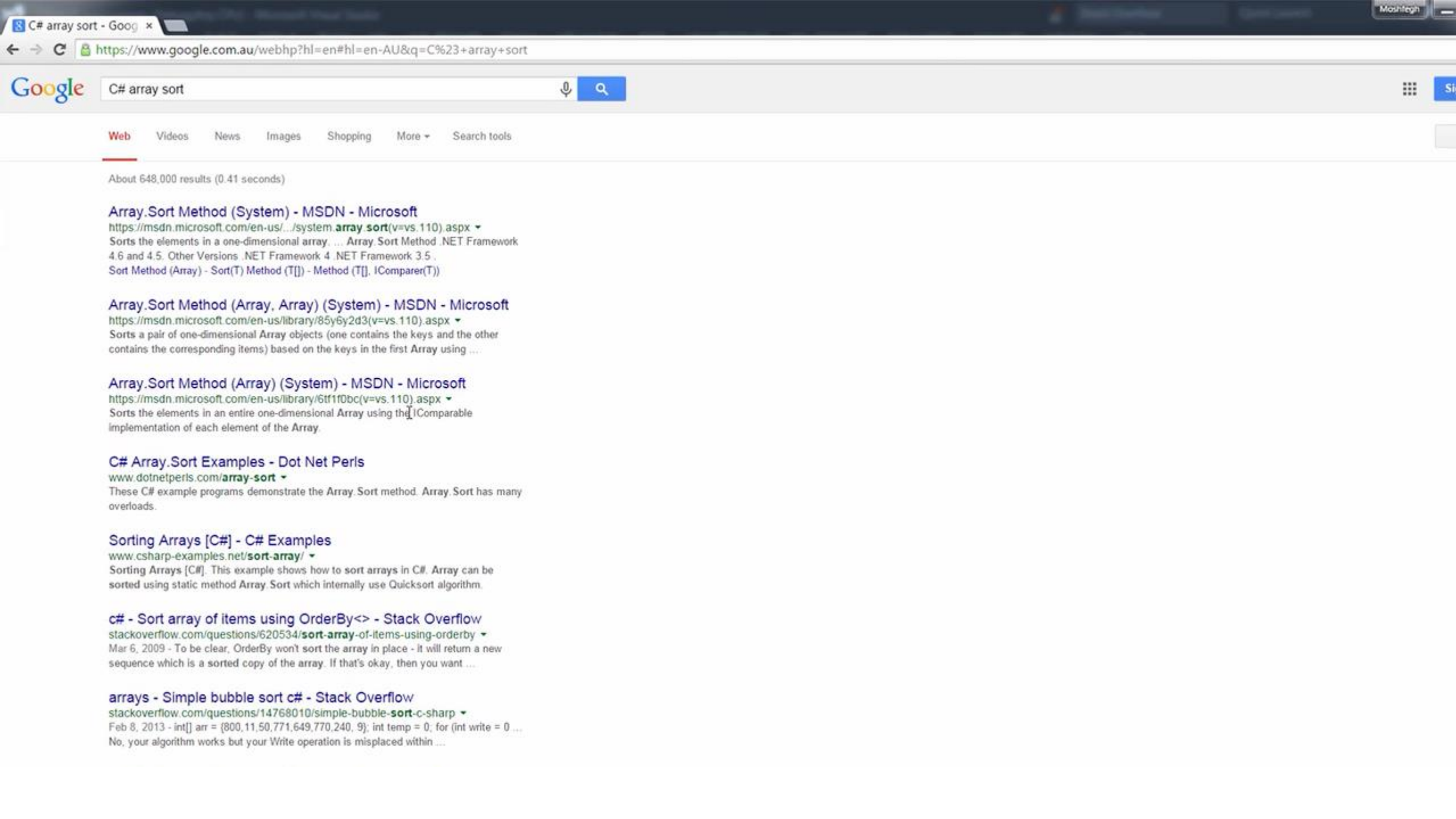
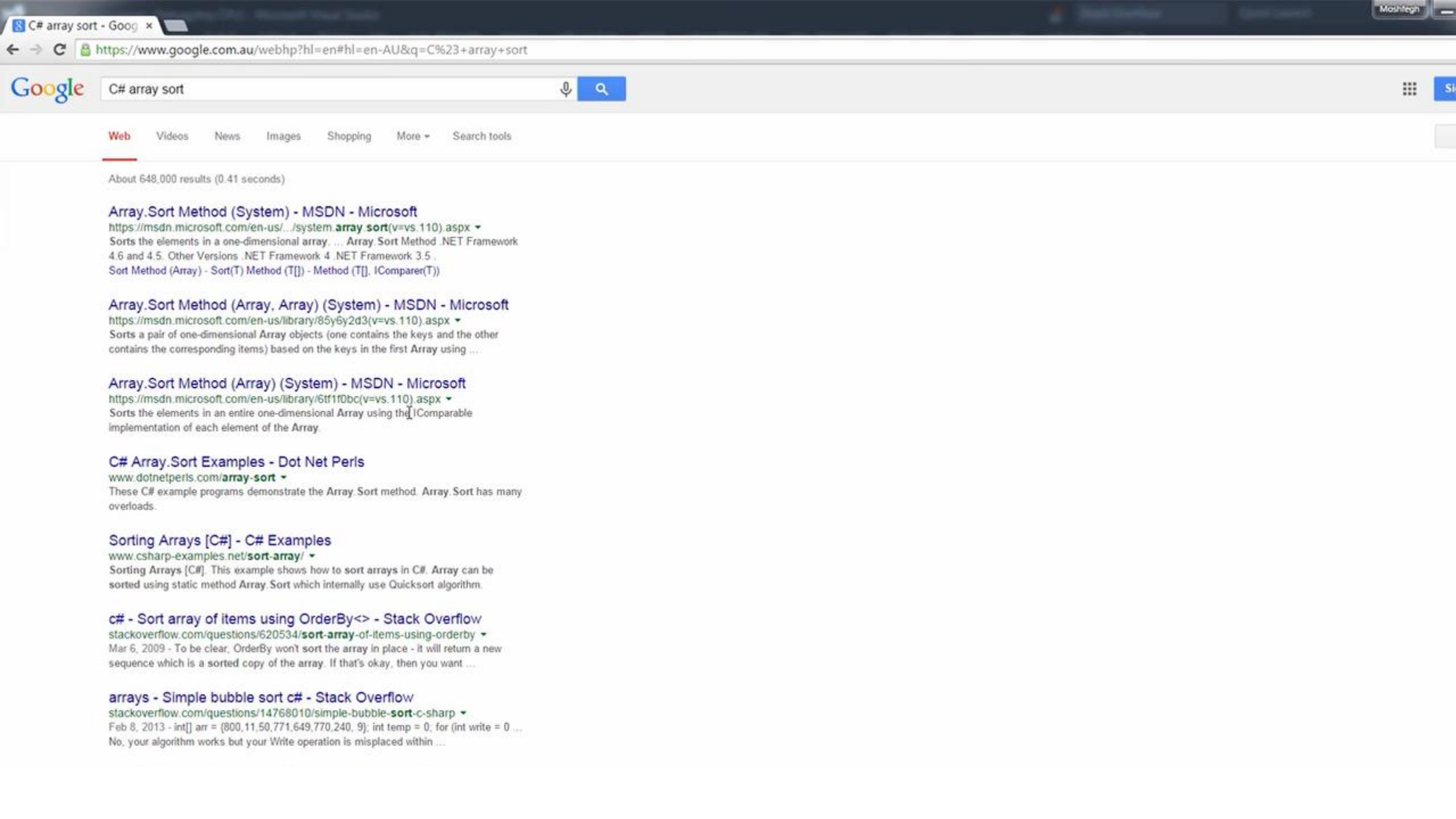
            Console.WriteLine("Effect of Clear()");
            foreach (var n in numbers)
                Console.WriteLine(n);
        }
    }
}
    
```

```

C:\Windows\system32\cmd.exe

Length: 6
Index of 9: 2
Effect of Clear()
0
0
9
2
14
6
Press any key to continue . . .
    
```





















- MSDN Library
- .NET Development
- .NET Framework 4.6 RC and 4.5
- .NET Framework Class Library
- System
- Array Class
- Array Methods

Sort Method

- Sort(T) Method (T[])
- Sort Method (Array)
- Sort(T) Method (T[], IComparer(T))
- Sort(T) Method (T[], Comparison(T))
- Sort(TKey, TValue) Method (TKey[], TValue[])
- Sort Method (Array, Array)
- Sort Method (Array, IComparer)
- Sort(T) Method (T[], Int32, Int32)
- Sort(TKey, TValue) Method (TKey[], TValue[], IComparer(TKey))
- Sort Method (Array, Array, IComparer)
- Sort Method (Array, Int32, Int32)
- Sort(T) Method (T[], Int32, Int32, IComparer(T))
- Sort(TKey, TValue) Method (TKey[], TValue[], Int32, Int32)
- Sort Method (Array, Array, Int32, Int32)
- Sort Method (Array, Int32, Int32, IComparer)
- Sort(TKey, TValue) Method (TKey[], TValue[], Int32, Int32, IComparer(TKey))

Overload List

	Name	Description
	Sort<T>(T[])	Sorts the elements in an entire Array using the IComparable<T> generic interface implementation of each element of the Array .
	Sort(Array)	Sorts the elements in an entire one-dimensional Array using the IComparable implementation of each element of the Array .
	Sort<T>(T[], IComparer<T>)	Sorts the elements in an Array using the specified IComparer<T> generic interface.
	Sort<T>(T[], Comparison<T>)	Sorts the elements in an Array using the specified Comparison<T> .
	Sort<TKey, TValue>(TKey[], TValue[])	Sorts a pair of Array objects (one contains the keys and the other contains the corresponding items) based on the keys in the first Array using the IComparable<T> generic interface implementation of each key.
	Sort(Array, Array)	Sorts a pair of one-dimensional Array objects (one contains the keys and the other contains the corresponding items) based on the keys in the first Array using the IComparable implementation of each key.
	Sort(Array, IComparer)	Sorts the elements in a one-dimensional Array using the specified IComparer .
	Sort<T>(T[], Int32, Int32)	Sorts the elements in a range of elements in an Array using the IComparable<T> generic interface implementation of each element of the Array .
	Sort<TKey, TValue>(TKey[], TValue[], IComparer<TKey>)	Sorts a pair of Array objects (one contains the keys and the other contains the corresponding items) based on the keys in the first Array using the specified IComparer<T> generic interface implementation of each key.
	Sort(Array, Array, IComparer)	Sorts a pair of one-dimensional Array objects (one contains the keys and the other contains the corresponding items) based on the keys in the first Array using the specified IComparer .
	Sort(Array, Int32, Int32)	Sorts the elements in a range of elements in a one-dimensional Array using the IComparable implementation of each element of the Array .
	Sort<T>(T[], Int32, Int32, IComparer<T>)	Sorts the elements in a range of elements in an Array using the specified IComparer<T> generic interface.
	Sort<TKey, TValue>(TKey[], TValue[], Int32, Int32)	Sorts a range of elements in a pair of Array objects (one contains the keys and the other contains the corresponding items) based on the keys in the first Array using the IComparable generic interface implementation of each key.
	Sort(Array, Array, Int32, Int32)	Sorts a range of elements in a pair of one-dimensional Array objects (one contains the keys and the other contains the corresponding items) based on the keys in the first Array using the IComparable implementation of each key.
	Sort(Array, Int32, Int32, IComparer)	Sorts the elements in a range of elements in a one-dimensional Array using the specified IComparer .
	Sort<TKey, TValue>(TKey[], TValue[], Int32, Int32, IComparer<TKey>)	Sorts a range of elements in a pair of Array objects (one contains the keys and the other contains the corresponding items) based on the keys in the first Array using the specified IComparer<T> generic interface.

Lists



Zouhair Rimale, Ph.D.

Expert technique .NET | SharePoint | ASP.NET MVC | WEB API | Angular

Arrays vs Lists

- Array: fixed size
- List: dynamic size



Zouhair Rimale, Ph.D.

Expert technique .NET | SharePoint | ASP.NET MVC | WEB API | Angular

Creating Lists

```
var numbers = new List<int>();
```

```
var numbers = new List<int>() { 1, 2, 3, 4 };
```



Zouhair Rimale, Ph.D.

Expert technique .NET | SharePoint | ASP.NET MVC | WEB API | Angular

Useful Methods

- Add()
- AddRange()
- Remove()
- RemoveAt()
- IndexOf()
- Contains()
- Count



Zouhair Rimale, Ph.D.

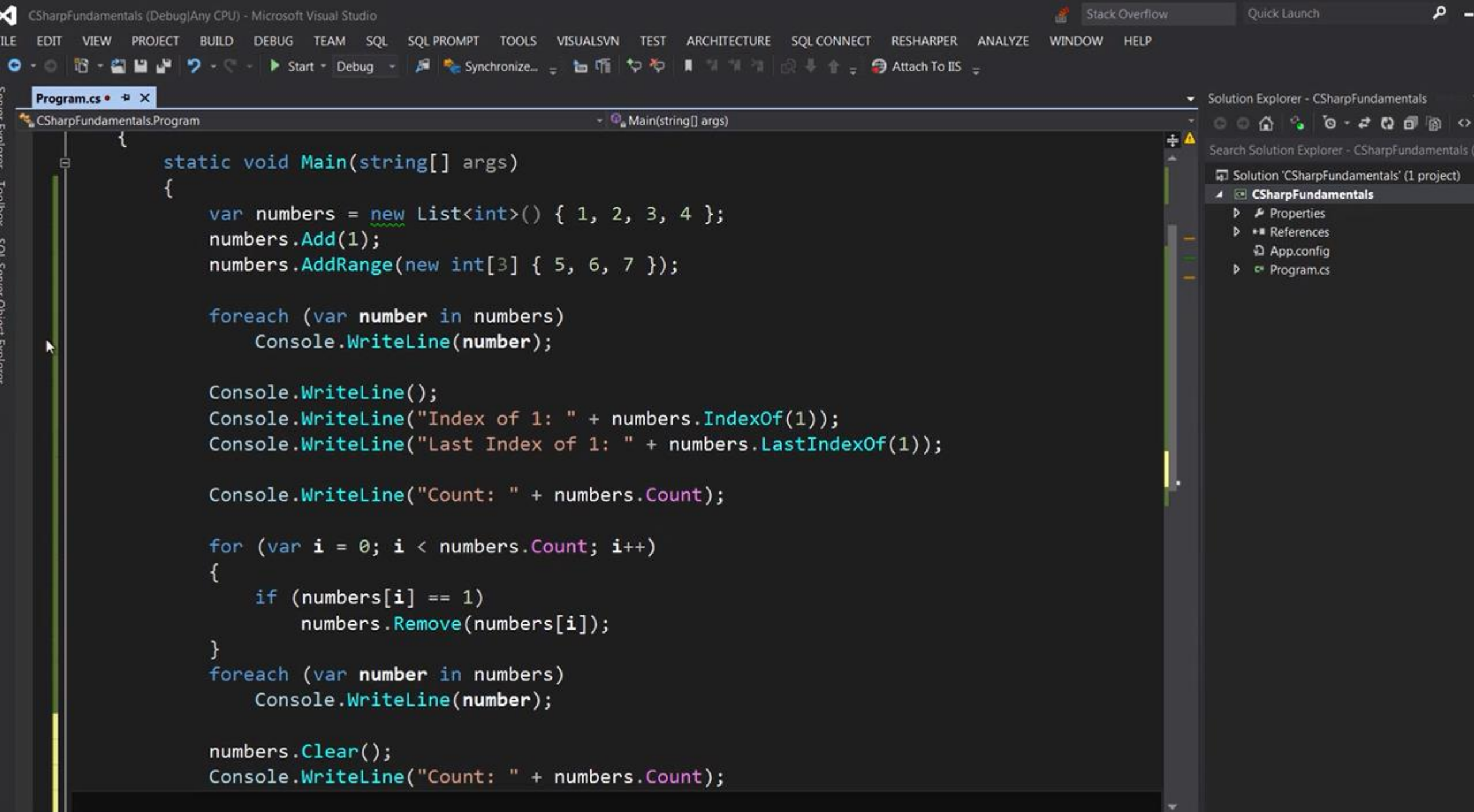
Expert technique .NET | SharePoint | ASP.NET MVC | WEB API | Angular

Demo Lists



Zouhair Rimale, Ph.D.

Expert technique .NET | SharePoint | ASP.NET MVC | WEB API | Angular



```
static void Main(string[] args)
{
    var numbers = new List<int>() { 1, 2, 3, 4 };
    numbers.Add(1);
    numbers.AddRange(new int[3] { 5, 6, 7 });

    foreach (var number in numbers)
        Console.WriteLine(number);

    Console.WriteLine();
    Console.WriteLine("Index of 1: " + numbers.IndexOf(1));
    Console.WriteLine("Last Index of 1: " + numbers.LastIndexOf(1));

    Console.WriteLine("Count: " + numbers.Count);

    for (var i = 0; i < numbers.Count; i++)
    {
        if (numbers[i] == 1)
            numbers.Remove(numbers[i]);
    }
    foreach (var number in numbers)
        Console.WriteLine(number);

    numbers.Clear();
    Console.WriteLine("Count: " + numbers.Count);
}
```

Program.cs

CSharpFundamentals.Program

Main(string[] args)

```
{
    static void Main(string[] args)
    {
        var numbers = new List<int>() { 1, 2, 3, 4 };
        numbers.Add(1);
        numbers.AddRange(new int[3] { 5, 6, 7 });

        foreach (var number in numbers)
            Console.WriteLine(number);

        Console.WriteLine();
        Console.WriteLine("Index of 1: " + numbers.IndexOf(1));
        Console.WriteLine("Last Index of 1: " + numbers.LastIndexOf(1));

        Console.WriteLine("Count: " + numbers.Count);

        for (var i = 0; i < numbers.Count; i++)
        {
            if (numbers[i] == 1)
                numbers.Remove(numbers[i]);
        }

        foreach (var number in numbers)
            Console.WriteLine(number);
    }
}
```

Solution Explorer - CSharpFundamentals

Search Solution Explorer - CSharpFundamentals (C#)

Solution 'CSharpFundamentals' (1 project)

CSharpFundamentals

Properties

References

App.config

Program.cs

```
Unhandled Exception: System.InvalidOperationException: Collection was modified;
enumeration operation may not execute.
   at System.ThrowHelper.ThrowInvalidOperationException(ExceptionResource resource)
   at System.Collections.Generic.List`1.Enumerator.MoveNextRare()
   at System.Collections.Generic.List`1.Enumerator.MoveNext()
   at CSharpFundamentals.Program.Main(String[] args) in c:\Projects\CSharpFundamentals\CSharpFundamentals\CSharpFundamentals\Program.cs:line 23
```


Exercise



Zouhair Rimale, Ph.D.

Expert technique .NET | SharePoint | ASP.NET MVC | WEB API | Angular

Exercise - 1

When you post a message on Facebook, depending on the number of people who like your post, Facebook displays different information.

- If no one likes your post, it doesn't display anything.
- If only one person likes your post, it displays: [Friend's Name] likes your post.
- If two people like your post, it displays: [Friend 1] and [Friend 2] like your post.
- If more than two people like your post, it displays: [Friend 1], [Friend 2] and [Number of Other People] others like your post.

Write a program and continuously ask the user to enter different names, until the user presses Enter (without supplying a name).

Depending on the number of names provided, display a message based on the above pattern.



Zouhair Rimale, Ph.D.

Expert technique .NET | SharePoint | ASP.NET MVC | WEB API | Angular

Exercise - 2

Write a program and ask the user to enter their name.

Use an array to reverse the name and then store the result in a new string.

Display the reversed name on the console.



Zouhair Rimale, Ph.D.

Expert technique .NET | SharePoint | ASP.NET MVC | WEB API | Angular

Exercise - 3

Write a program and ask the user to enter 5 numbers.

If a number has been previously entered, display an error message and ask the user to re-try.

Once the user successfully enters 5 unique numbers, sort them and display the result on the console.



Zouhair Rimale, Ph.D.

Expert technique .NET | SharePoint | ASP.NET MVC | WEB API | Angular

Exercise - 4

Write a program and ask the user to continuously enter a number or type "Quit" to exit.

The list of numbers may include duplicates.

Display the unique numbers that the user has entered



Zouhair Rimale, Ph.D.

Expert technique .NET | SharePoint | ASP.NET MVC | WEB API | Angular

Exercise - 5

Write a program and ask the user to supply a list of comma separated numbers (e.g 5, 1, 9, 2, 10).

If the list is empty or includes less than 5 numbers, display "Invalid List" and ask the user to re-try; otherwise, display the 3 smallest numbers in the list.



Zouhair Rimale, Ph.D.

Expert technique .NET | SharePoint | ASP.NET MVC | WEB API | Angular