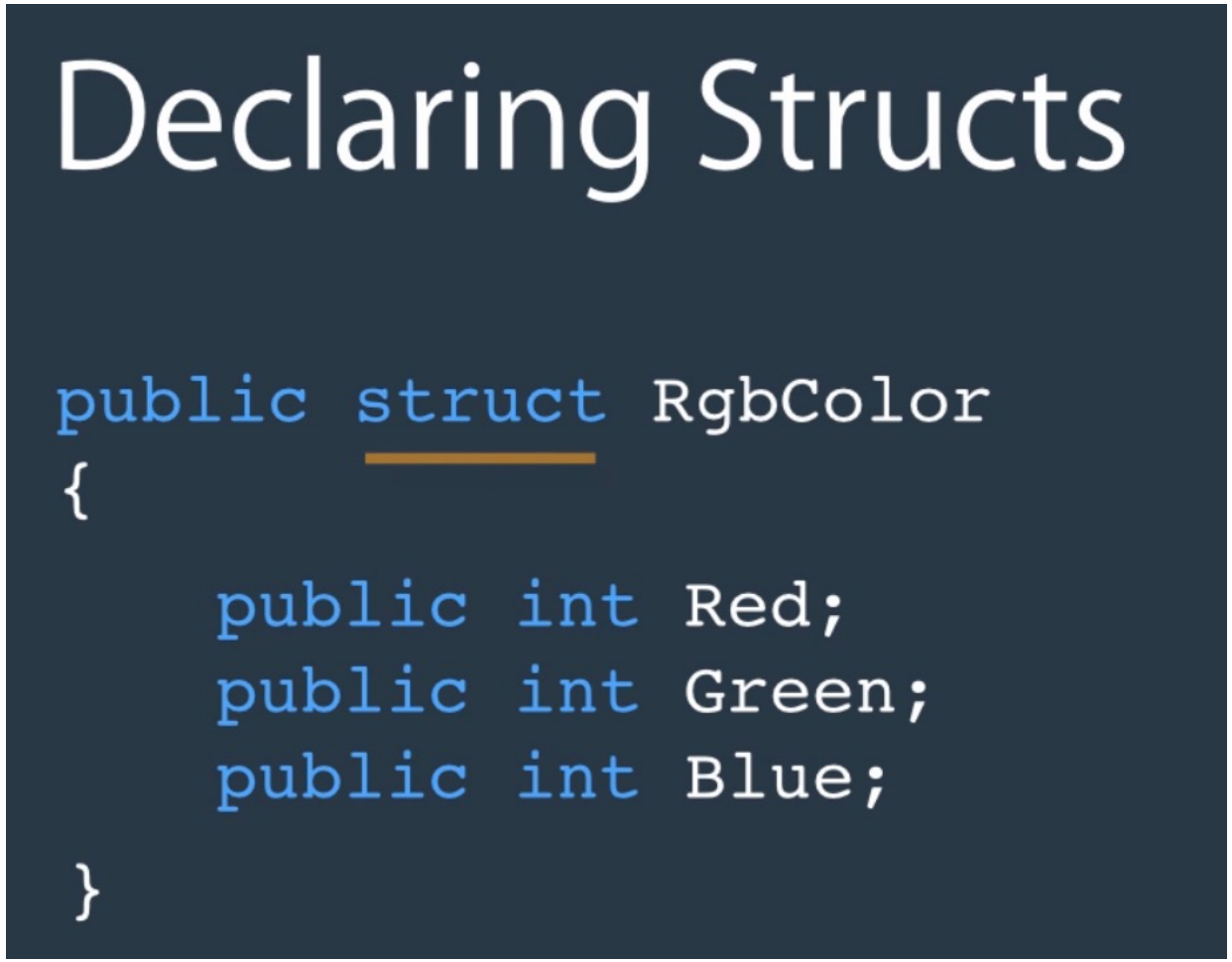
4. Non-Primitive Types

2. Structs
3. Arrays
4. Strings

2. Structure



When creating small light weight objects or points having x and y or when you want to create thousands or ten of thousands of objects of that type; its more efficient to define as structures.

3. Arrays

- What is an Array
- Declaring Arrays
- Initializing Arrays
- Access Array Elements

Array	A data structure to store a collection of variables of the same type.

Declaring Arrays

```
int number1;
int number2;
int number3;


int[] numbers = new int[3];
```

Need to specify the size of the array at the time of the decleration and is fixed cannot be changed later in the program.

new operator – Array is a class in C#
We create an instance of the Array class using new operator.

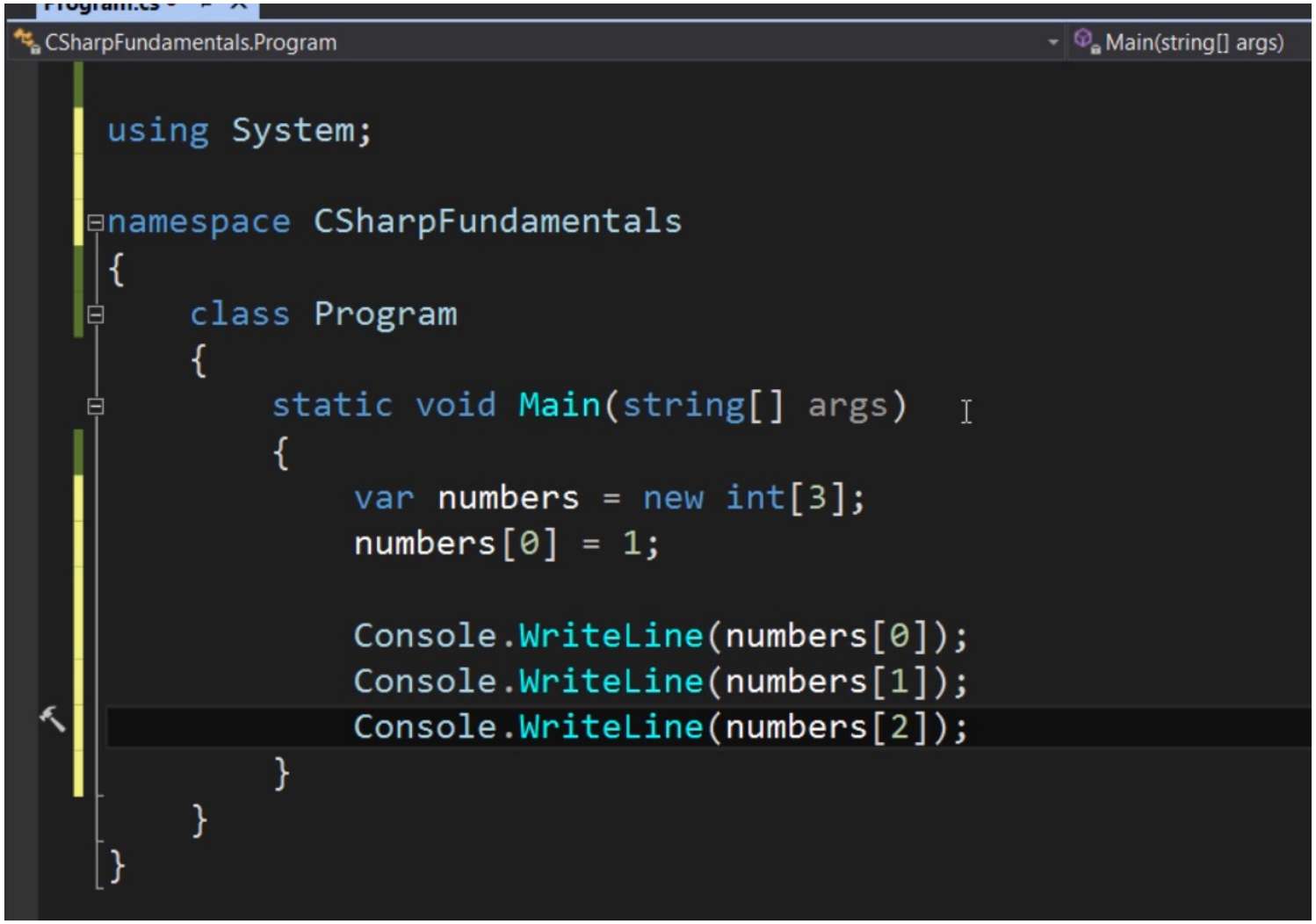
Accessing Array Elements

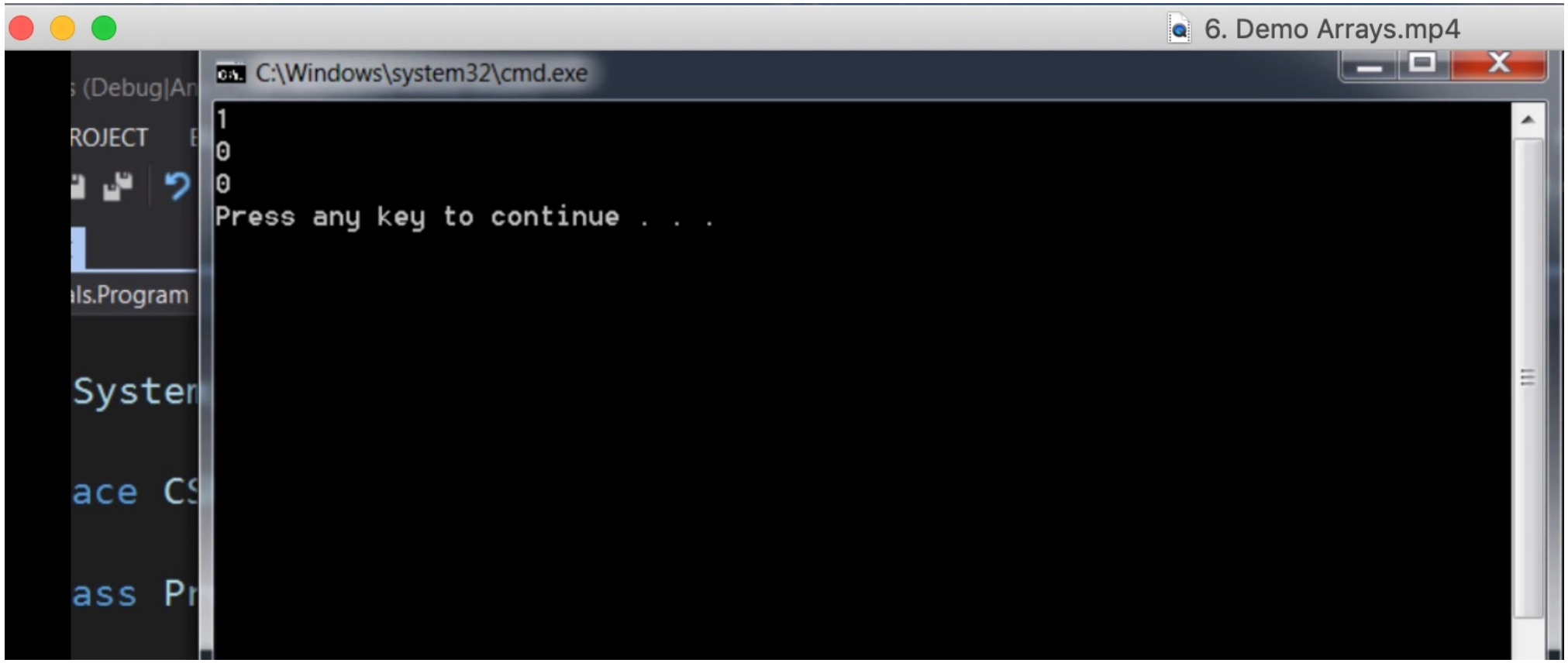
```
int[] numbers = new int[3];

numbers[0] = 1;
numbers[1] = 2;
numbers[2] = 3;
```

Accessing Array Elements

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

~~numbers[0] = 1;~~
~~numbers[1] = 2;~~
~~numbers[2] = 3;~~


Program.cs
CSharpFundamentals.Program
Main(string[] args)
using System;
namespace CSharpFundamentals
{
class Program
{
static void Main(string[] args)
{
var numbers = new int[3];
numbers[0] = 1;
Console.WriteLine(numbers[0]);
Console.WriteLine(numbers[1]);
Console.WriteLine(numbers[2]);
}
}
}



Initialize with a default value. What will happen if the array is of type boolean?

Program.cs

CSharpFundamentals.Program — Main(string[] args)

```
using System;

namespace CSharpFundamentals
{
    class Program
    {
        static void Main(string[] args)
        {
            var numbers = new int[3];
            numbers[0] = 1;

            Console.WriteLine(numbers[0]);
            Console.WriteLine(numbers[1]);
            Console.WriteLine(numbers[2]);

            var flags = new bool[3];
            flags[0] = true;

            Console.WriteLine(flags[0]);
            Console.WriteLine(flags[1]);
            Console.WriteLine(flags[2]);

            var names = new string[3] {"Jack", "John", "Mary"};
        }
    }
}
```

100 %

Object initalization syntex

4. Strings

- What is a string
- How to create strings
- Escape characters and verbatim strings

String

A sequence of characters.
e.g. “Hello World”

Creating Strings

Using String Literals

```
string firstName = “Mosh”;
```

Using String Concatenation

```
string name = firstName + “ “ + lastName;
```

Using String Format

```
string name = string.Format("{0} {1}", firstName, lastName);
```


```
string name = string.Format("{0} {1}", firstName, lastName);
```


Format String – A kind of a template
Inside template – we have Place holders which are 0 index


```
string name = string.Format("{0} {1}", firstName, lastName);
```



```
string name = string.Format("{0} {1}", firstName, lastName);
```

Using String Join

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

string list = string.Join(",", numbers);
```

```
string name = “Mosh”;

char firstChar = name[0];
```

Strings are Immutable

- Once you create them, you cannot change them.

```
string name = "Mosh";

char firstChar = name[0];

name[0] = 'm';
```

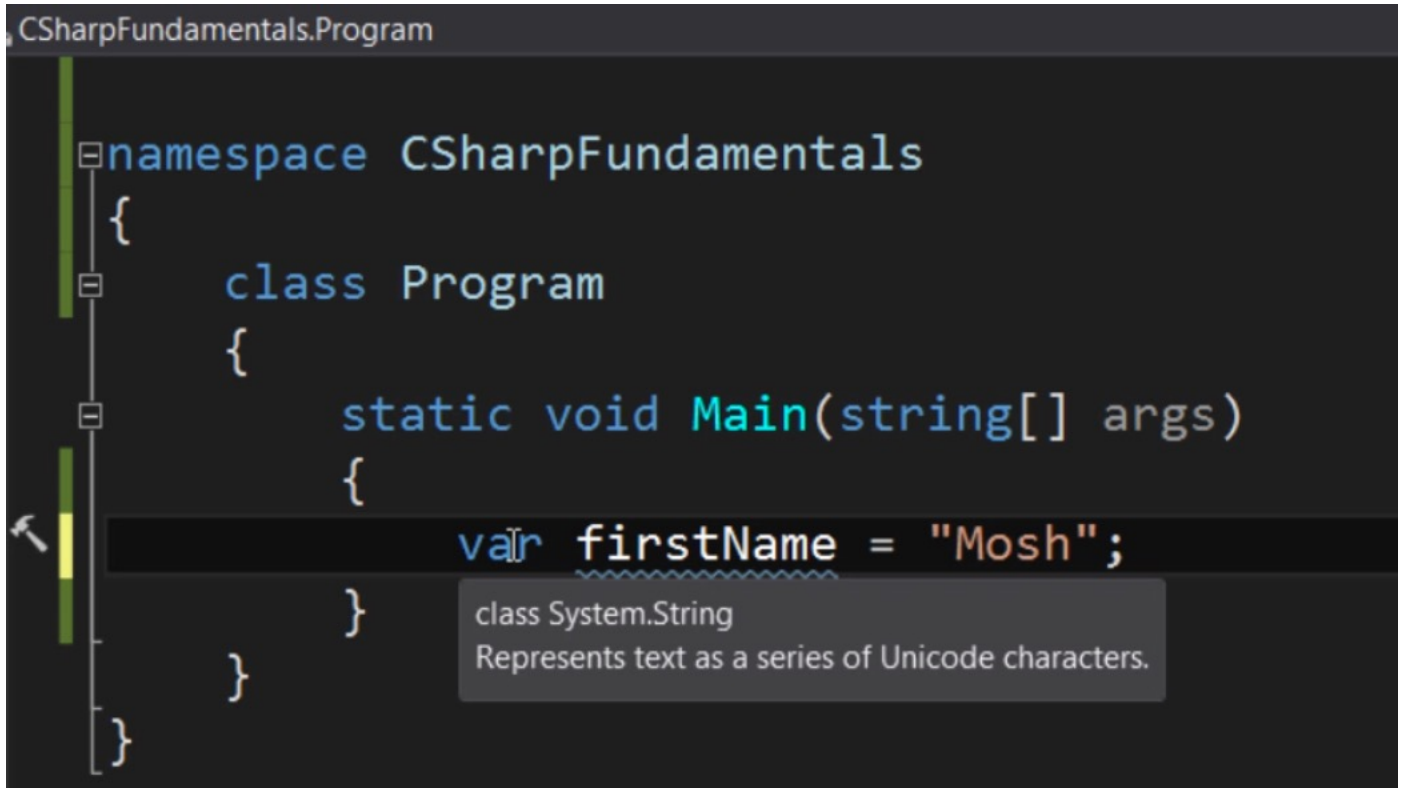
We have methods in string to modifed a string – however, they always return a new string

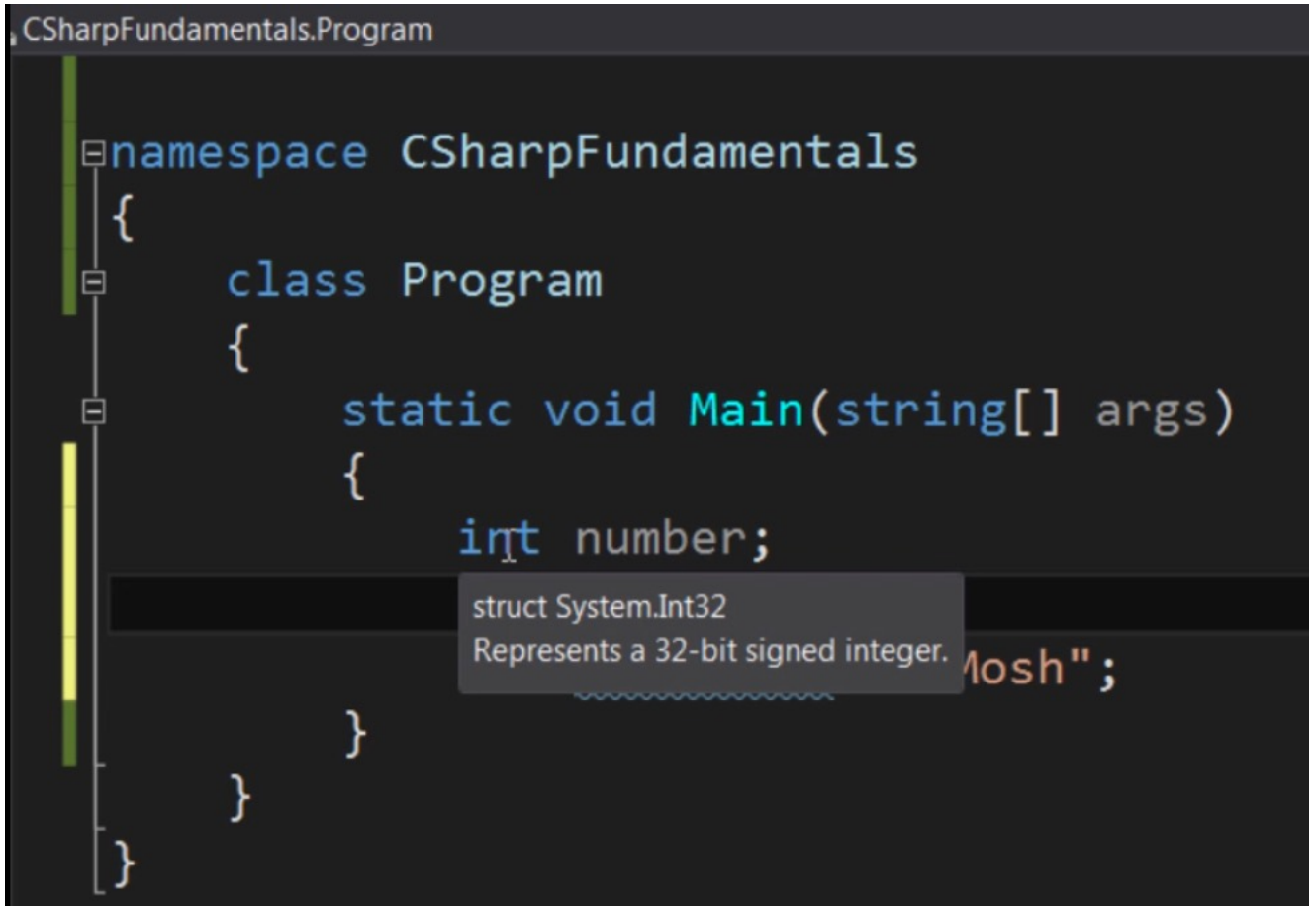
Escape Characters

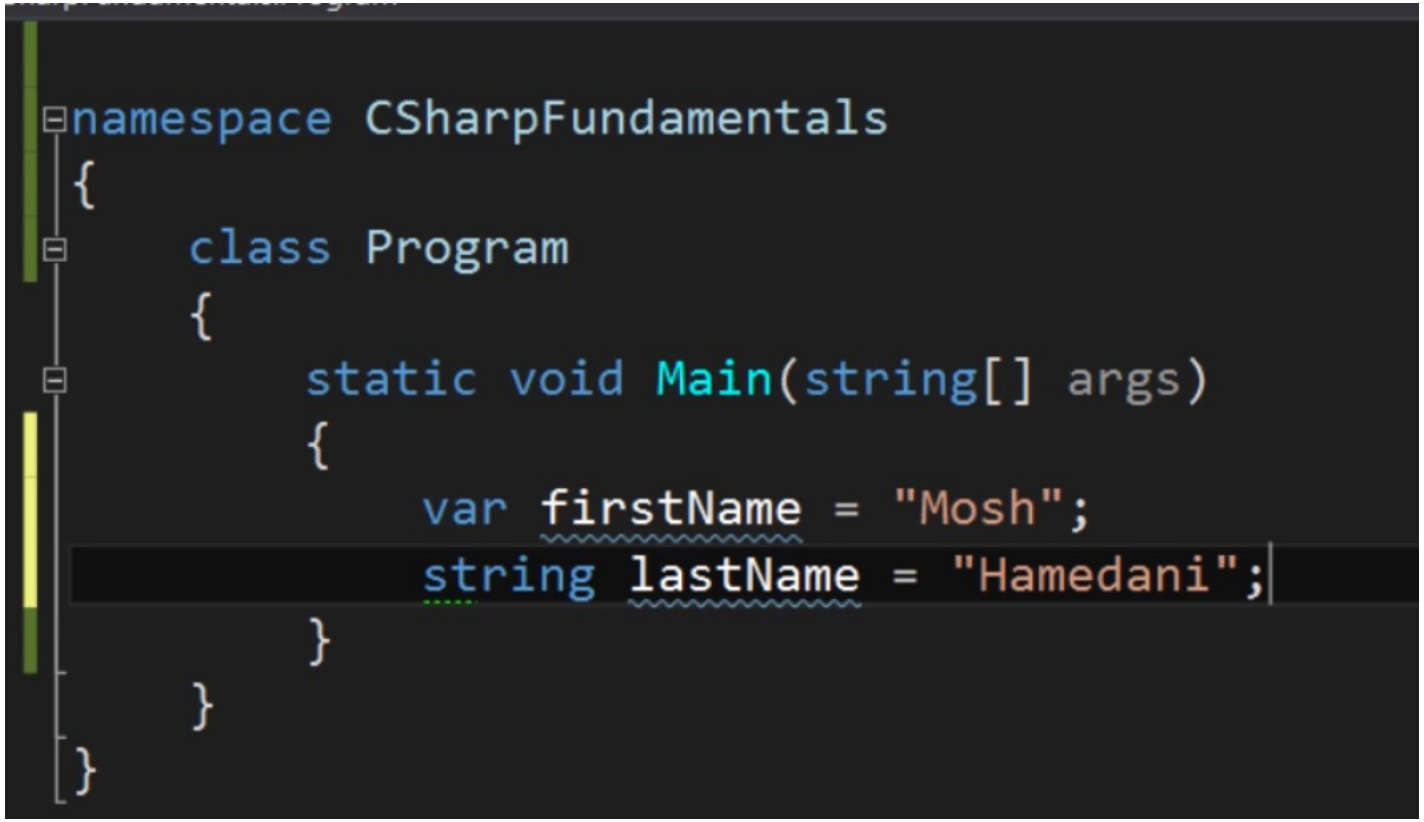
Char	Description
\n	New Line
\t	Tab
\\	Backslash
\’	Single Quotation Mark
\”	Double Quotation Mark

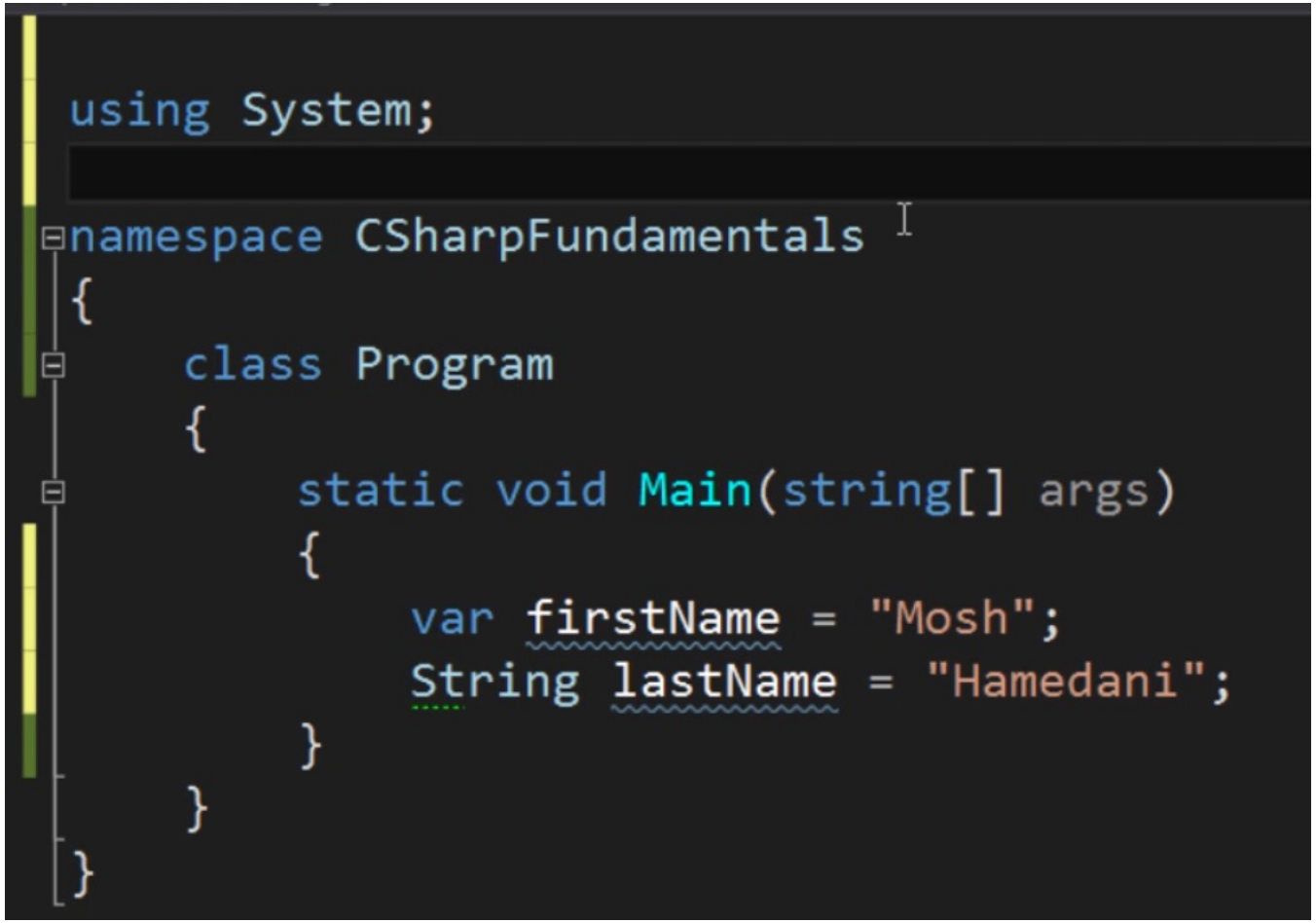
Verbatim Strings

```
string path = "c:\\projects\\project1\\folder1";

string path = @"c:\projects\project1\folder1";
```

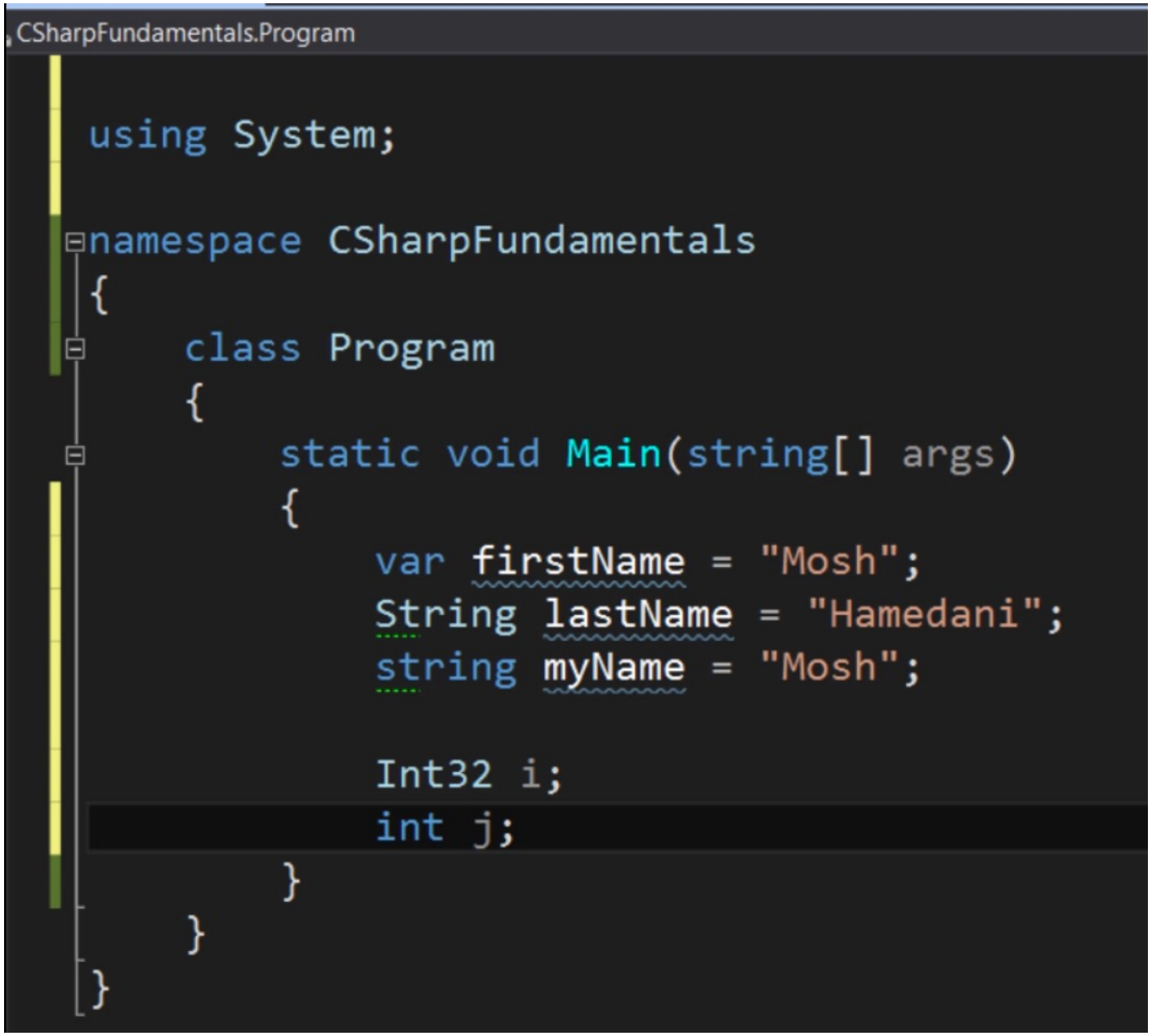

CSharpFundamentals.Program
namespace CSharpFundamentals
{
class Program
{
static void Main(string[] args)
{
var firstName = "Mosh";
}
}
}
class System.String
Represents text as a series of Unicode characters.

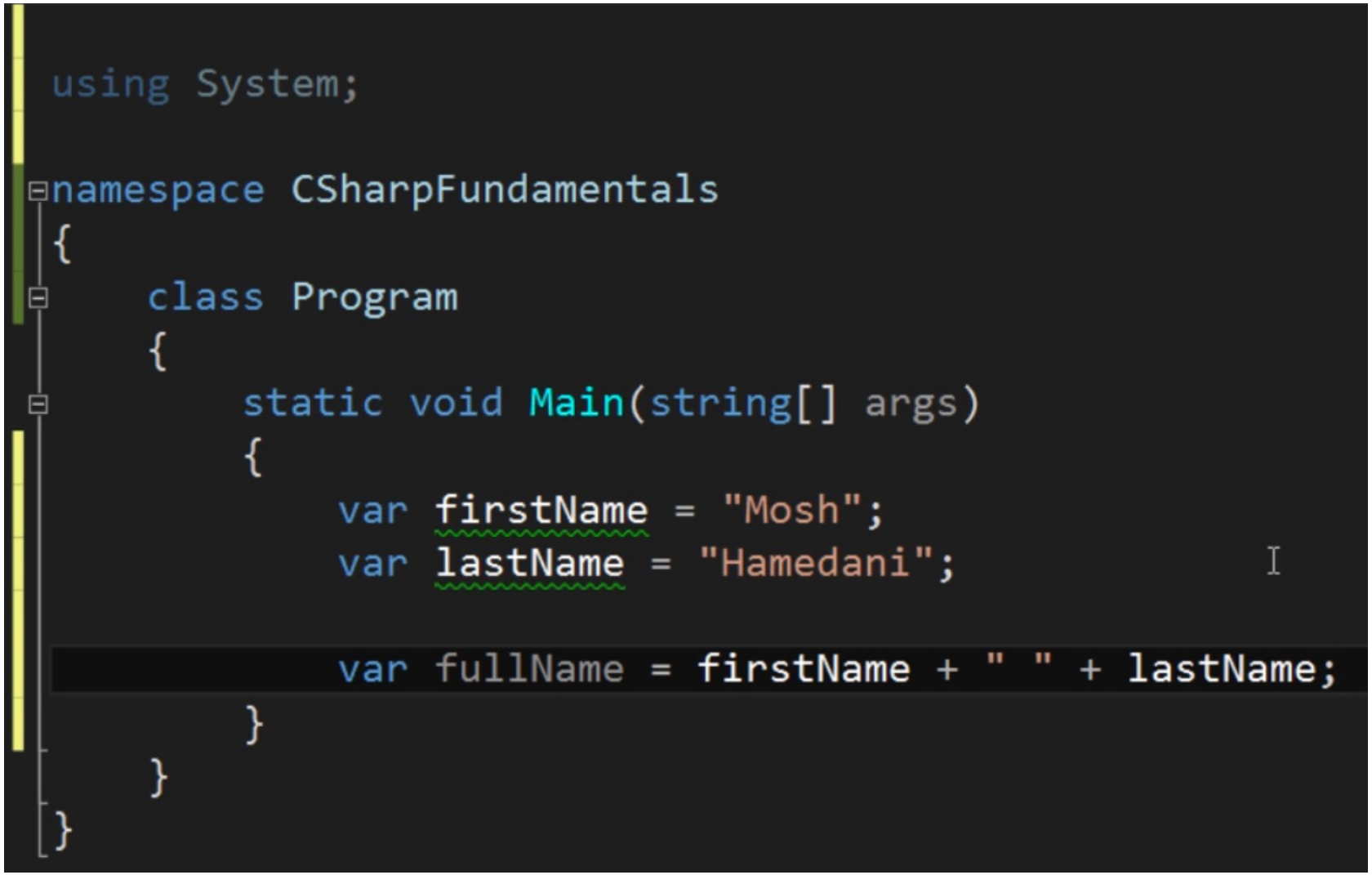

CSharpFundamentals.Program
namespace CSharpFundamentals
{
class Program
{
static void Main(string[] args)
{
int number;
struct System.Int32
Represents a 32-bit signed integer.
Mosh";
}
}
}


namespace CSharpFundamentals
{
class Program
{
static void Main(string[] args)
{
var firstName = "Mosh";
string lastName = "Hamedani";
}
}
}

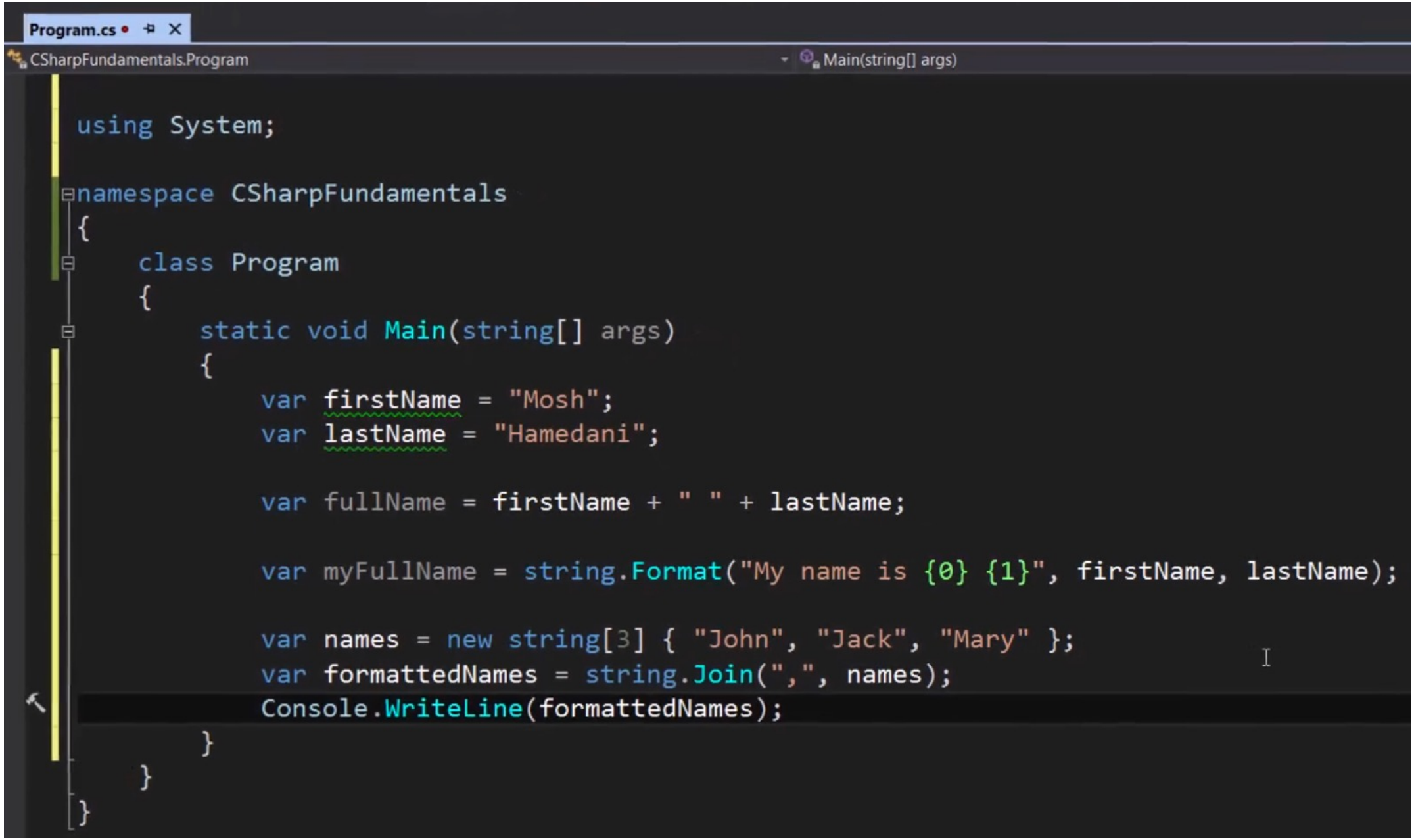


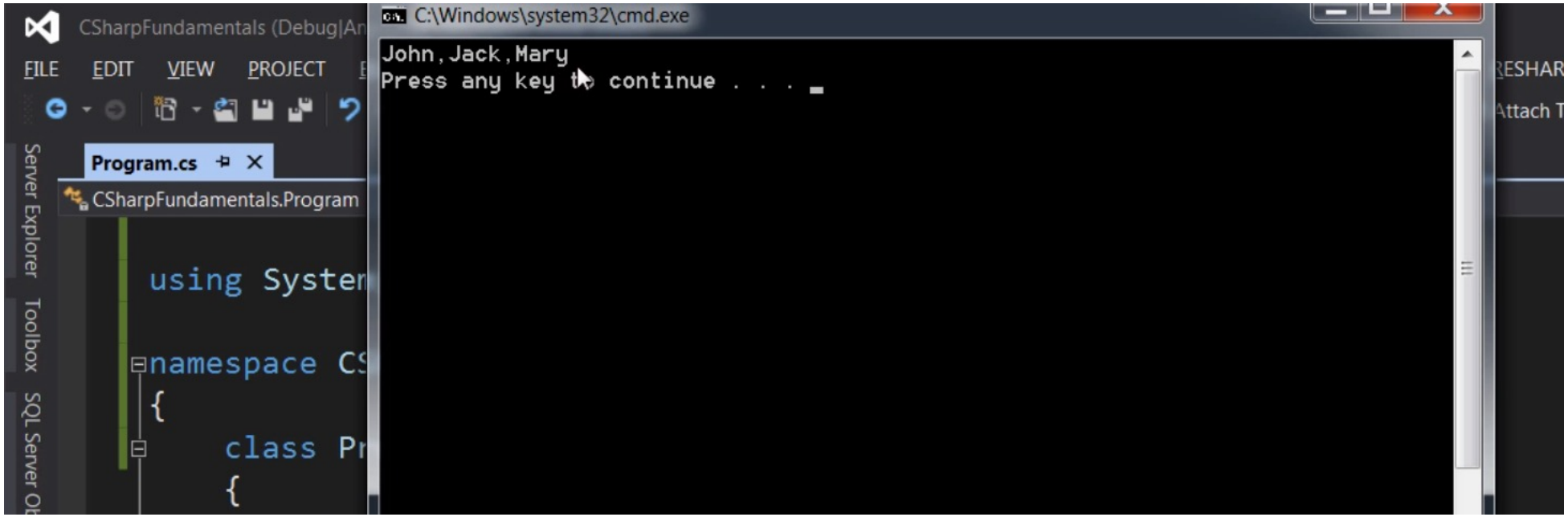
```
using System;

namespace CSharpFundamentals
{
    class Program
    {
        static void Main(string[] args)
        {
            var firstName = "Mosh";
            String lastName = "Hamedani";
        }
    }
}
```


CSharpFundamentals.Program
using System;
namespace CSharpFundamentals
{
class Program
{
static void Main(string[] args)
{
var firstName = "Mosh";
String lastName = "Hamedani";
string myName = "Mosh";
Int32 i;
int j;
}
}
}


using System;
namespace CSharpFundamentals
{
class Program
{
static void Main(string[] args)
{
var firstName = "Mosh";
var lastName = "Hamedani";
var fullName = firstName + " " + lastName;
}
}
}

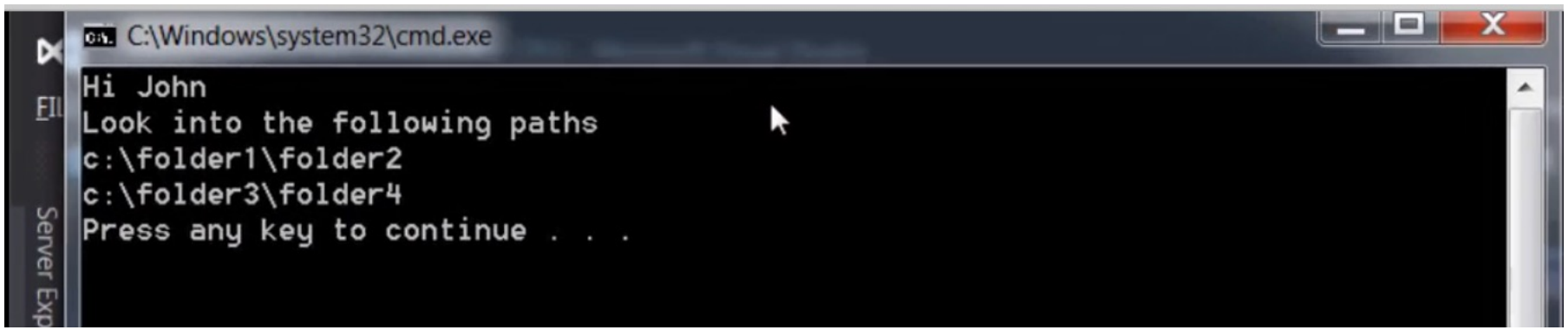
```
using System;

namespace CSharpFundamentals
{
    class Program
    {
        static void Main(string[] args)
        {
            var firstName = "Mosh";
            var lastName = "Hamedani";

            var fullName = firstName + " " + lastName;

            var myFullName = string.Format("My name is {0} {1}", firstName, lastName);
        }
    }
}
```


Program.cs
CSharpFundamentals.Program
Main(string[] args)
using System;
namespace CSharpFundamentals
{
class Program
{
static void Main(string[] args)
{
var firstName = "Mosh";
var lastName = "Hamedani";
var fullName = firstName + " " + lastName;
var myFullName = string.Format("My name is {0} {1}", firstName, lastName);
var names = new string[3] { "John", "Jack", "Mary" };
var formattedNames = string.Join(",", names);
Console.WriteLine(formattedNames);
}
}
}


CSharpFundamentals (Debug|An
FILE
EDIT
VIEW
PROJECT
Program.cs
CSharpFundamentals.Program
using Syste
namespace CS
{
class Pr
{
Server Explorer
Toolbox
SQL Server Ob
C:\Windows\system32\cmd.exe
John,Jack,Mary
Press any key to continue . . .
RESHAR
Attach T

CSharpFundamentals.Program | Main(string[] args)

```
using System;

namespace CSharpFundamentals
{
    class Program
    {
        static void Main(string[] args)
        {
            var firstName = "Mosh";
            var lastName = "Hamedani";

            var fullName = firstName + " " + lastName;

            var myFullName = string.Format("My name is {0} {1}", firstName, lastName);

            var names = new string[3] { "John", "Jack", "Mary" };
            var formattedNames = string.Join(",", names);

            var text = "Hi John\nLook into the following paths\nc:\\folder1\\folder2\nc:\\folde
            Console.WriteLine(text);
        }
    }
}
```


C:\Windows\system32\cmd.exe
Hi John
Look into the following paths
c:\folder1\folder2
c:\folder3\folder4
Press any key to continue . . .

```
namespace CSharpFundamentals
{
    class Program
    {
        static void Main(string[] args)
        {
            var firstName = "Mosh";
            var lastName = "Hamedani";

            var fullName = firstName + " " + lastName;

            var myFullName = string.Format("My name is {0} {1}", firstName, lastName);

            var names = new string[3] { "John", "Jack", "Mary" };
            var formattedNames = string.Join(",", names);

            var text = @"Hi John
Look into the following paths
c:\folder1\folder2
c:\folder3\folder4";
            Console.WriteLine(text);
        }
    }
}
```