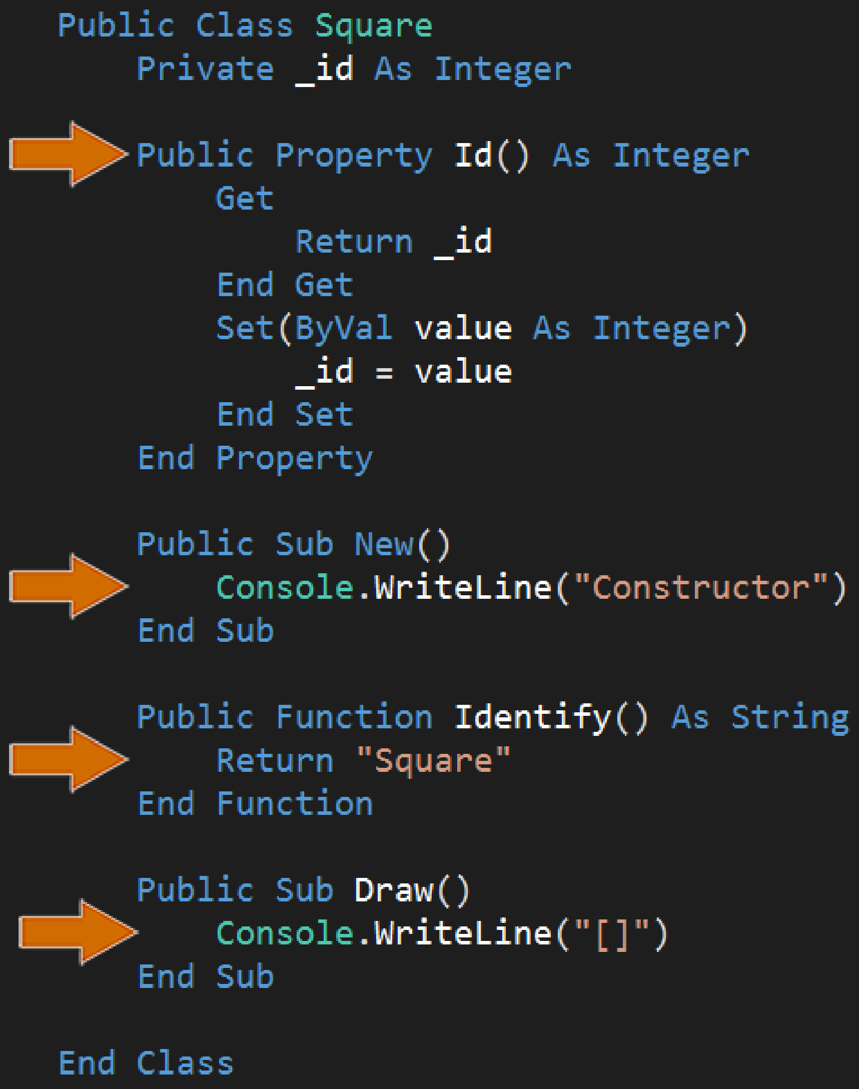
C# for Visual Basic .NET Developers

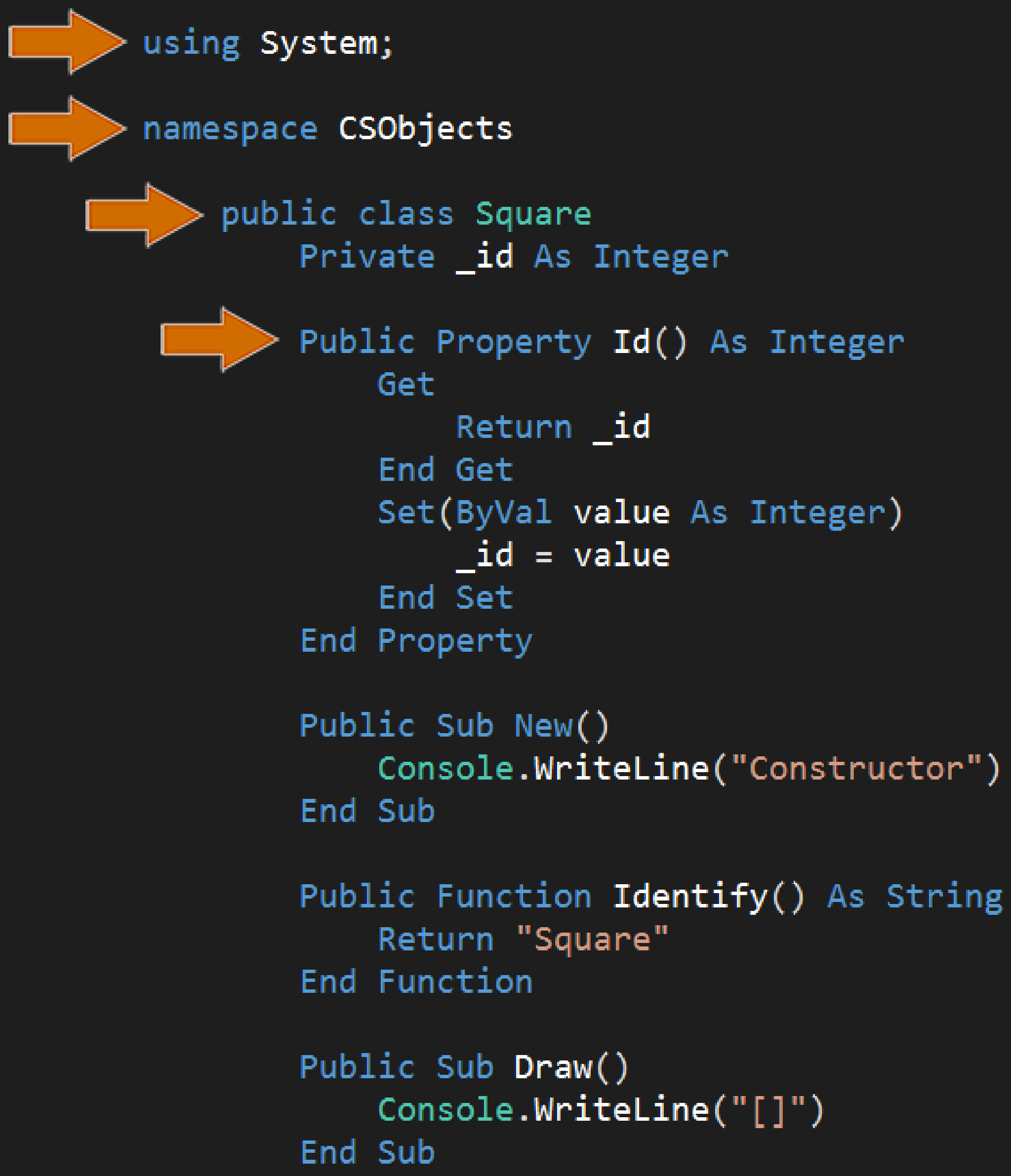
Classes, Interfaces and Inheritance

Craig Shoemaker
craigshoemaker.net
@craigshoemaker

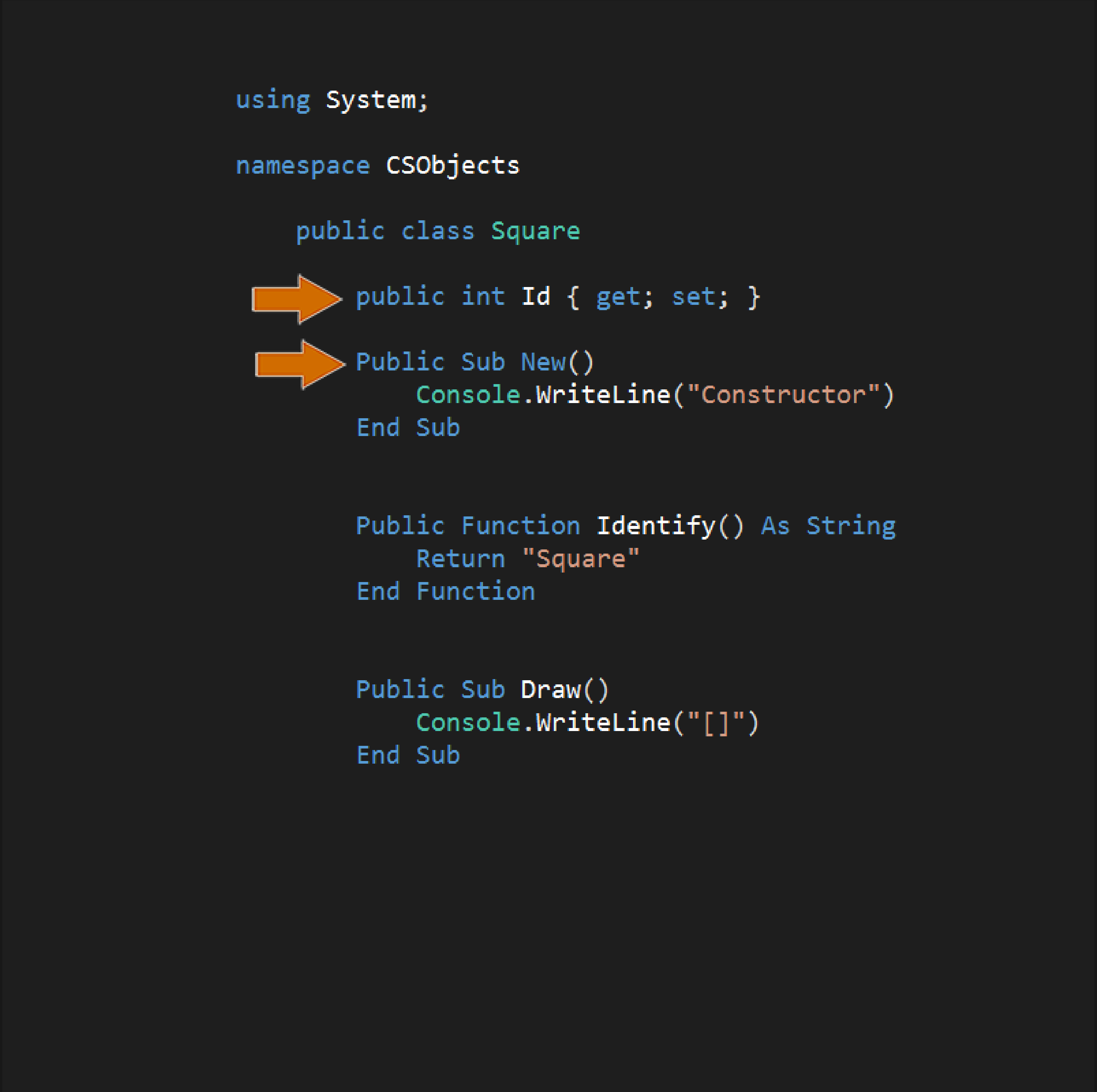
pluralsight
hardcore developer training



```
Public Class Square
    Private _id As Integer

    Public Property Id() As Integer
        Get
            Return _id
        End Get
        Set(ByVal value As Integer)
            _id = value
        End Set
    End Property

    Public Sub New()
        Console.WriteLine("Constructor")
    End Sub

    Public Function Identify() As String
        Return "Square"
    End Function

    Public Sub Draw()
        Console.WriteLine("[]")
    End Sub

End Class
```



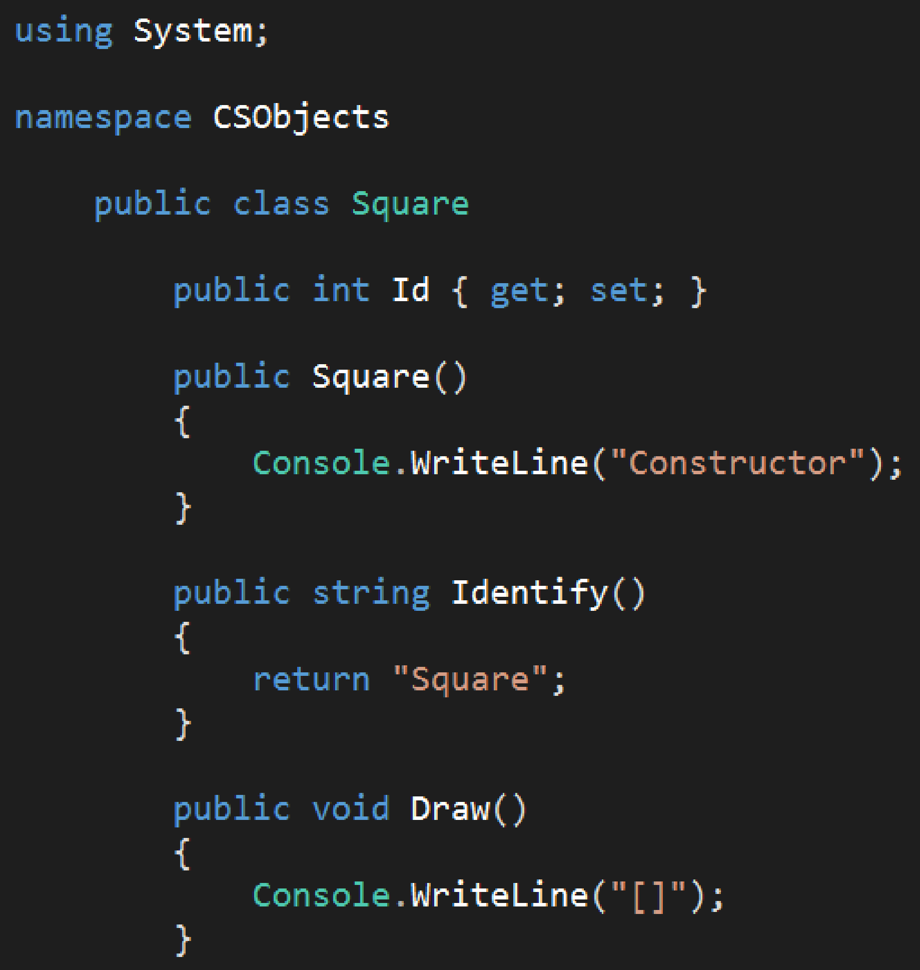
```
using System;
namespace CSObjects
    public class Square
        Private _id As Integer

        Public Property Id() As Integer
            Get
                Return _id
            End Get
            Set(ByVal value As Integer)
                _id = value
            End Set
        End Property

        Public Sub New()
            Console.WriteLine("Constructor")
        End Sub

        Public Function Identify() As String
            Return "Square"
        End Function

        Public Sub Draw()
            Console.WriteLine("[]")
        End Sub
```



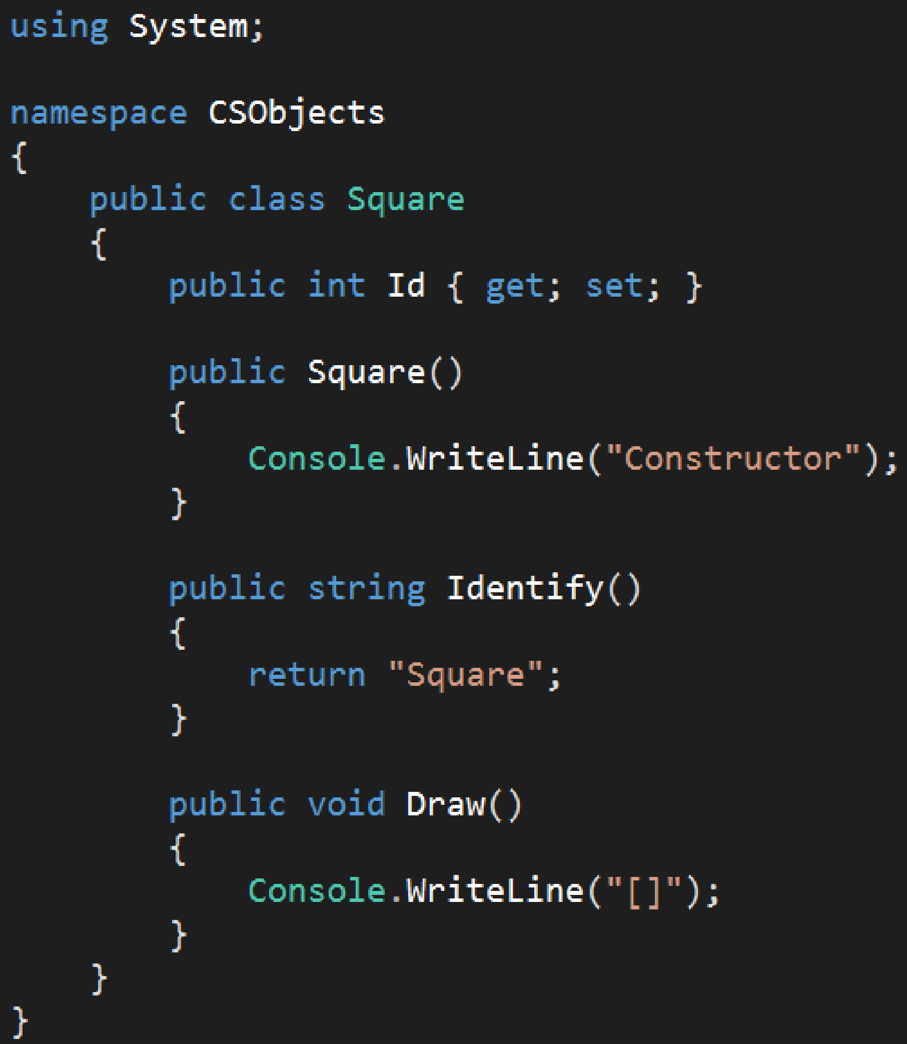
```
using System;

namespace CSObjects

    public class Square

        public int Id { get; set; }

        Public Sub New()
            Console.WriteLine("Constructor")
        End Sub


        Public Function Identify() As String
            Return "Square"
        End Function


        Public Sub Draw()
            Console.WriteLine("[]")
        End Sub
```

```
using System;

namespace CSObjects

    public class Square

        public int Id { get; set; }

        public Square()
        {
            Console.WriteLine("Constructor");
        }

        Public Function Identify() As String
            Return "Square"
        End Function


        Public Sub Draw()
            Console.WriteLine("[]")
        End Sub
```

```
using System;

namespace CSObjects

    public class Square

        public int Id { get; set; }

        public Square()
        {
            Console.WriteLine("Constructor");
        }

        public string Identify()
        {
            return "Square";
        }

        Public Sub Draw()
            Console.WriteLine("[]")
        End Sub
```



```
using System;

namespace CSObjects

    public class Square

        public int Id { get; set; }

        public Square()
        {
            Console.WriteLine("Constructor");
        }

        public string Identify()
        {
            return "Square";
        }

        public void Draw()
        {
            Console.WriteLine("[]");
        }
```



```
using System;

namespace CSObjects
{
    public class Square
    {
        public int Id { get; set; }

        public Square()
        {
            Console.WriteLine("Constructor");
        }

        public string Identify()
        {
            return "Square";
        }

        public void Draw()
        {
            Console.WriteLine("[]");
        }
    }
}
```

Access Modifier Equivalents

VB.NET	C#
Public	public
Private	private
Protected	protected
Friend	internal
Protected Friend	protected internal

Source: http://bit.ly/16H9TZ1

VB.NET Casting Operators

VB.NET Casting Operators

DirectCast

VB.NET Casting Operators

DirectCast

CType

VB.NET Casting Operators

DirectCast
CType
TryCast

VB.NET Casting Operators

DirectCast

CType

TryCast

CBool, CByte, CChar, etc.

C# Casting Operators

as

C# Casting Operators

as

()

Casting Equivalents

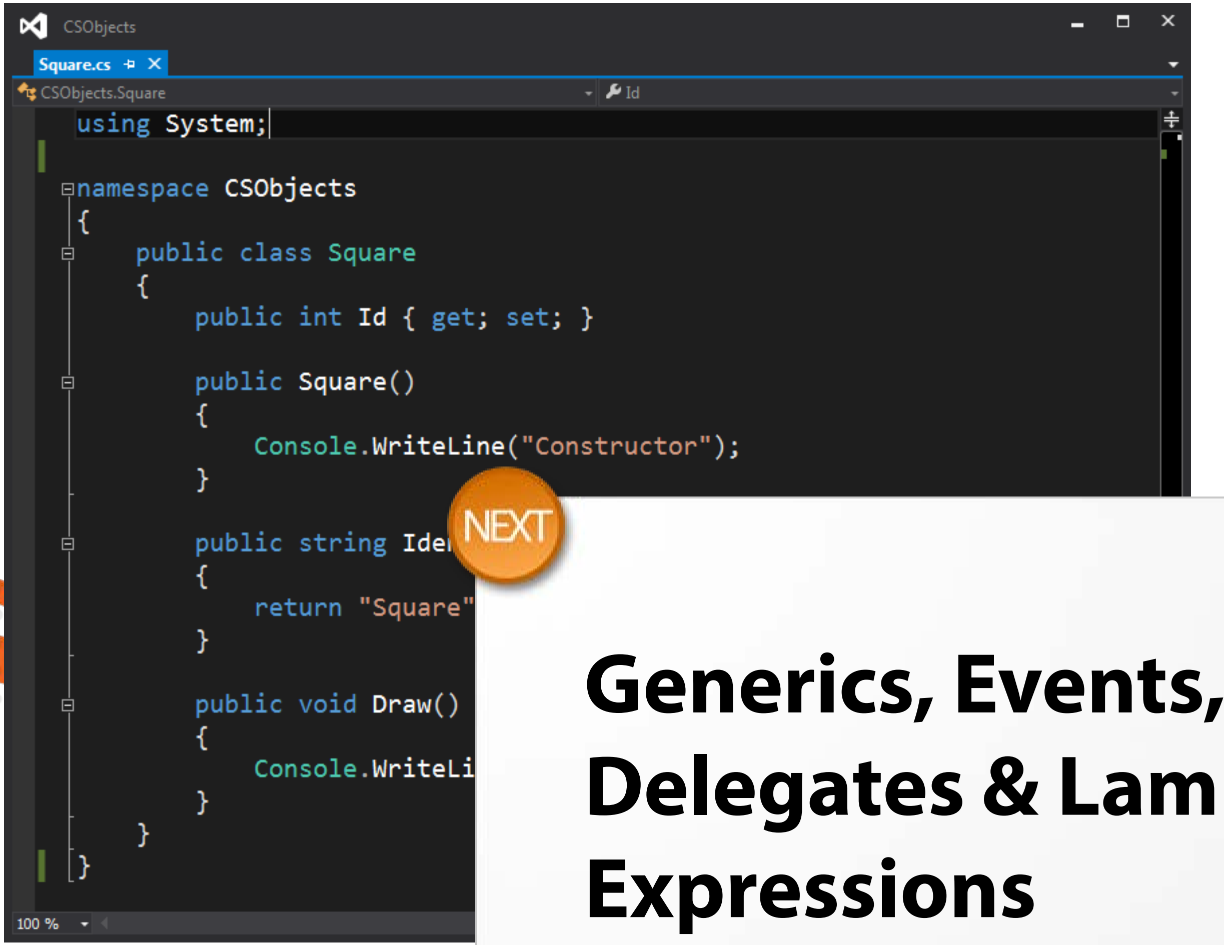
VB.NET	C#	

CType
DirectCast
CBool, CByte, etc.
	((type)object)
	Convert.To___()
	type.Parse()
	type.TryParse()
		*

TryCast	(object as type)	

* Equivalent, but not exact counterparts. Source: http://bit.ly/15qIWYk

Summary



```
using System;

namespace CSObjects
{
    public class Square
    {
        public int Id { get; set; }

        public Square()
        {
            Console.WriteLine("Constructor");
        }

        public string Ide
        {
            return "Square"
        }

        public void Draw()
        {
            Console.WriteLi
        }
    }
}
```

NEXT

Generics, Events, Delegates & Lambda Expressions