

# Generics



**Zouhair Rimale, Ph.D.**

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

Generics (Debug|Any CPU) - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM SQL SQL PROMPT TOOLS VISUALSVN TEST ARCHITECTURE SQL CONNECT DOTCOVER RESHARPER ANALYZE WINDOW HELP

Start Debug Synchronize Attach To IIS

List.cs BookList.cs Program.cs Sample.cs

Generics.List this[int index]

```
using System;

namespace Generics
{
    public class List
    {
        public void Add(int number)
        {
            throw new NotImplementedException();
        }

        public int this[int index]
        {
            get { throw new NotImplementedException(); }
        }
    }
}
```



```
namespace Generics
{
    public class BookList
    {
        public void Add(Book book)
        {
            throw new NotImplementedException();
        }

        public Book this[int index]
        {
            get { throw new NotImplementedException(); }
        }
    }
}
```

struct System.Int32  
Represents a 32-bit signed integer.



```
public class ObjectList
{
    public void Add(object value)
    {
    }

    public object this[int index]
    {
        get { throw new NotImplementedException(); }
    }
}
```

```
public class GenericList<T>
{
    public void Add(T value)
    {
    }

    public T this[int index]
    {
        get { throw new NotImplementedException(); }
    }
}
```



Generics (Debug)Any CPU - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM SQL SQL PROMPT TOOLS VISUALSVN TEST ARCHITECTURE SQL CONNECT DOTCOVER RESHARPER ANALYZE WINDOW HELP

Start Debug Synchronize Attach To IIS

List.cs BookList.cs Program.cs Sample.cs

Generics.Program Main(string[] args)

```
namespace Generics
{
    class Program
    {
        static void Main(string[] args)
        {
            var book = new Book { Isbn = "1111", Title = "C# Advanced" };

            //var numbers = new List();
            //numbers.Add(10);

            //var books = new BookList();
            //books.Add(book);

            var numbers = new GenericList<int>();
            numbers.Add(10);

            var books = new GenericList<Book>();
            books.Add(new Book());
        }
    }
}
```



```
public class GenericDictionary<TKey, TValue>
{
    public void Add(TKey key, TValue value)
    {
    }
}
```

```
static void Main(string[] args)
{
    var book = new Book { Isbn = "1111", Title = "C# Advanced" };

    //var numbers = new List();
    //numbers.Add(10);

    //var books = new BookList();
    //books.Add(book);

    var numbers = new GenericList<int>();
    numbers.Add(10);

    var books = new GenericList<Book>();
    books.Add(new Book());

    var dictionary = new GenericDictionary<string, Book>();
    dictionary.Add("1234", new Book());
}
```



```
public class Utilities
{
    public int Max(int a, int b)
    {
        return a > b ? a : b;
    }

    public T Max<T>(T a, T b)
    {
        return a > b ? a : b;
    }
}
```

```
public T Max<T>(T a, T b) where T : IComparable
{
    return a.CompareTo(b) > 0 ? a : b;
}
```





Generics (Debug)Any CPU - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM SQL SQL PROMPT TOOLS VISUALSVN TEST ARCHITECTURE SQL CONNECT DOTCOVER RESHARPER ANALYZE WINDOW HELP

Start Debug Synchronize Attach To IIS

List.cs BookList.cs Program.cs Sample.cs Utilities.cs

Generics.Utilities<T> Max(T a, T b)

```
using System;

namespace Generics
{
    public class Utilities<T> where T : IComparable
    {
        public int Max(int a, int b)
        {
            return a > b ? a : b;
        }

        public T Max(T a, T b)
        {
            return a.CompareTo(b) > 0 ? a : b;
        }
    }
}
```





Generics (Debug|Any CPU) - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM SQL SQL PROMPT TOOLS VISUALSVN TEST ARCHITECTURE SQL CONNECT DOTCOVER RESHARPER ANALYZE WINDOW HELP

Start Debug Synchronize Attach To IIS

List.cs BookList.cs Program.cs Sample.cs Utilities.cs

Generics.Utilities<T> Max(int a, int b)

```
using System;

namespace Generics
{
    // where T : IComparable
    // where T : Product
    // where T : struct
    // where T : class
    // where T : new()
    public class Utilities<T> where T : IComparable
    {
        public int Max(int a, int b)
        {
            return a > b ? a : b;
        }

        public T Max(T a, T b)
        {
            return a.CompareTo(b) > 0 ? a : b;
        }
    }
}
```



```
namespace Generics
{
    public class Product
    {
        public string Title { get; set; }
        public float Price { get; set; }
    }
}
```

```
namespace Generics
{
    public class DiscountCalculator<TProduct> where TProduct : Product
    {
        public float CalculateDiscount(TProduct product)
        {
            return product.Price;
        }
    }
}
```



```
public class Nullable<T> where T : struct
{
    private object _value;

    public Nullable()
    {
    }

    public Nullable(T value)
    {
        _value = value;
    }

    public bool HasValue
    {
        get { return _value != null; }
    }

    public T GetValueOrDefault()
    {
        if (HasValue)
            return (T)_value;

        return default(T);
    }
}
```



```
namespace Generics
{
    class Program
    {
        static void Main(string[] args)
        {
            var number = new Nullable<int>(5);
            Console.WriteLine("Has Value ?" + number.HasValue);
            Console.WriteLine("Value: " + number.GetValueOrDefault());
        }
    }
}
```

C:\Windows\system32\cmd.exe

```
Has Value ?True
Value: 5
Press any key to continue . . .
```

```
class Program
{
    static void Main(string[] args)
    {
        var number = new Nullable<int>();
        Console.WriteLine("Has Value ?" + number.HasValue);
        Console.WriteLine("Value: " + number.GetValueOrDefault());
    }
}
```

C:\Windows\system32\cmd.exe

```
Has Value ?False
Value: 0
Press any key to continue . . .
```



```
namespace Generics
{
    // where T : Comparable
    // where T : Product
    // where T : struct
    // where T : class
    // where T : new()
    public class Utilities<T> where T : Comparable
    {
        public int Max(int a, int b)
        {
            return a > b ? a : b;
        }

        public void DoSomething(T value)
        {
            var obj = new T();
        }

        public T Max(T a, T b)
        {
            return a.CompareTo(b) > 0 ? a : b;
        }
    }
}
```



```

namespace Generics
{
    // where T : IComparable
    // where T : Product
    // where T : struct
    // where T : class
    // where T : new()
    public class Utilities<T> where T : IComparable, new()
    {
        public int Max(int a, int b)
        {
            return a > b ? a : b;
        }

        public void DoSomething(T value)
        {
            var obj = new T();
        }

        public T Max(T a, T b)
        {
            return a.CompareTo(b) > 0 ? a : b;
        }
    }
}

```



# Delegates



**Zouhair Rimale, Ph.D.**

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



# Delegates

- An object that knows how to call a method (or a group of methods)
- A reference to a function



# Why do we need delegates?

- For designing extensible and flexible applications (eg frameworks)



```
namespace Delegates
{
    public class Photo
    {
        public static Photo Load(string path)
        {
            return new Photo();
        }

        public void Save()
        {
        }
    }
}
```

```
public class PhotoFilters
{
    public void ApplyBrightness(Photo photo)
    {
        Console.WriteLine("Apply brightness");
    }

    public void ApplyContrast(Photo photo)
    {
        Console.WriteLine("Apply contrast");
    }

    public void Resize(Photo photo)
    {
        Console.WriteLine("Resize photo");
    }
}
```



```
namespace Delegates
{
    public class PhotoProcessor
    {
        public void Process(string path)
        {
            var photo = Photo.Load(path);

            var filters = new PhotoFilters();
            filters.ApplyBrightness(photo);
            filters.ApplyContrast(photo);
            filters.Resize(photo);

            photo.Save();
        }
    }
}
```



```
public class PhotoProcessor
{
    public delegate void PhotoFilterHandler(Photo photo);

    public void Process(string path, PhotoFilterHandler filterHandler)
    {
        var photo = Photo.Load(path);

        filterHandler(photo);

        photo.Save();
    }
}
```

```
namespace Delegates
{
    class Program
    {
        static void Main(string[] args)
        {
            var processor = new PhotoProcessor();
            var filters = new PhotoFilters();
            PhotoProcessor.PhotoFilterHandler filterHandler = filters.ApplyBrightness;
            filterHandler += filters.ApplyContrast;

            processor.Process("photo.jpg", filterHandler);
        }
    }
}
```



```

class Program
{
    static void Main(string[] args)
    {
        var processor = new PhotoProcessor();
        var filters = new PhotoFilters();
        PhotoProcessor.PhotoFilterHandler filterHandler = filters.ApplyBrightness;
        filterHandler += filters.ApplyContrast;
        filterHandler += RemoveRedEyeFilter;

        processor.Process("photo.jpg", filterHandler);
    }

    static void RemoveRedEyeFilter(Photo photo)
    {
        Console.WriteLine("Apply RemoveRedEye");
    }
}

```



```
public class PhotoProcessor
{
    public void Process(string path, Action<Photo> filterHandler)
    {
        var photo = Photo.Load(path);

        filterHandler(photo);

        photo.Save();
    }
}
```

```
static void Main(string[] args)
{
    var processor = new PhotoProcessor();
    var filters = new PhotoFilters();
    Action<Photo> filterHandler = filters.ApplyBrightness;
    filterHandler += filters.ApplyContrast;
    filterHandler += RemoveRedEyeFilter;

    processor.Process("photo.jpg", filterHandler);
}

static void RemoveRedEyeFilter(Photo photo)
{
    Console.WriteLine("Apply RemoveRedEye");
}
}
```





# Delegates

- An object that knows how to call a method (or a group of methods)
- A reference to a function



# Why do we need delegates?

- For designing extensible and flexible applications (eg frameworks)



# Interfaces or Delegates?

Use a delegate when

- An eventing design pattern is used.
- The caller doesn't need to access other properties or methods on the object implementing the method.



# Lambda Expressions



**Zouhair Rimale, Ph.D.**

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

# What is a Lambda Expression?

An anonymous method

- No access modifier
- No name
- No return statement



# Why do we use them?

For convenience



```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine(Square(5));
    }

    static int Square(int number)
    {
        return number*number;
    }
}
```





```
static void Main(string[] args)
{
    // args => expression
    number => number*number

    Console.WriteLine(Square(5));
}
```

```
static int Square(int number)
{
    return number*number;
}
```

```
static void Main(string[] args)
{
    // args => expression
    //number => number*number;

    Func<int, int> square = Square;

    Console.WriteLine(Square(5));
}
```

```
static int Square(int number)
{
    return number*number;
}
```



```

static void Main(string[] args)
{
    // args => expression
    // number => number*number;

    Func<int, int> square = number => number*number;

    Console.WriteLine(Square(5));
}

static int Square(int number)
{
    return number*number;
}

```

```

static void Main(string[] args)
{
    // args => expression
    // () => ...
    // x => ...
    // (x, y, z) => ...

    Func<int, int> square = number => number*number;

    Console.WriteLine(square(5));
}

```



```
static void Main(string[] args)
{
    // args => expression

    // () => ...
    // x => ...
    // (x, y, z) => ...

    const int factor = 5;

    Func<int, int> multipler = n => n*factor;

    var result = multipler(10);

    Console.WriteLine(result);
}
```



```
public class BookRepository
{
    public List<Book> GetBooks()
    {
        return new List<Book>
        {
            new Book() {Title = "Title 1", Price = 5},
            new Book() {Title = "Title 2", Price = 7},
            new Book() {Title = "Title 3", Price = 17}
        };
    }
}

static void Main(string[] args)
{
    var books = new BookRepository().GetBooks();

    var cheapBooks = books.FindAll(IsCheaperThan10Dollars);

    foreach (var book in cheapBooks)
    {
        Console.WriteLine(book.Title);
    }
}

static bool IsCheaperThan10Dollars(Book book)
{
    return book.Price < 10;
}
```



```
using System;

namespace LambdaExpressions
{
    class Program
    {
        static void Main(string[] args)
        {
            var books = new BookRepository().GetBooks();

            var cheapBooks = books.FindAll(b => b.Price < 10);

            foreach (var book in cheapBooks)
            {
                Console.WriteLine(book.Title);
            }
        }
    }
}
```



# Event and Delegates



**Zouhair Rimale, Ph.D.**

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



# Events

- A mechanism for communication between objects
- Used in building *Loosely Coupled Applications*
- Helps extending applications





```
public class VideoEncoder
{
    public void Encode(Video video)
    {
        // Encoding logic
        // ...

        _mailService.Send(new Mail());
    }
}
```





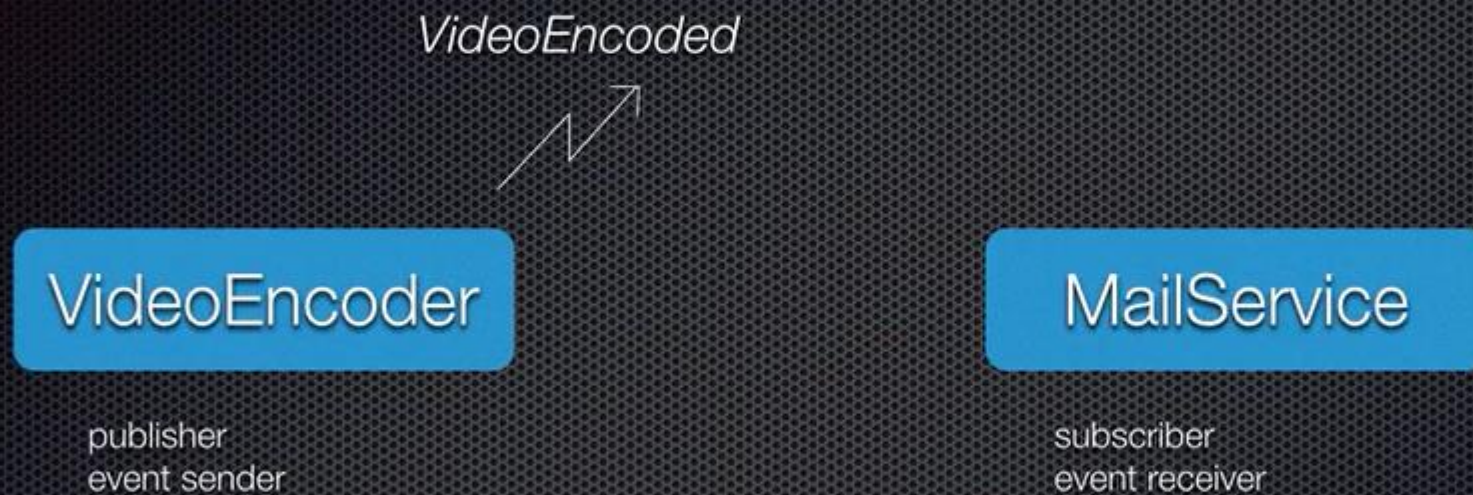
```
public class VideoEncoder
{
    public void Encode(Video video)
    {
        // Encoding logic
        // ...

        _mailService.Send(new Mail());

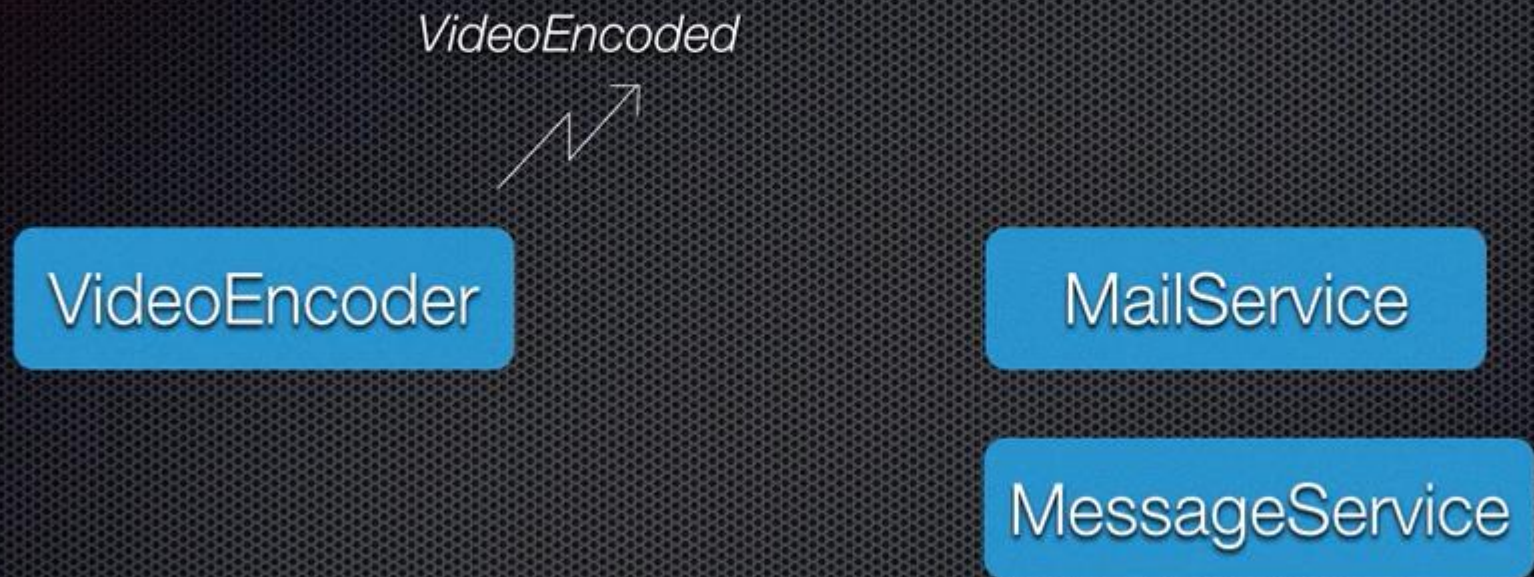
        _messageService.Send(new Text());
    }
}
```











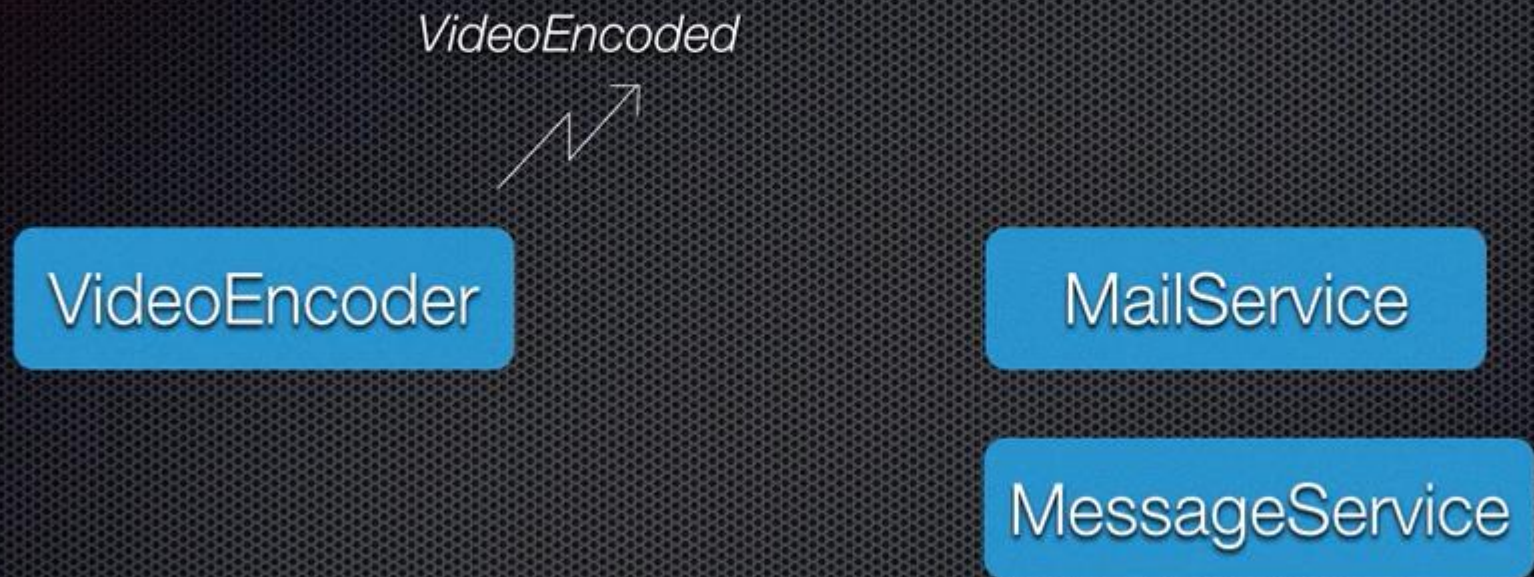


```
public class VideoEncoder
{
    public void Encode(Video video)
    {
        // Encoding logic
        // ...

        OnVideoEncoded();
    }
}
```





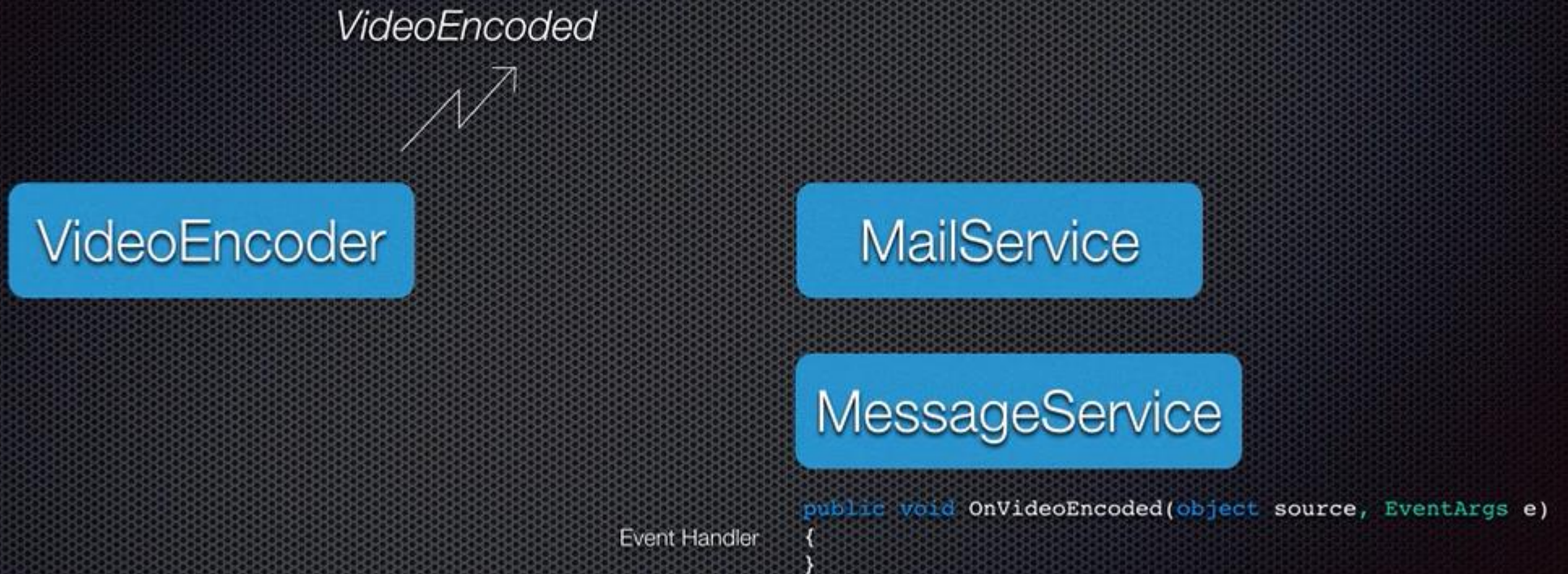




```
public void OnVideoEncoded(object source, EventArgs e)
{
}
```







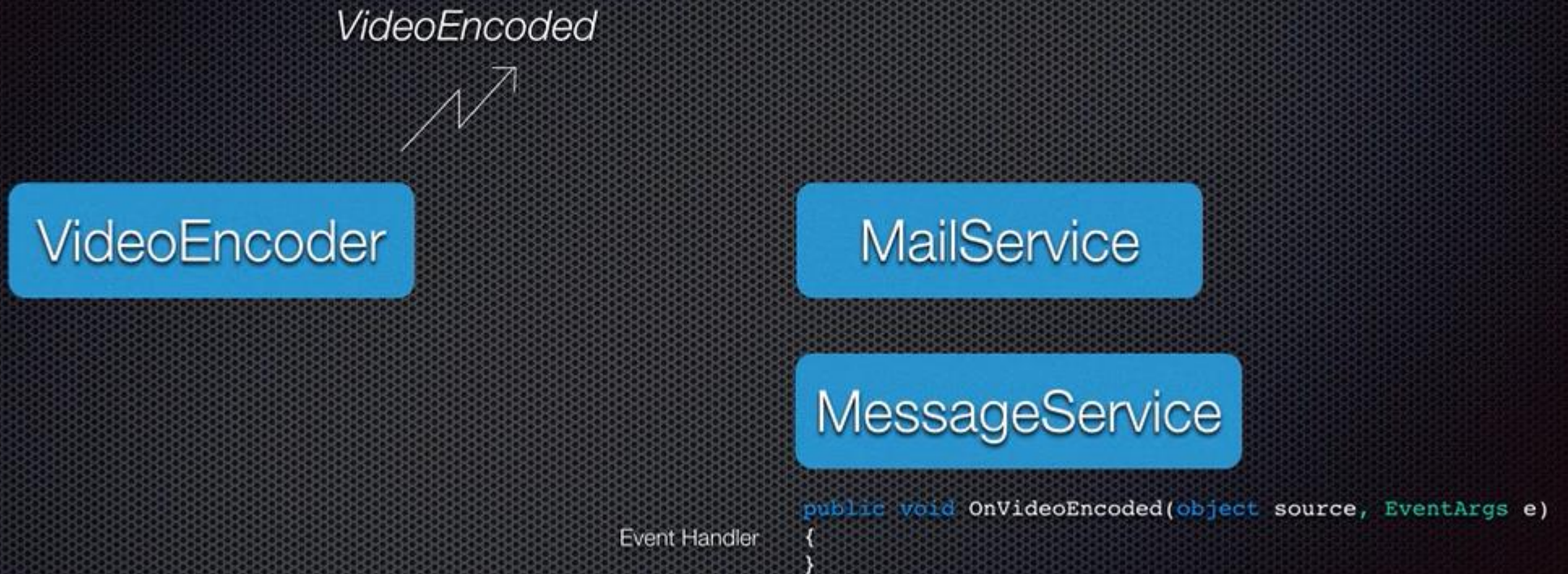


# Delegates

- Agreement / Contract between Publisher and Subscriber
- Determines the signature of the event handler method in Subscriber







EventsAndDelegates.Video

namespace EventsAndDelegates

{

public class Video

{

public string Title { get; set; }

}

}

public class VideoEncoder

{

public void Encode(Video video)

{

Console.WriteLine("Encoding Video...");

Thread.Sleep(3000);

}

}

class Program

{

static void Main(string[] args)

{

var video = new Video() { Title = "Video 1" };

var videoEncoder = new VideoEncoder();

videoEncoder.Encode(video);

}

}

Search Solution Explorer - EventsAndDelegates

Solution 'EventsAndDelegates' (1 project)

EventsAndDelegates

Properties

References

App.config

Program.cs

App.config

Program.cs

Video.cs

VideoEncoder.cs

Properties

References

App.config

Program.cs

Video.cs

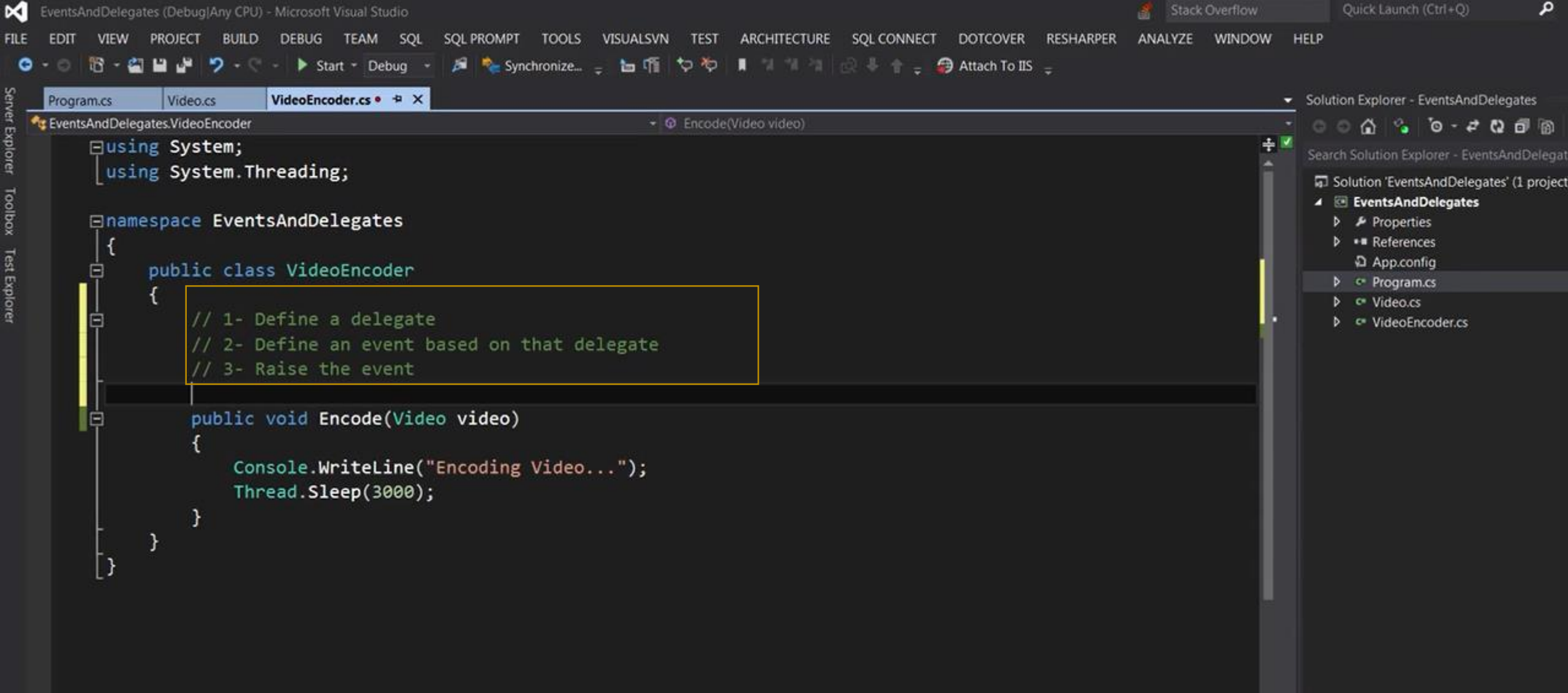
VideoEncoder.cs

C:\Windows\system32\cmd.exe

Encoding Video...







C# Intermediate



**Zouhair Rimale, Ph.D.**

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

box Test Explorer

namespace EventsAndDelegates

{

public class VideoEncoder

{

// 1- Define a delegate I

// 2- Define an event based on that delegate

// 3- Raise the event

public delegate void VideoEncodedEventHandler(object source, EventArgs args);

public void Encode(Video video)

{

Console.WriteLine("Encoding Video...");

Thread.Sleep(3000);

}

}

EventsAndDelegates

Properties

References

App.config

Program.cs

Video.cs

VideoEncoder.cs

C# Intermediate

A circular profile picture of a man with dark hair and a beard, wearing glasses and a dark jacket over a light-colored shirt.

**Zouhair Rimale, Ph.D.**

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

```
public class VideoEncoder
{
    // 1- Define a delegate
    // 2- Define an event based on that delegate
    // 3- Raise the event

    public delegate void VideoEncodedEventHandler(object source, EventArgs args);

    public event VideoEncodedEventHandler VideoEncoded;

    public void Encode(Video video)
    {
        Console.WriteLine("Encoding Video...");
        Thread.Sleep(3000);
    }
}
```

Program.cs  
Video.cs  
VideoEncoder.cs





```

public class VideoEncoder
{
    // 1- Define a delegate
    // 2- Define an event based on that delegate
    // 3- Raise the event

    public delegate void VideoEncodedEventHandler(object source, EventArgs args);

    public event VideoEncodedEventHandler VideoEncoded;

    public void Encode(Video video)
    {
        Console.WriteLine("Encoding Video...");
        Thread.Sleep(3000);

        OnVideoEncoded();
    }

    protected virtual void OnVideoEncoded()
    {
        if (VideoEncoded != null)
            VideoEncoded(this, EventArgs.Empty);
    }
}

```

App.config  
 Program.cs  
 Video.cs  
 VideoEncoder.cs



```

public class VideoEncoder
{
    // 1- Define a delegate
    // 2- Define an event based on that delegate
    // 3- Raise the event

    public delegate void VideoEncodedEventHandler(object source, EventArgs args);

    public event VideoEncodedEventHandler VideoEncoded;

    public void Encode(Video video)
    {
        Console.WriteLine("Encoding Video...");
        Thread.Sleep(3000);

        OnVideoEncoded();
    }

    protected virtual void OnVideoEncoded()
    {
        if (VideoEncoded != null)
            VideoEncoded(this, EventArgs.Empty);
    }
}

```

App.config  
 Program.cs  
 Video.cs  
 VideoEncoder.cs



```
namespace EventsAndDelegates
{
    class Program
    {
        static void Main(string[] args)
        {
            var video = new Video() { Title = "Video 1" };
            var videoEncoder = new VideoEncoder(); //publisher
            var mailService = new MailService(); //subscriber

            videoEncoder.VideoEncoded += mailService.OnVideoEncoded;

            videoEncoder.Encode(video);
        }
    }

    public class MailService
    {
        public void OnVideoEncoded(object source, EventArgs e)
        {
            Console.WriteLine("MailService: Sending an email...");
        }
    }
}
```

EventsAndDelegates

- Properties
- References
- App.config
- Program.cs
- Video.cs
- VideoEncoder.cs

C:\Windows\system32\cmd.exe

```
Encoding Video...
MailService: Sending an email...
Press any key to continue . . .
```



```
public class VideoEncoder
{
    // 1- Define a delegate
    // 2- Define an event based on that delegate
    // 3- Raise the event

    public delegate void VideoEncodedEventHandler(object source, EventArgs args);

    public event VideoEncodedEventHandler VideoEncoded;

    public void Encode(Video video)
    {
        Console.WriteLine("Encoding Video...");
        Thread.Sleep(3000);

        OnVideoEncoded();
    }

    protected virtual void OnVideoEncoded()
    {
        if (VideoEncoded != null)
        {
            VideoEncoded(this, EventArgs.Empty);
        }
    }
}
```

App.config  
Program.cs  
Video.cs  
VideoEncoder.cs



```
namespace EventsAndDelegates
{
    class Program
    {
        static void Main(string[] args)
        {
            var video = new Video() { Title = "Video 1" };
            var videoEncoder = new VideoEncoder(); //publisher
            var mailService = new MailService(); //subscriber

            videoEncoder.VideoEncoded += mailService.OnVideoEncoded;

            videoEncoder.Encode(video);
        }
    }

    public class MailService
    {
        public void OnVideoEncoded(object source, EventArgs e)
        {
            Console.WriteLine("MailService: Sending an email...");
        }
    }
}
```

EventsAndDelegates

- Properties
- References
- App.config
- Program.cs
- Video.cs
- VideoEncoder.cs

C:\Windows\system32\cmd.exe

```
Encoding Video...
MailService: Sending an email...
Press any key to continue . . .
```



**Zouhair Rimale, Ph.D.**

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



```
namespace EventsAndDelegates
{
    public class MailService
    {
        public void OnVideoEncoded(object source, EventArgs e)
        {
            Console.WriteLine("MailService: Sending an email...");
        }
    }
}
```

Solution 'EventsAndDelegates' (1 project)

- EventsAndDelegates
  - Properties
  - References
  - App.config
  - MailService.cs
  - Program.cs
  - Video.cs
  - VideoEncoder.cs

```
namespace EventsAndDelegates
{
    public class MessageService
    {
        public void OnVideoEncoded(object source, EventArgs args)
        {
            Console.WriteLine("MessageService: Sending a text message...");
        }
    }
}
```

Solution 'EventsAndDelegates' (1 project)

- EventsAndDelegates
  - Properties
  - References
  - App.config
  - MailService.cs
  - MessageService.cs
  - Program.cs
  - Video.cs
  - VideoEncoder.cs



```
class Program
{
    static void Main(string[] args)
    {
        var video = new Video() { Title = "Video 1" };
        var videoEncoder = new VideoEncoder(); //publisher
        var mailService = new MailService(); //subscriber
        var messageService = new MessageService(); //subscriber

        videoEncoder.VideoEncoded += mailService.OnVideoEncoded;
        videoEncoder.VideoEncoded += messageService.OnVideoEncoded;

        videoEncoder.Encode(video);
    }
}
```

- Properties
- References
- App.config
- MailService.cs
- MessageService.cs
- Program.cs
- Video.cs
- VideoEncoder.cs

```
C:\Windows\system32\cmd.exe

Encoding Video...
MailService: Sending an email...
MessageService: Sending a text message...
Press any key to continue . . .
```





```

public class VideoEncoder
{
    // 1- Define a delegate
    // 2- Define an event based on that delegate
    // 3- Raise the event

    public delegate void VideoEncodedEventHandler(object source, EventArgs args);

    public event VideoEncodedEventHandler VideoEncoded;

    public void Encode(Video video)
    {
        Console.WriteLine("Encoding Video...");
        Thread.Sleep(3000);

        OnVideoEncoded();
    }

    protected virtual void OnVideoEncoded()
    {
        if (VideoEncoded != null)
            VideoEncoded(this, EventArgs.Empty);
    }
}

```

- App.config
- MailService.cs
- MessageService.cs
- Program.cs
- Video.cs
- VideoEncoder.cs





C# Intermediate



**Zouhair Rimale, Ph.D.**

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

```
it Explorer
public class VideoEncoder
{
    // 1- Define a delegate
    // 2- Define an event based on that delegate
    // 3- Raise the event

    public delegate void VideoEncodedEventHandler(object source, EventArgs args);

    public event VideoEncodedEventHandler VideoEncoded;

    public void Encode(Video video)
    {
        Console.WriteLine("Encoding Video...");
        Thread.Sleep(3000);

        OnVideoEncoded();
    }

    protected virtual void OnVideoEncoded()
    {
        if (VideoEncoded != null)
            VideoEncoded(this, EventArgs.Empty);
    }
}

Solution Explorer
  App.config
  MailService.cs
  MessageService.cs
  Program.cs
  Video.cs
  VideoEncoder.cs
```



```
public class VideoEventArgs : EventArgs
{
    public Video Video { get; set; }
}
```

```
public class VideoEncoder
{
    // 1- Define a delegate
    // 2- Define an event based on that delegate
    // 3- Raise the event

    public delegate void VideoEncodedEventHandler(object source, VideoEventArgs args);

    public event VideoEncodedEventHandler VideoEncoded;

    public void Encode(Video video)
    {
        Console.WriteLine("Encoding Video...");
        Thread.Sleep(3000);

        OnVideoEncoded(video);
    }

    protected virtual void OnVideoEncoded(Video video)
    {
        if (VideoEncoded != null)
            VideoEncoded(this, new VideoEventArgs() { Video = video });
    }
}
```

- App.config
- MailService.cs
- MessageService.cs
- Program.cs
- Video.cs
- VideoEncoder.cs

- References
- App.config
- MailService.cs
- MessageService.cs
- Program.cs
- Video.cs
- VideoEncoder.cs



```
namespace EventsAndDelegates
{
    public class MailService
    {
        public void OnVideoEncoded(object source, VideoEventArgs e)
        {
            Console.WriteLine("MailService: Sending an email..." + e.Video.Title);
        }
    }
}
```

EventsAndDelegates

- Properties
- References
- App.config
- MailService.cs
- MessageService.cs
- Program.cs
- Video.cs
- VideoEncoder.cs

```
namespace EventsAndDelegates
{
    public class MessageService
    {
        public void OnVideoEncoded(object source, VideoEventArgs args)
        {
            Console.WriteLine("MessageService: Sending a text message..." + args.Video.Title);
        }
    }
}
```

EventsAndDelegates

- Properties
- References
- App.config
- MailService.cs
- MessageService.cs
- Program.cs
- Video.cs
- VideoEncoder.cs

```
C:\Windows\system32\cmd.exe
Encoding Video...
MailService: Sending an email...Video 1
MessageService: Sending a text message...Video 1
Press any key to continue . . .
```





```
Test Explorer
public class VideoEncoder
{
    // 1- Define a delegate
    // 2- Define an event based on that delegate
    // 3- Raise the event

    public delegate void VideoEncodedEventHandler(object source, VideoEventArgs args);

    public event VideoEncodedEventHandler VideoEnc

    public void Encode(Video video)
    {
        Console.WriteLine("Encoding Video...");
        Thread.Sleep(3000);

        OnVideoEncoded(video);
    }

    protected virtual void OnVideoEncoded(Video video)
    {
        if (VideoEncoded != null)
            VideoEncoded(this, new VideoEventArgs(){ Video = video });
    }
}
```

class System.Object  
Supports all classes in the .NET Framework class hierarchy and provides low-level services to derived classes. This is the ultimate base class of all classes in the .NET Framework.

- References
- App.config
- MailService.cs
- MessageService.cs
- Program.cs
- Video.cs
- VideoEncoder.cs





```
public class VideoEncoder
{
    // 1- Define a delegate
    // 2- Define an event based on that delegate
    // 3- Raise the event

    // EventHandler
    // EventHandler<TEventArgs>

    public event EventHandler<VideoEventArgs> VideoEncoded;

    public void Encode(Video video)
    {
        Console.WriteLine("Encoding Video...");
        Thread.Sleep(3000);

        OnVideoEncoded(video);
    }

    protected virtual void OnVideoEncoded(Video video)
    {
        if (VideoEncoded != null)
            VideoEncoded(this, new VideoEventArgs() { Video = video });
    }
}
```

- References
- App.config
- MailService.cs
- MessageService.cs
- Program.cs
- Video.cs
- VideoEncoder.cs

```
C:\Windows\system32\cmd.exe
Encoding Video...
MailService: Sending an email...Video 1
MessageService: Sending a text message...Video 1
Press any key to continue . . .
```



# Extension Methods



**Zouhair Rimale, Ph.D.**

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

# What are Extension Methods?

Allow us to add methods to an existing class without

- changing its source code, or
- creating a new class that inherits from it



# Demo Extension Methods



**Zouhair Rimale, Ph.D.**

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



```
Program.cs
ExtensionMethods.RichString

using System;

namespace ExtensionMethods
{
    class Program
    {
        static void Main(string[] args)
        {
            string post = "This is supposed to be a very long blog post blah blah blah...";
        }
    }

    public class RichString : String
    {
    }
}
```



```
tem;
```

```
ExtensionMethods
```

```
Program
```

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

```
    string post = "This is supposed to be a very long blog post blah blah blah...";  
    var shortenedPost = post.Shorten(5);
```

I



```
using System;
using System.Linq;

namespace ExtensionMethods
{
    public static class StringExtensions
    {
        public static string Shorten(this String str, int numberOfWords)
        {
            if (numberOfWords < 0)
                throw new ArgumentOutOfRangeException("numberOfWords should be greater than or equal to 0.");

            if (numberOfWords == 0)
                return "";

            var words = str.Split(' ');

            if (words.Length <= numberOfWords)
                return str;

            return string.Join(" ", words.Take(numberOfWords)) + "...";
        }
    }
}
```



```
Program.cs • StringExtensions.cs •
ExtensionMethods.Program Main(string[] args)

using System;

namespace ExtensionMethods
{
    class Program
    {
        static void Main(string[] args)
        {
            string post = "This is supposed to be a very long blog post blah blah blah...";
            var shortenedPost = post.Shorten(5);

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





# LINQ



**Zouhair Rimale, Ph.D.**

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

# You can query

- Objects in memory, eg collections (*LINQ to Objects*)
- Databases (*LINQ to Entities*)
- XML (*LINQ to XML*)
- ADO.NET Data Sets (*LINQ to Data Sets*)



# Demo LINQ



**Zouhair Rimale, Ph.D.**

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

```
namespace Linq
{
    public class Book
    {
        public string Title { get; set; }
        public float Price { get; set; }
    }
}
```

```
namespace Linq
{
    public class BookRepository
    {
        public IEnumerable<Book> GetBooks()
        {
            return new List<Book>
            {
                new Book() {Title = "ADO.Net Step by Step", Price = 5 },
                new Book() {Title = "ASP.NET MVC", Price = 9.99f },
                new Book() {Title = "ASP.NET Web API", Price = 12 },
                new Book() {Title = "C# Advanced Topics", Price = 7 },
                new Book() {Title = "C# Advanced Topics", Price = 9 }
            };
        }
    }
}
```





```
namespace Linq
{
    class Program
    {
        static void Main(string[] args)
        {
            var books = new BookRepository().GetBooks();

            var cheapBooks = new List<Book>();
            foreach (var book in books)
            {
                if (book.Price < 10)
                    cheapBooks.Add(book);
            }

            foreach (var book in cheapBooks)
                Console.WriteLine(book.Title + " " + book.Price);
        }
    }
}
```



```
class Program
{
    static void Main(string[] args)
    {
        var books = new BookRepository().GetBooks();

        // LINQ Query Operators
        var cheaperBooks =
            from b in books
            where b.Price < 10
            orderby b.Title
            select b;

        // LINQ Extension Methods
        var cheapBooks = books
            .Where(b => b.Price < 10)
            .OrderBy(b => b.Title)
            .Select(b => b.Title);

        foreach (var book in cheapBooks)
        {
            Console.WriteLine(book);
            //Console.WriteLine(book.Title + " " + book.Price);
        }
    }
}
```

(range variable) Book b



```
static void Main(string[] args)
{
    var books = new BookRepository().GetBooks();

    // LINQ Extension Methods
    var book = books.Single(b => b.Title == "ASP.NET MVC");

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

```
static void Main(string[] args)
{
    var books = new BookRepository().GetBooks();

    // LINQ Extension Methods
    var book = books.FirstOrDefault(b => b.Title == "C# Advanced Topics");

    Console.WriteLine(book.Title + " " + book.Price);
}
```

```
static void Main(string[] args)
{
    var books = new BookRepository().GetBooks();

    // LINQ Extension Methods
    var pagedBooks = books.Skip(2).Take(3);

    foreach (var pagedBook in pagedBooks)
    {
        Console.WriteLine(pagedBook.Title);
    }
}
```

```
static void Main(string[] args)
{
    var books = new BookRepository().GetBooks();

    // LINQ Extension Methods
    var maxPrice = books.Max(b => b.Price);
    var minPrice = books.Min(b => b.Price);

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





```
static void Main(string[] args)
{
    var books = new BookRepository().GetBooks();

    // LINQ Extension Methods
    books.Where();
    books.Single();
    books.SingleOrDefault();

    books.First();
    books.FirstOrDefault();

    books.Last();
    books.LastOrDefault();

    books.Min();
    books.Max();
    books.Count();
    books.Sum();
    books.Average(b => b.Price);

    books.Skip(5).Take(3);
}
```



# Nullable Types



**Zouhair Rimale, Ph.D.**

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

# Value Types

- Cannot be null
- `bool hasAccess = true; // or false`



# Database

Customers.Birthday (datetime NULL)





# Demo Nullable Types



**Zouhair Rimale, Ph.D.**

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



```
class Program
{
    static void Main(string[] args)
    {
        DateTime date = null;
    }
}
```

Cannot convert source type 'null' to target type 'System.DateTime'

```
static void Main(string[] args)
{
    Nullable<DateTime> date = null;
}
```

```
DateTime? date = null;

Console.WriteLine("GetValueOrDefault(): " + date.GetValueOrDefault());
Console.WriteLine("HasValue: " + date.HasValue);
Console.WriteLine("Value: " + date.Value);
```

```
C:\Windows\system32\cmd.exe
GetValueOrDefault(): 1/01/0001 12:00:00 AM
HasValue: False

Unhandled Exception: System.InvalidOperationException: Nullable object must have a value.
   at System.ThrowHelper.ThrowInvalidOperationException(ExceptionResource resource)
   at System.Nullable`1.get_Value()
   at NullableTypes.Program.Main(String[] args) in c:\Projects\NullableTypes\NullableTypes\Program.cs:line 14
```



Test Explorer

```
class Program
```

```
{
```

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

```
    {
```

```
        DateTime? date = new DateTime(2014, 1, 1);
```

```
        DateTime date2 = date.GetValueOrDefault();
```

```
        DateTime? date3 = date2;
```

```
        Console.WriteLine(date3.GetValueOrDefault());
```

```
    }
```

```
}
```

# Null Coalescing Operator



```
static void Main(string[] args)
{
    DateTime? date = null;
    DateTime date2;

    if (date != null)
        date2 = date.GetValueOrDefault();
    else
        date2 = DateTime.Today;

    Console.WriteLine(date2);
}
```

```
static void Main(string[] args)
{
    DateTime? date = null;
    DateTime date2 = date ?? DateTime.Today;

    DateTime date3 = (date != null) ? date.GetValueOrDefault() : DateTime.Today;

    Console.WriteLine(date2);
}
```





# Dynamic



**Zouhair Rimale, Ph.D.**

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

# Programming Languages

- Statically-typed languages: C#, Java
- Dynamically-typed languages: Ruby, Javascript, Python



# Type resolution

- Static languages: at compile-time
- Dynamic languages: at run-time



# Benefits

- Static languages: early feedback (compile-time)
- Dynamic languages: easier and faster to code





# C# History

- Started as a static language
- .NET 4 added the dynamic capability, to improve interoperability with
  - COM (eg writing office applications)
  - Dynamic languages (IronPython)



# Demo Dynamic



**Zouhair Rimale, Ph.D.**

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

```
{
class Program
{
    static void Main(string[] args)
    {
        object obj = "Mosh";
        //obj.GetHashCode()

        var methodInfo = obj.GetType().GetMethod("GetHashCode");
        methodInfo.Invoke(null, null);
    }
}
```



```
static void Main(string[] args)
{
    object obj = "Mosh";
    //obj.GetHashCode()

    var methodInfo = obj.GetType().GetMethod("GetHashCode");
    methodInfo.Invoke(null, null);

    object excelObject = "mosh";
    excelObject.Optimize();
}
```

```
static void Main(string[] args)
{
    object obj = "Mosh";
    //obj.GetHashCode()

    var methodInfo = obj.GetType().GetMethod("GetHashCode");
    methodInfo.Invoke(null, null);

    dynamic excelObject = "mosh";
    excelObject.Optimize();
}
```







Test Explorer

```
class Program
{
    static void Main(string[] args)
    {
        dynamic name = "Mosh";
        name++;
    }
}
```

Local variable 'name' is only assigned but its value is never used

```
C:\Windows\system32\cmd.exe

Unhandled Exception: Microsoft.CSharp.RuntimeBinder.RuntimeBinderException: Oper
ator '++' cannot be applied to operand of type 'string'
   at CallSite.Target(Closure , CallSite , Object )
   at System.Dynamic.UpdateDelegates.UpdateAndExecute1[T0,TRet](CallSite site, T
0 arg0)
   at DynamicBinding.Program.Main(String[] args) in c:\Projects\DynamicBinding\D
ynamicBinding\Program.cs:line 9
```



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

```
{
```

```
    dynamic a = 10;
```

```
    dynamic b = 5;
```

```
    var c = a + b;
```

```
}
```

```
}
```

dynamic

Represents an object whose operations will be resolved at runtime.



namespace DynamicBinding

{

class Program

{

static void Main(string[] args)

{

int i = 5;

dynamic d = i;

}

}

}

133 %

Watch 1

Name	Value	Type
a	The name 'a' does not exist in the current context.	
b	The name 'b' does not exist in the current context.	
c	The name 'c' does not exist in the current context.	
d	5	dynamic (int)

Call Stack

Name	Language
DynamicBinding.exe!DynamicBinding.Program.Main(string[] args) Line 10	C#
[External Code]	

All Projects (C#)

No code analysis detected.



# Exception Handling



**Zouhair Rimale, Ph.D.**

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



```

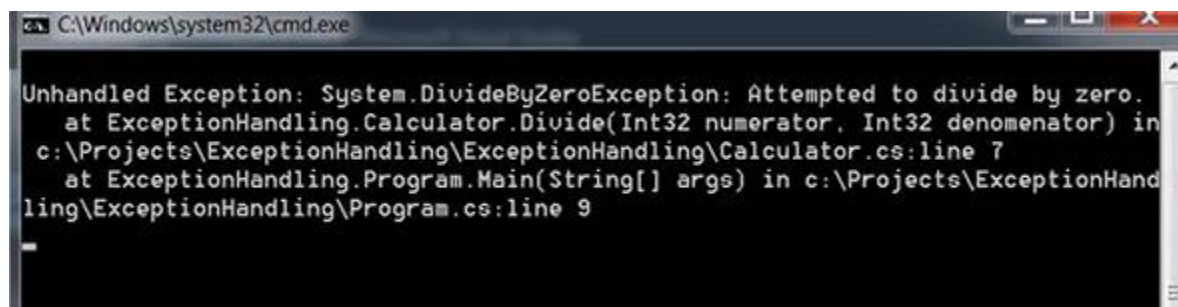
namespace ExceptionHandling
{
    public class Calculator
    {
        public int Divide(int numerator, int denominator)
        {
            return numerator / denominator;
        }
    }
}

```

```

class Program
{
    static void Main(string[] args)
    {
        var calculator = new Calculator();
        var result = calculator.Divide(5, 0);
    }
}

```



```

C:\Windows\system32\cmd.exe

Unhandled Exception: System.DivideByZeroException: Attempted to divide by zero.
   at ExceptionHandling.Calculator.Divide(Int32 numerator, Int32 denominator) in
c:\Projects\ExceptionHandling\ExceptionHandling\Calculator.cs:line 7
   at ExceptionHandling.Program.Main(String[] args) in c:\Projects\ExceptionHand
ling\ExceptionHandling\Program.cs:line 9

```



```
namespace ExceptionHandling
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                var calculator = new Calculator();
                var result = calculator.Divide(5, 0);
            }
            catch (Exception)
            {
                Console.WriteLine("Sorry, an unexpected error occurred.");
            }
        }
    }
}
```



## e ExceptionHandling

### s Program

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

```
{
```

```
try
```

```
{
```

```
    var calculator = new Calculator();
```

```
    var result = calculator.Divide(5, 0);
```

```
}
```

```
catch (Exception ex)
```

```
{
```

```
    Console.WriteLine
```

```
}
```

```
}
```

ex ["Attempted to divide by zero."] ⇄

[System.DivideByZeroException]

Data

HelpLink

HResult

InnerException

Message

Source

StackTrace

TargetSite

Static members

Non-Public members

("Attempted to divide by zero.")

(System.Collections.ListDictionaryInternal)

null

-2147352558

null

Q - "Attempted to divide by zero."

Q - "ExceptionHandling"

Q - " at ExceptionHandling.Calculator.Divide(Int32 numerator, Int32 denominator) in c:\\Projects\\ExceptionHandling\\ExceptionHan

(Int32 Divide(Int32, Int32))

Value

Type

Name

ExceptionHandling.exe!ExceptionHandling.Program.Main(string[] args) Line 17



```

    try
    {
        var calculator = new Calculator();
        var result = calculator.Divide(5, 0);
    }
    catch (Exception ex)
    {
        // Handle exception
    }
}

```

**DivideByZeroException**

- ▷ GC
- ▷ GCCollectionMode
- ▷ GCNotificationStatus
- ▷ Guid
- ▷ IAppDomainSetup
- ▷ IAsyncResult
- ▷ ICloneable

public class **DivideByZeroException** : [System.ArithmeticException](#)

Member of [System](#)

**Summary:**

The exception that is thrown when there is an attempt to divide an integral or decimal value by zero.

- ▷ ConsoleKeyInfo
- ▷ ConsoleModifiers
- ▷ ConsoleSpecialKey
- ▷ ContextBoundObject
- ▷ ContextMarshalException
- ▷ ContextStaticAttribute

public class **ArithmeticException** : [System.SystemException](#)

Member of [System](#)

**Summary:**

- ▷ SByte
- ▷ SerializableAttribute
- ▷ Single
- ▷ StackOverflowException
- ▷ STAThreadAttribute
- ▷ String
- ▷ StringComparer

public class **SystemException** : [System.Exception](#)

Member of [System](#)

**Summary:**

Defines the base class for predefined exceptions in the System namespace.



```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            var calculator = new Calculator();
            var result = calculator.Divide(5, 0);
        }
        catch (DivideByZeroException ex)
        {
            Console.WriteLine("You cannot divide by 0.");
        }
        catch (ArithmeticException ex)
        {
        }
        catch (Exception ex)
        {
            Console.WriteLine("Sorry, an unexpected error occurred.");
        }
    }
}
```





```
{
class Program
{
    static void Main(string[] args)
    {
        StreamReader streamReader = null;
        try
        {
            streamReader = new StreamReader(@"c:\file.zip");
            var content = streamReader.ReadToEnd();
            throw new Exception("Oops");
        }
        catch (Exception ex)
        {
            Console.WriteLine("Sorry, an unexpected error occurred.");
        }
        finally
        {
            if (streamReader != null)
                streamReader.Dispose();
        }
    }
}
```



```

class Program
{
    static void Main(string[] args)
    {
        try
        {
            using (var streamReader = new StreamReader(@"c:\file.zip"))
            {
                var content = streamReader.ReadToEnd();
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine("Sorry, an unexpected error occurred.");
        }
    }
}

```



```
namespace ExceptionHandling
{
    public class YouTubeApi
    {
        public List<Video> GetVideos(string user)
        {
            try
            {
                // Access YouTube web service
                // Read the data
                // Create a list of Video objects
            }
            catch (Exception ex)
            {
                throw;
            }

            return new List<Video>();
        }
    }
}
```



```

namespace ExceptionHandling
{
    public class YouTubeException : Exception
    {
        public YouTubeException(string message, Exception innerException)
        : base(message, innerException)
        {
        }
    }
}

```

```

public List<Video> GetVideos(string user)
{
    try
    {
        // Access YouTube web service
        // Read the data
        // Create a list of Video objects
        throw new Exception("Oops some low level YouTube error.");
    }
    catch (Exception ex)
    {
        // Log

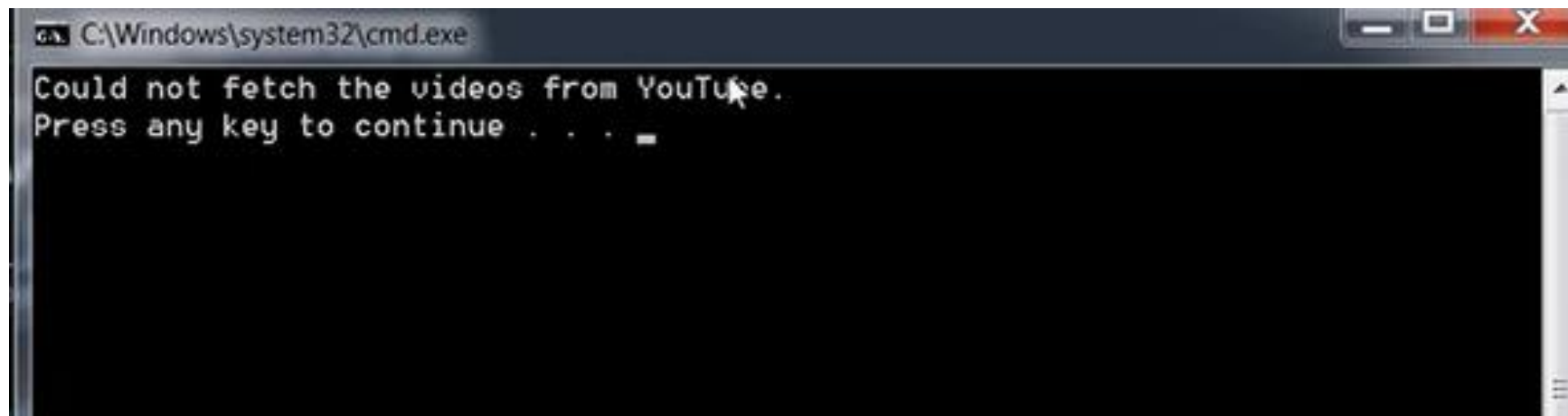
        throw new YouTubeException("Could not fetch the videos from YouTube.", ex);
    }

    return new List<Video>();
}

```



```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            var api = new YouTubeApi();
            var videos = api.GetVideos("mosh");
        }
        catch (Exception ex)
        {
            Console.WriteLine(ex.Message);
        }
    }
}
```



A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Windows\system32\cmd.exe'. The window contains the following text: 'Could not fetch the videos from YouTube.' followed by 'Press any key to continue . . .'. The text is displayed in a monospaced font on a black background.





```
public List<Video> GetVideos(string user)
{
    try
    {
        // Access YouTube web service
        // Read the data
        // Create a list of Video objects
        throw new Exception("Oops some low level YouTube error.");
    }
    catch (Exception ex)
    {
        // Log

        throw new YouTubeException("Could not fetch the videos from YouTube.", ex);
    }
}
```

133 %

Watch 1		
Name	Value	Type

Call Stack	
Name	Language
ExceptionHandling.exe!ExceptionHandling.YouTubeApi.GetVideos(string user) Line 17	C#
ExceptionHandling.exe!ExceptionHandling.Program.Main(string[] args) Line 14 + 0x10 bytes	C#
[External Code]	

C# Intermediate



**Zouhair Rimale, Ph.D.**

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

# Asynchronous Programming

## “Async / Await”



**Zouhair Rimale, Ph.D.**

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

# Synchronous Program Execution

- Program is executed line by line, one at a time.
- When a function is called, program execution has to wait until the function returns.



# Synchronous Program Execution

1 : ...

2 : **TimeConsumingProcess ( ) ;** 🏢👤

3 : ...

4 : ...



# Asynchronous Program Execution

- When a function is called, program execution continues to the next line, **without** waiting for the function to complete.





# Asynchronous Program Execution

1 : ...

2 : **TimeConsumingProcess ( callback ) ;**

3 : ...

4 : ...

```
callback()  
{  
    ...  
}
```



# So what's the difference?

- Asynchronous programming improves responsiveness.



# Real-world Examples

- Windows Media Player
- Web Browser



# When to use asynchronous?

- Accessing the Web
- Working with files and databases
- Working with images



# How?

- Traditional approaches:
  - Multi-threading
  - Callbacks
- New approach since NET 4.5
  - Async / Await





# Demo Asynchronous Programming

## “Async / Await”



**Zouhair Rimale, Ph.D.**

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

Visual Studio interface showing a Windows application project named 'Async'.

The **Solution Explorer** on the right shows the project structure:

- Solution 'Async' (1 project)
  - Async
    - Properties
    - References
    - App.config
    - App.xaml
    - MainWindow.xaml

The **MainWindow** window displays a simple UI with a single button labeled "Button".

The **Code View** shows the following C# code:

```
private void Button_Click(object sender, RoutedEventArgs e)
{
    DownloadHtml("http://msdn.microsoft.com");
}

public void DownloadHtml(string url)
{
    var webClient = new WebClient();
    var html = webClient.DownloadString(url);

    using (var streamWriter = new StreamWriter(@"c:\projects\result.html"))
    {
        streamWriter.Write(html);
    }
}
```



```
Async.MainWindow
{
    DownloadHtml("http://msdn.microsoft.com");
}
Button_Click

public async Task DownloadHtmlAsync(string url)
{
    var webClient = new WebClient();
    var html = await webClient.DownloadStringTaskAsync(url);

    using (var streamWriter = new StreamWriter(@"c:\projects\result.html"))
    {
        await streamWriter.WriteAsync(html);
    }
}

public void DownloadHtml(string url)
{
    var webClient = new WebClient();
    var html = webClient.DownloadString(url);

    using (var streamWriter = new StreamWriter(@"c:\projects\result.html"))
    {
        streamWriter.Write(html);
    }
}
MainWindow
```

100 %

Search Solution Explorer - Async (Ctrl+;)

Solution 'Async' (1 project)

- Async
  - Properties
  - References
  - App.config
  - App.xaml
  - MainWindow.xaml



MainWindow.xaml    MainWindow.xaml.cs    Async.MainWindow

Button\_Click(object sender, RoutedEventArgs e)

```
public MainWindow()
{
    InitializeComponent();
}

private void Button_Click(object sender, RoutedEventArgs e)
{
    DownloadHtmlAsync("http://m");
}

public async Task DownloadHtmlA
{
    var webClient = new WebClient();
    var html = await webClient.D

    using (var streamWriter = ne
    {
        await streamWriter.Writ
    }
}

public void DownloadHtml(string url)
{
    var webClient = new WebClient();
    var html = webClient.DownloadString(url);

    using (var streamWriter = new StreamWriter(@"c:\projects\result.html"))
```

MainWindow

Button

Solution Explorer - Async

Search Solution Explorer - Async (Ctrl+;)

Solution 'Async' (1 project)

- Async
  - Properties
  - References
  - App.config
  - App.xaml
  - MainWindow.xaml





Async.MainWindow

```
public MainWindow()
{
    InitializeComponent();
}

private void Button_Click(object sender, RoutedEventArgs e)
{
    // DownloadHtml("http://msdn.microsoft.com");

    var html = GetHtml("http://msdn.microsoft.com");
    MessageBox.Show(html.Substring(0, 10));
}

public string GetHtml(string url)
{
    var webClient = new WebClient();

    return webClient.DownloadString(url);
}

public async Task DownloadHtmlAsync(string url)...

public void DownloadHtml(string url)...
```

Search Solution Explorer - Async (Ctrl+)

Solution 'Async' (1 project)

- Async
  - Properties
  - References
  - App.config
  - App.xaml
  - MainWindow.xaml

MainWindow

Button

<!DOCTYPE

OK

C# Intermediate



**Zouhair Rimale, Ph.D.**

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



```
Async.MainWindow
Button_Click(object sender, RoutedEventArgs e)

public MainWindow()
{
    InitializeComponent();
}

private void Button_Click(object sender, RoutedEventArgs e)
{
    // DownloadHtml("http://msdn.microsoft.com");

    var html = GetHtmlAsync("http://msdn.microsoft.com");
    MessageBox.Show(html.Substring(0, 10));
}

public async Task<string> GetHtmlAsync(string url)
{
    var webClient = new WebClient();

    return await webClient.DownloadStringTaskAsync(url);
}

public string GetHtml(string url)
{
    var webClient = new WebClient();

    return webClient.DownloadString(url);
}
```

Search Solution Explorer - Async (Ctrl+)

Solution 'Async' (1 project)

- Async
  - Properties
  - References
  - App.config
  - App.xaml
  - MainWindow.xaml



```
private void Button_Click(object sender, RoutedEventArgs e)
{
    DownloadHtml("http://msdn.microsoft.com");

    var html = await GetHtmlAsync("http://msdn.microsoft.com");
    MessageBox.Show(html.Substring(0, 10));
}
```

```
private async void Button_Click(object sender, RoutedEventArgs e)
{
    DownloadHtml("http://msdn.microsoft.com");

    var html = await GetHtmlAsync("http://msdn.microsoft.com");
    MessageBox.Show(html.Substring(0, 10));
}

public async Task<string> GetHtmlAsync(string url)
{
    var webClient = new WebClient();

    return await webClient.DownloadStringTaskAsync(url);
}
```

