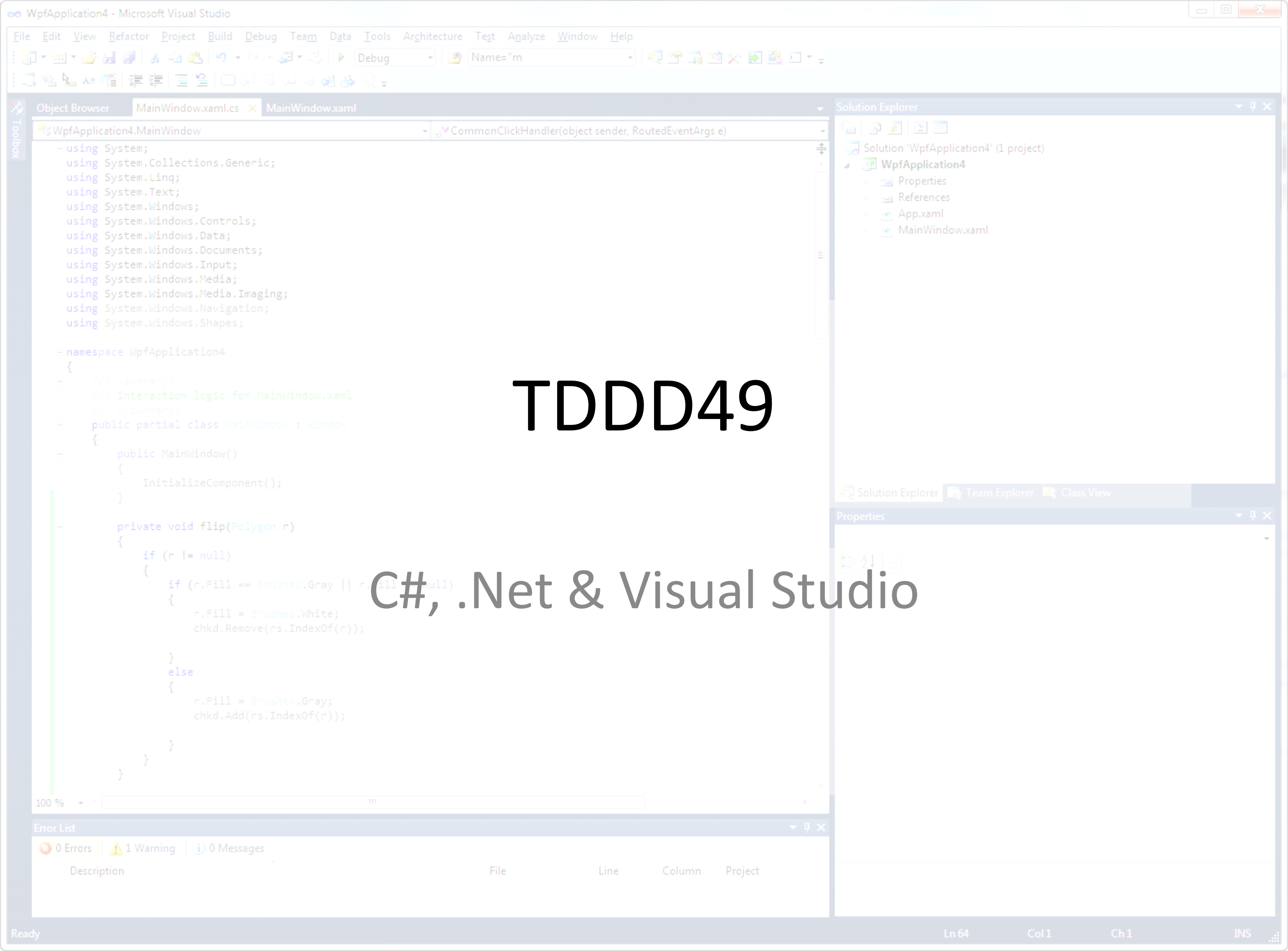
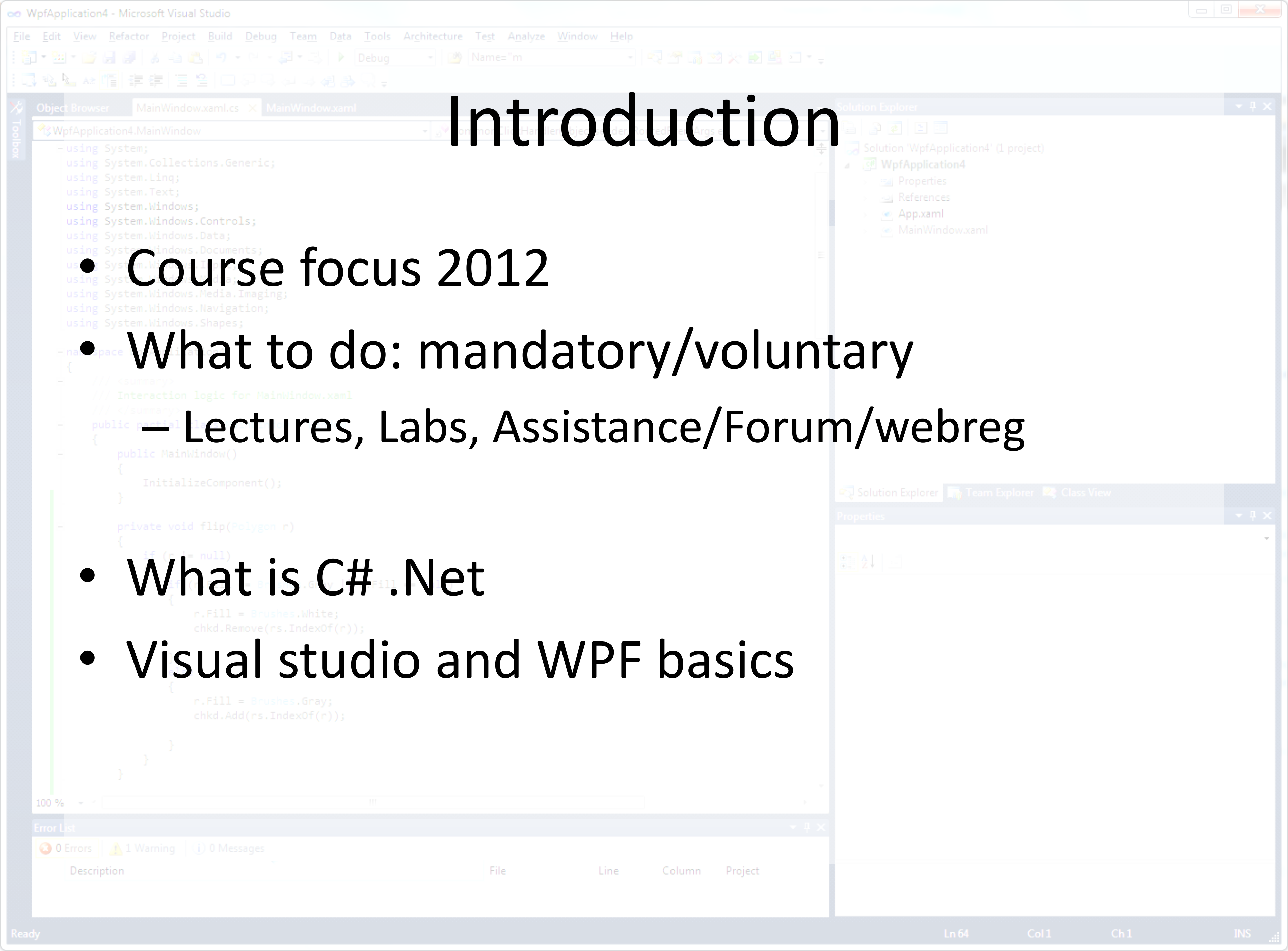
TDDD49

C#, .Net & Visual Studio

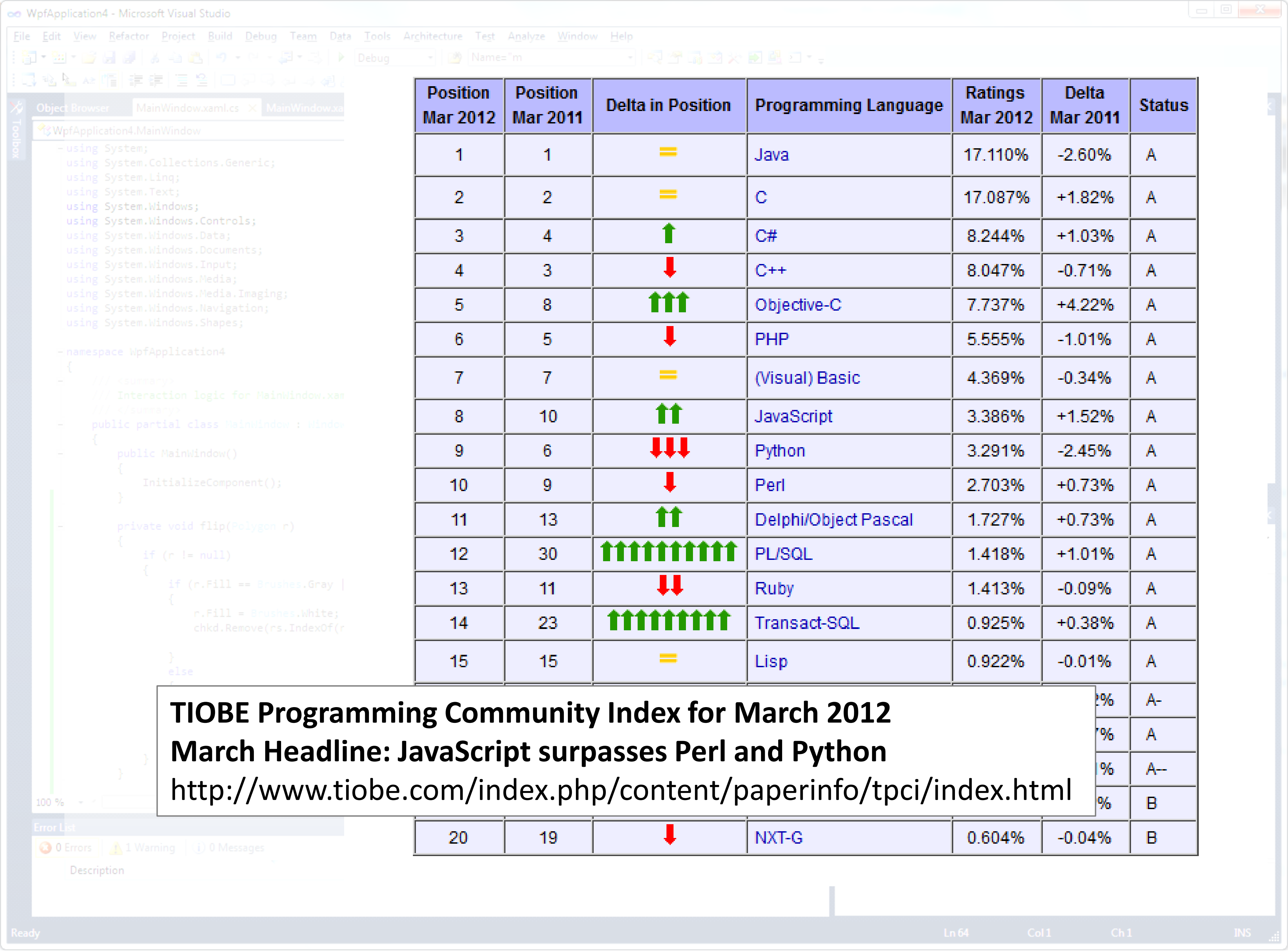


```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace WpfApplication4
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }

        private void flip(Polygon r)
        {
            if (r != null)
            {
                if (r.Fill == Brushes.Gray || r.Fill == null)
                {
                    r.Fill = Brushes.White;
                    chkd.Remove(rs.IndexOf(r));

                }
                else
                {
                    r.Fill = Brushes.Gray;
                    chkd.Add(rs.IndexOf(r));

                }
            }
        }
```



Introduction

- Course focus 2012
- What to do: mandatory/voluntary
  - Lectures, Labs, Assistance/Forum/webreg
- What is C# .Net
- Visual studio and WPF basics


WpfApplication4 - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Team Data Tools Architecture Test Analyze Window Help
Debug
Name="m"
Toolbox
Object Browser
MainWindow.xaml.cs
MainWindow.xa
WpfApplication4.MainWindow
100 %
Error List
0 Errors
1 Warning
0 Messages
Description
Ready
Ln 64
Col 1
Ch 1
INS

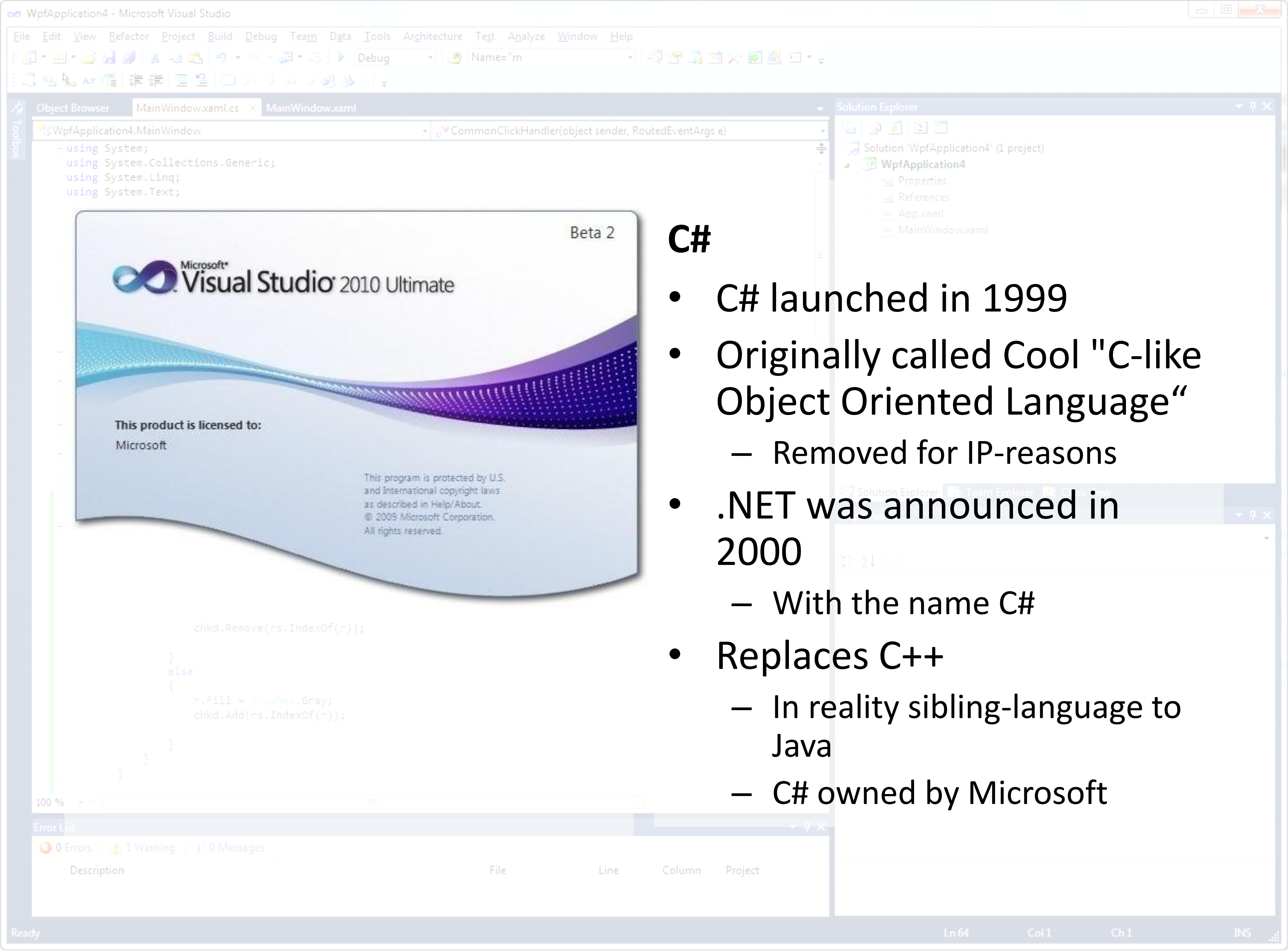
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace WpfApplication4
{
    /// <summary>
    /// Interaction logic for MainWindow.xam
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }

        private void flip(Polygon r)
        {
            if (r != null)
            {
                if (r.Fill == Brushes.Gray |
                {
                    r.Fill = Brushes.White;
                    chkd.Remove(rs.IndexOf(r

                }
                else
```

Position Mar 2012	Position Mar 2011	Delta in Position	Programming Language	Ratings Mar 2012	Delta Mar 2011	Status
1	1	=	Java	17.110%	-2.60%	A
2	2	=	C	17.087%	+1.82%	A
3	4	↑	C#	8.244%	+1.03%	A
4	3	↓	C++	8.047%	-0.71%	A
5	8	↑↑↑	Objective-C	7.737%	+4.22%	A
6	5	↓	PHP	5.555%	-1.01%	A
7	7	=	(Visual) Basic	4.369%	-0.34%	A
8	10	↑↑	JavaScript	3.386%	+1.52%	A
9	6	↓↓↓	Python	3.291%	-2.45%	A
10	9	↓	Perl	2.703%	+0.73%	A
11	13	↑↑	Delphi/Object Pascal	1.727%	+0.73%	A
12	30	↑↑↑↑↑↑↑↑↑↑	PL/SQL	1.418%	+1.01%	A
13	11	↓↓	Ruby	1.413%	-0.09%	A
14	23	↑↑↑↑↑↑↑↑↑	Transact-SQL	0.925%	+0.38%	A
15	15	=	Lisp	0.922%	-0.01%	A
					2%	A-
					7%	A
					1%	A--
					%	B
20	19	↓	NXT-G	0.604%	-0.04%	B

TIOBE Programming Community Index for March 2012
March Headline: JavaScript surpasses Perl and Python
http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

C#

- C# launched in 1999
- Originally called Cool "C-like Object Oriented Language“
  - Removed for IP-reasons
- .NET was announced in 2000
  - With the name C#
- Replaces C++
  - In reality sibling-language to Java
  - C# owned by Microsoft

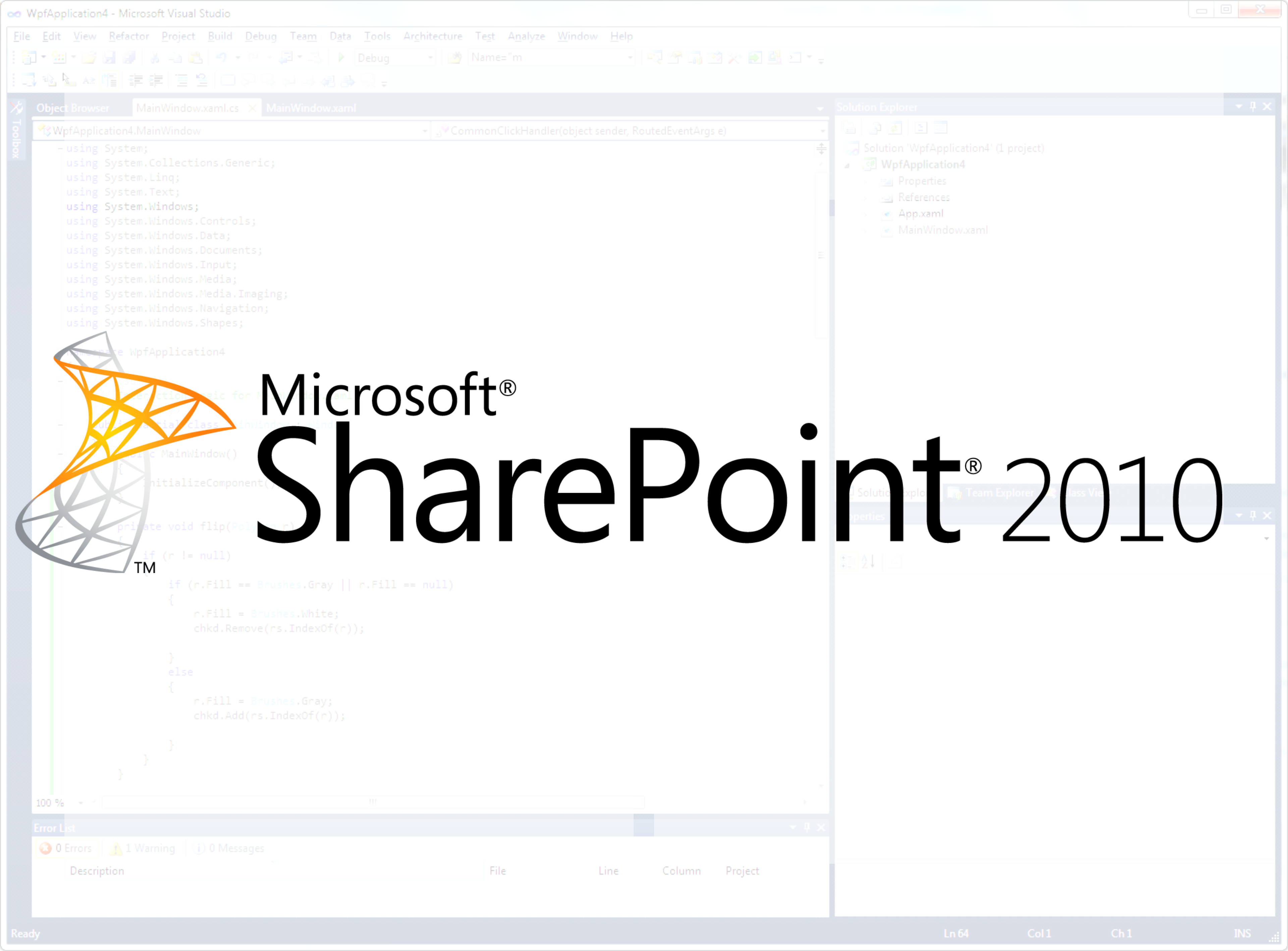


```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
            chkd.Remove(rs.IndexOf(r));
        }
        else
        {
            r.Fill = Brushes.Gray;
            chkd.Add(rs.IndexOf(r));
        }
    }
}
```

This product is licensed to:
Microsoft

This program is protected by U.S. and International copyright laws as described in Help/About.
© 2009 Microsoft Corporation.
All rights reserved.



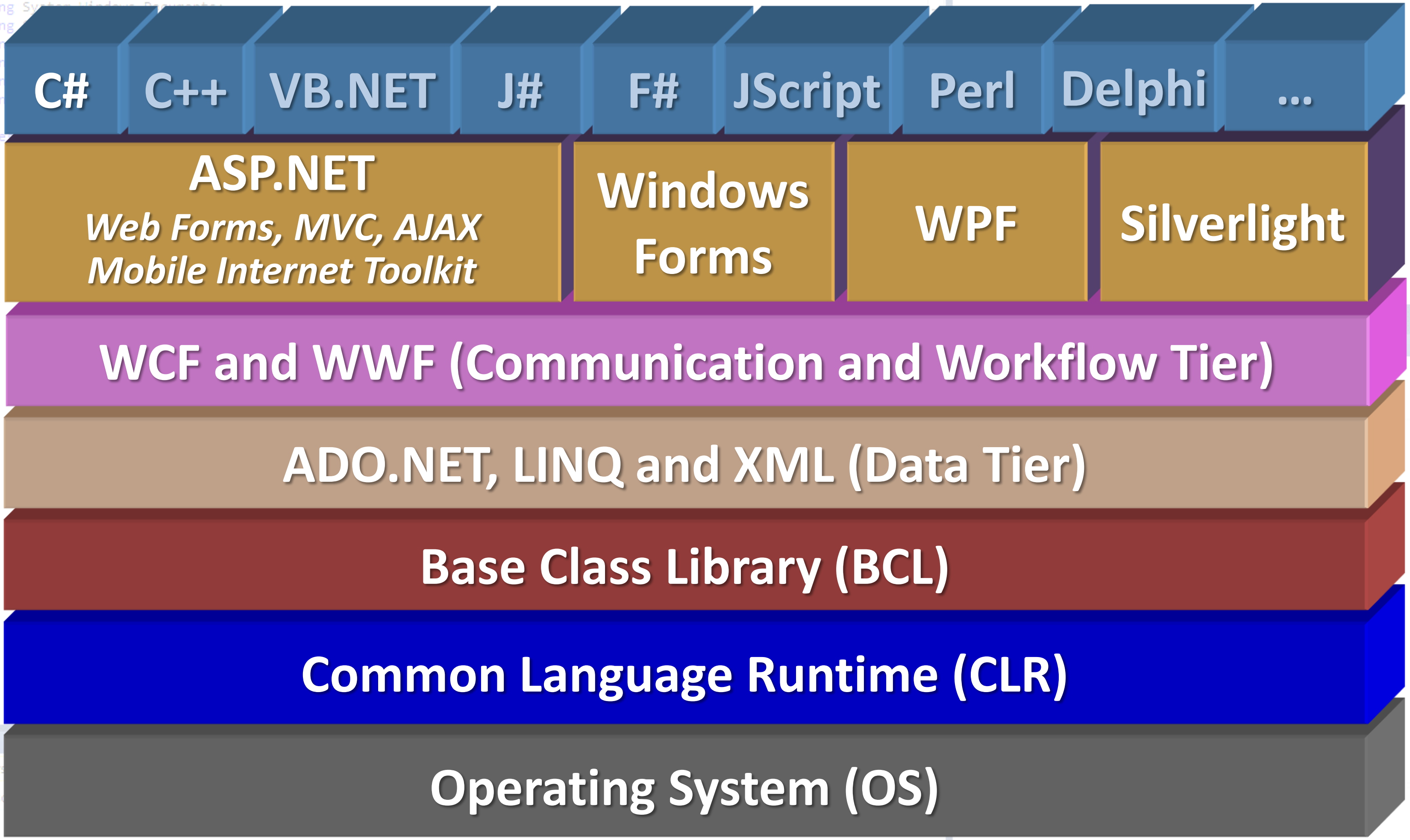
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace WpfApplication4

        /// Interaction logic for MainWindow.xaml

    public partial class MainWindow : Window

        public MainWindow()
        {
            InitializeComponent();
        }

        private void flip(Polygon r)
        {
            if (r != null)
            {
                if (r.Fill == Brushes.Gray || r.Fill == null)
                {
                    r.Fill = Brushes.White;
                    chkd.Remove(rs.IndexOf(r));

                }
                else
                {
                    r.Fill = Brushes.Gray;
                    chkd.Add(rs.IndexOf(r));

                }
            }
        }
```

Microsoft®

SharePoint® 2010

TM

.NET Framework Architecture (7)



C# | C++ | VB.NET | J# | F# | JScript | Perl | Delphi | …

ASP.NET
Web Forms, MVC, AJAX
Mobile Internet Toolkit

Windows Forms

WPF

Silverlight

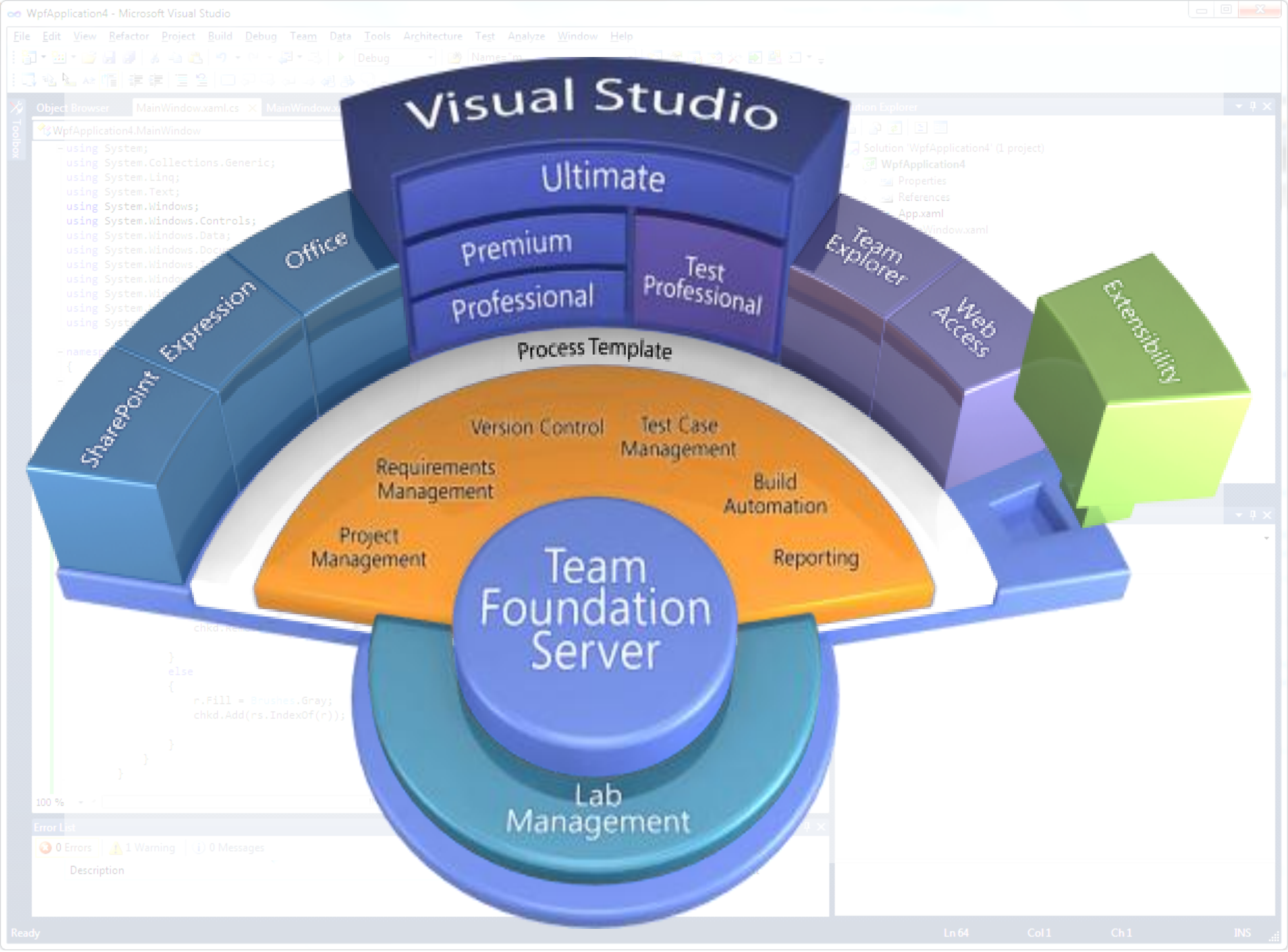
WCF and WWF (Communication and Workflow Tier)

ADO.NET, LINQ and XML (Data Tier)

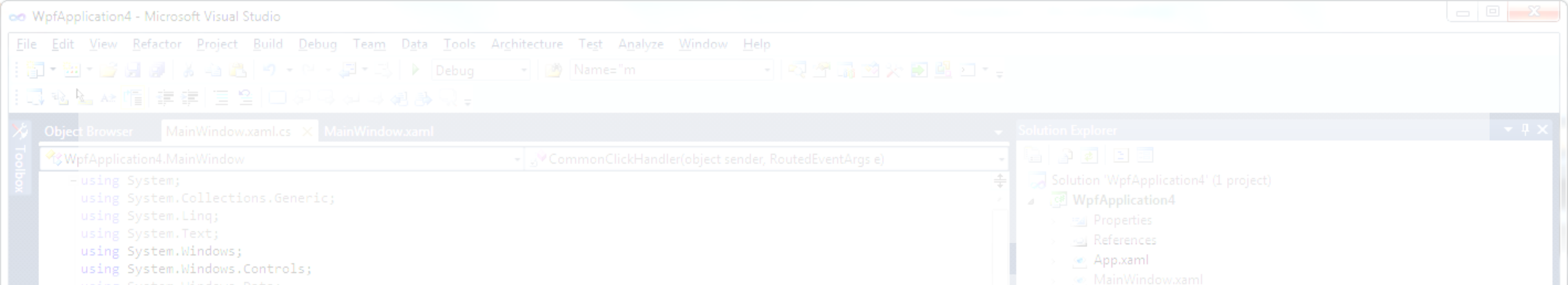
Base Class Library (BCL)

Common Language Runtime (CLR)

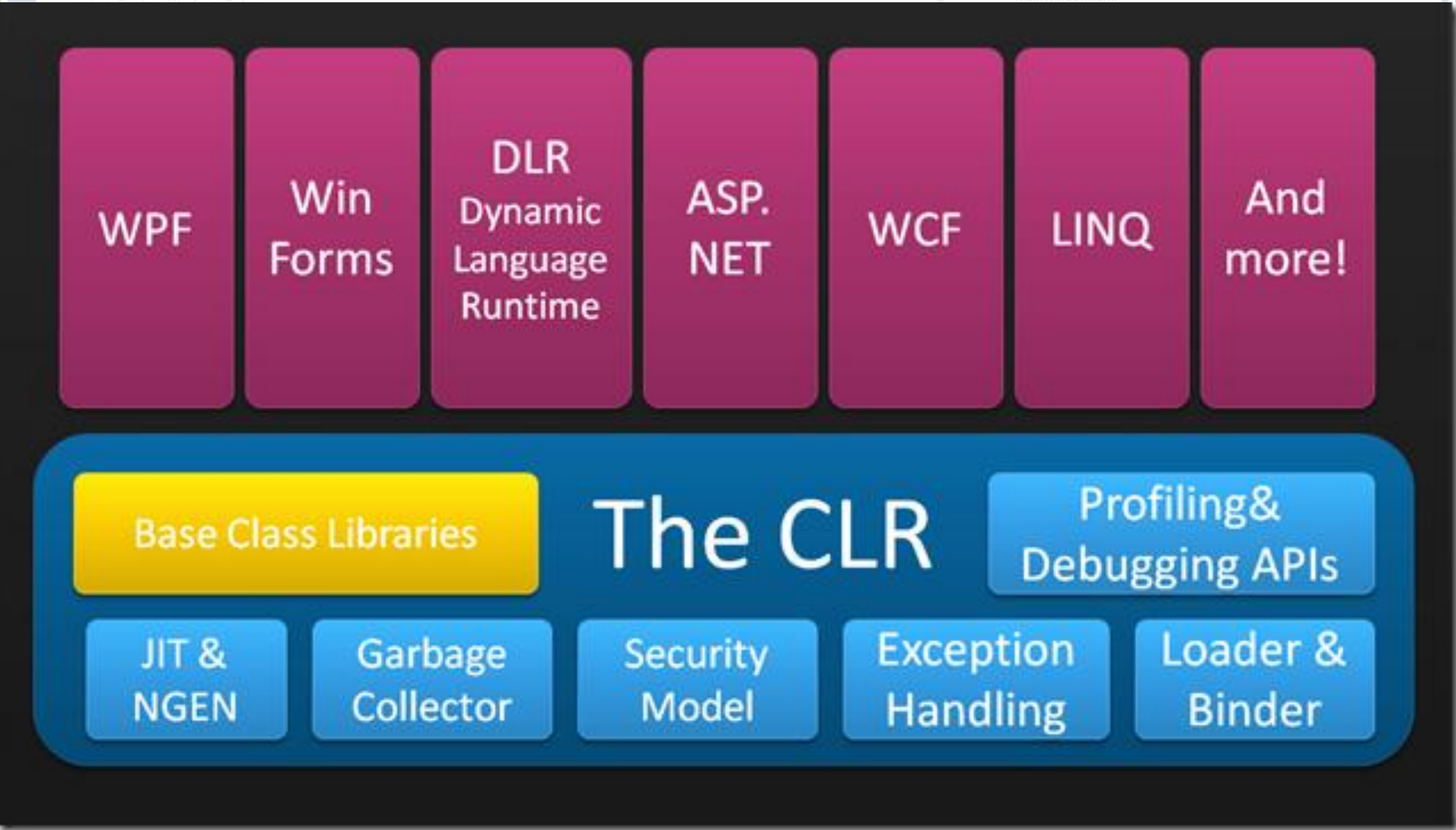
Operating System (OS)



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Docu
using System.Windows.I
using System.Windo
using System.Wi
using System
using Syst

namesp
{

            chkd.Re
        }
        else
        {
            r.Fill = Brushes.Gray;
            chkd.Add(rs.IndexOf(r));
        }
    }
}
```



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
```



WPF

Win Forms

DLR Dynamic Language Runtime

ASP. NET

WCF

LINQ

And more!

Base Class Libraries

The CLR

Profiling& Debugging APIs

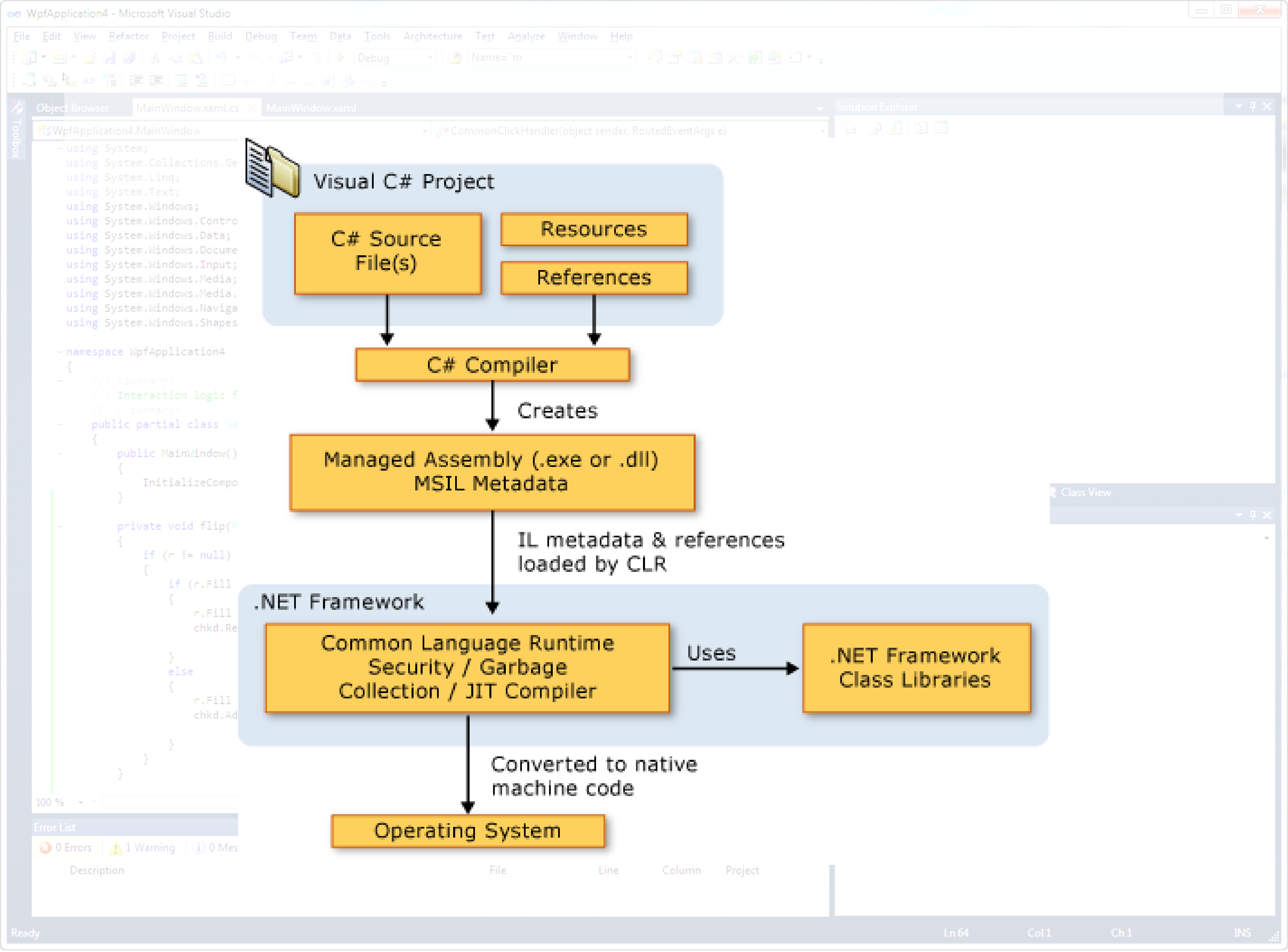
JIT & NGEN

Garbage Collector

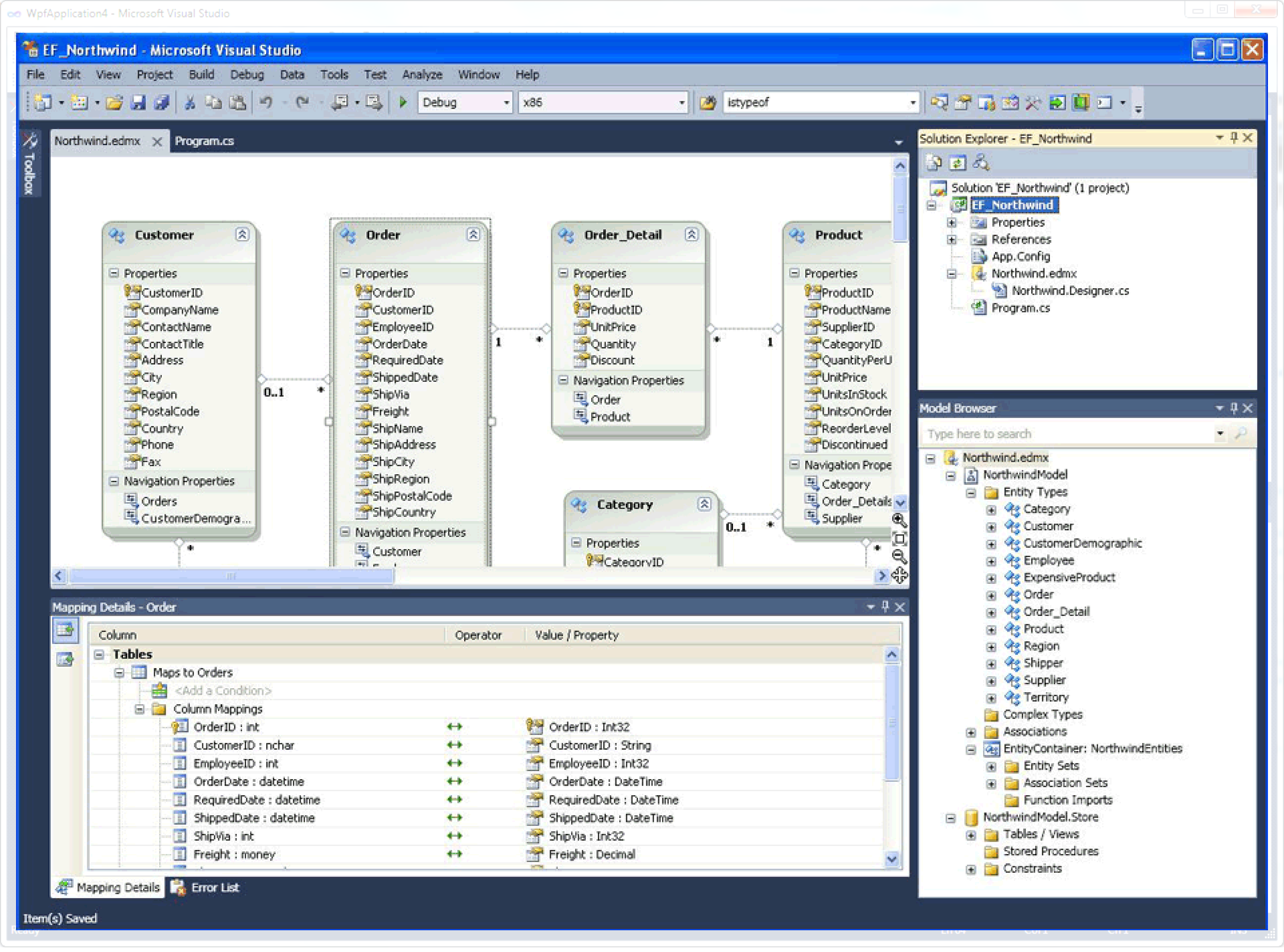
Security Model

Exception Handling

Loader & Binder


WpfApplication4 - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Team Data Tools Architecture Test Analyze Window Help
Debug
Name="m
Object Browser
MainWindow.xaml.cs
MainWindow.xaml
Solution Explorer
WpfApplication4.MainWindow
CommonClickHandler(object sender, RoutedEventArgs e)
Toolbox
Class View
100 %
Error List
0 Errors
1 Warning
0 Mes
Description
File
Line
Column
Project
Ready
Ln 64
Col 1
Ch 1
INS
Visual C# Project
C# Source File(s)
Resources
References
C# Compiler
Creates
Managed Assembly (.exe or .dll) MSIL Metadata
IL metadata & references loaded by CLR
.NET Framework
Common Language Runtime Security / Garbage Collection / JIT Compiler
Uses
.NET Framework Class Libraries
Converted to native machine code
Operating System

```
using System;
using System.Collections.Ge
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Contro
using System.Windows.Data;
using System.Windows.Docume
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.
using System.Windows.Naviga
using System.Windows.Shapes

namespace WpfApplication4
{
    /// <summary>
    /// Interaction logic f
    /// </summary>
    public partial class Ma
    {
        public MainWindow()
        {
            InitializeCompo
        }

        private void flip(P
        {
            if (r != null)
            {
                if (r.Fill
                {
                    r.Fill
                    chkd.Re
                }
                else
                {
                    r.Fill
                    chkd.Ad
                }
            }
        }
```



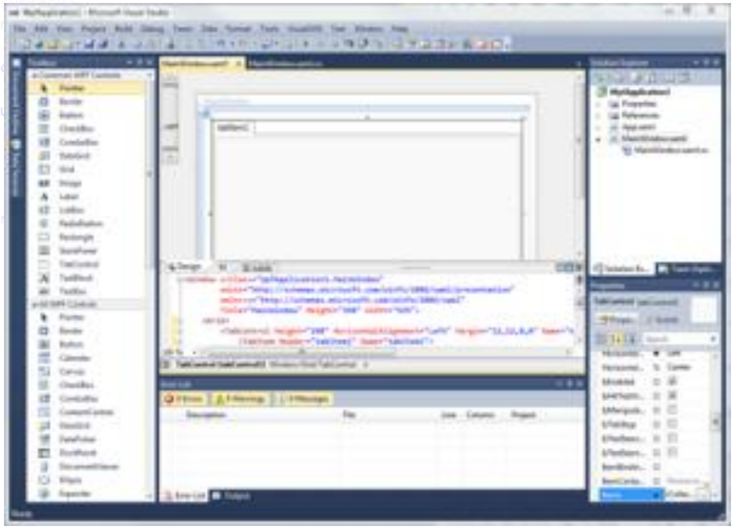
Column	Operator	Value / Property
Tables		
Maps to Orders		
<Add a Condition>		
Column Mappings		
OrderID : int	↔	OrderID : Int32
CustomerID : nchar	↔	CustomerID : String
EmployeeID : int	↔	EmployeeID : Int32
OrderDate : datetime	↔	OrderDate : DateTime
RequiredDate : datetime	↔	RequiredDate : DateTime
ShippedDate : datetime	↔	ShippedDate : DateTime
ShipVia : int	↔	ShipVia : Int32
Freight : money	↔	Freight : Decimal

Item(s) Saved

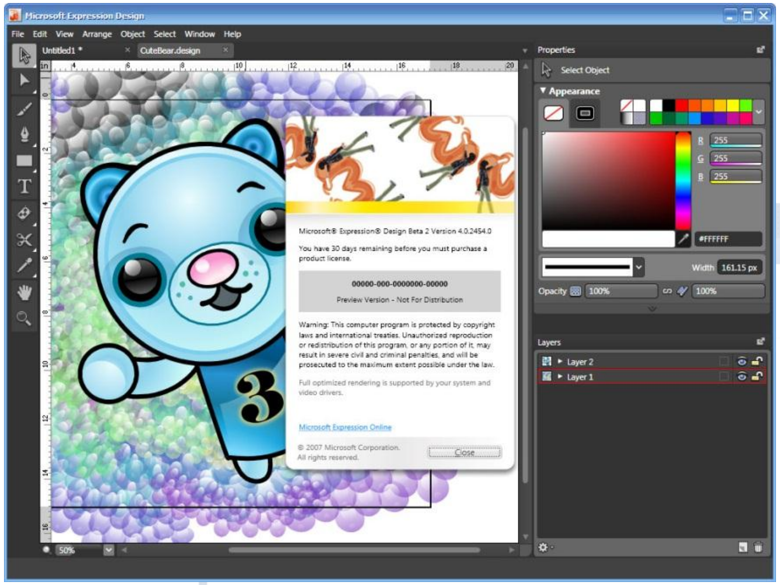
WpfApplication4 - Microsoft Visual Studio

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace WpfApplication4
{
```

Microsoft® Visual Studio®



Microsoft® Expression®



Microsoft® Expression® Design Beta 2 Version 4.0.2454.0

You have 30 days remaining before you must purchase a product license.

00000-000-0000000-00000

Preview Version - Not For Distribution

Warning: This computer program is protected by copyright laws and international treaties. Unauthorized reproduction or redistribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Full optimized rendering is supported by your system and video drivers.

Microsoft Expression Online

© 2007 Microsoft Corporation.
All rights reserved.

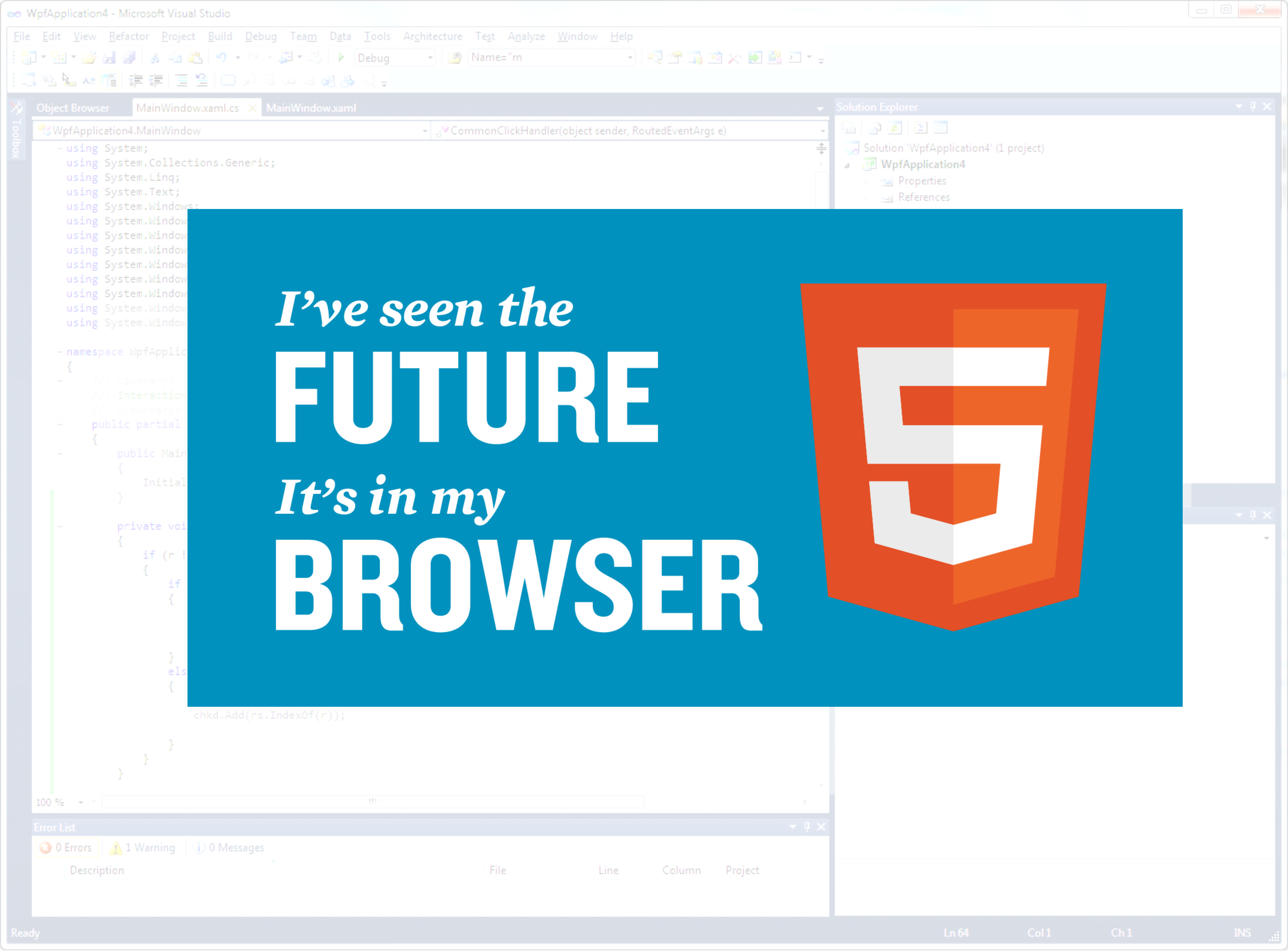
Close

Error List

0 Errors | 1 Warning | 0 Messages

Description File Line Column Project

Ready Ln 64 Col 1 Ch 1 INS



WpfApplication4 - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Team Data Tools Architecture Test Analyze Window Help
Debug
Name="m
Object Browser
MainWindow.xaml.cs
MainWindow.xaml
WpfApplication4.MainWindow
CommonClickHandler(object sender, RoutedEventArgs e)
Solution Explorer
Solution 'WpfApplication4' (1 project)
WpfApplication4
Properties
References
Toolbox
100 %
Error List
0 Errors
1 Warning
0 Messages
Description
File
Line
Column
Project
Ready
Ln 64
Col 1
Ch 1
INS

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Window
using System.Window
using System.Window
using System.Window
using System.Window
using System.Window
using System.Window
using System.Window

namespace WpfApplic
{
    /// <summary>
    /// Interaction
    /// </summary>
    public partial
    {
        public Main
        {
            Initial
        }

        private voi
        {
            if (c !
            {
                if
                {

                }
                els
                {
                    chkd.Add(rs.IndexOf(c));

                }
            }
        }
```

I've seen the

FUTURE

It's in my

BROWSER

```
using System;
using System.Collection
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.C
using System.Windows.D
using System.Windows.D
using System.Windows.I
using System.Windows.M
using System.Windows.M
using System.Windows.N
using System.Windows.S

namespace WpfApplicati
{
    /// <summary>
    /// Interaction lo
    /// </summary>
    public partial cla
    {
        public MainWin
        {
            Initialize
        }

        private void f
        {
            if (r != n
            {
                if (r.
                {
                    r.
                    ch

                }
                else
                {
                    r.
                    ch

                }
            }
        }
```

0 Errors | 1 Warning | 0 Messages

Description	File	Line	Column	Project

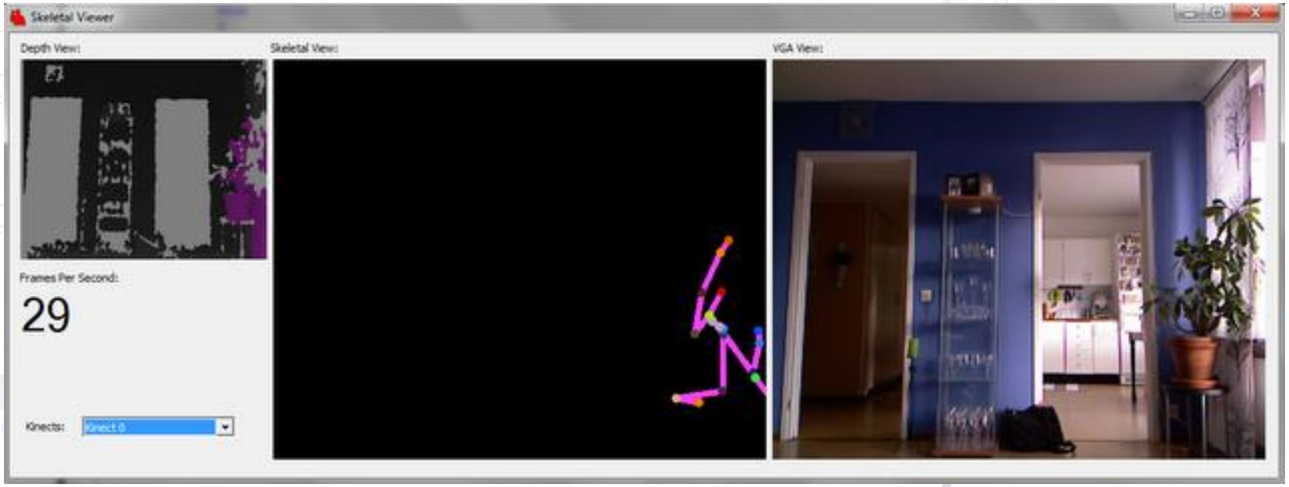
WpfApplication4 - Microsoft Visual Studio

File Edit View Refactor Project Build Debug Team Data Tools Architecture Test Analyze Window Help

Windows Phone 7 Series

KINECT for XBOX 360

```
private void flip(Polygon r)
{
    if (r != null)
    {
        if (r
        {

        }
        else
        {

        }
    }
}
```



Skeletal Viewer

Depth View:

Skeletal View:

VGA View:

Frames Per Second:

29

Kinects: Kinect 0

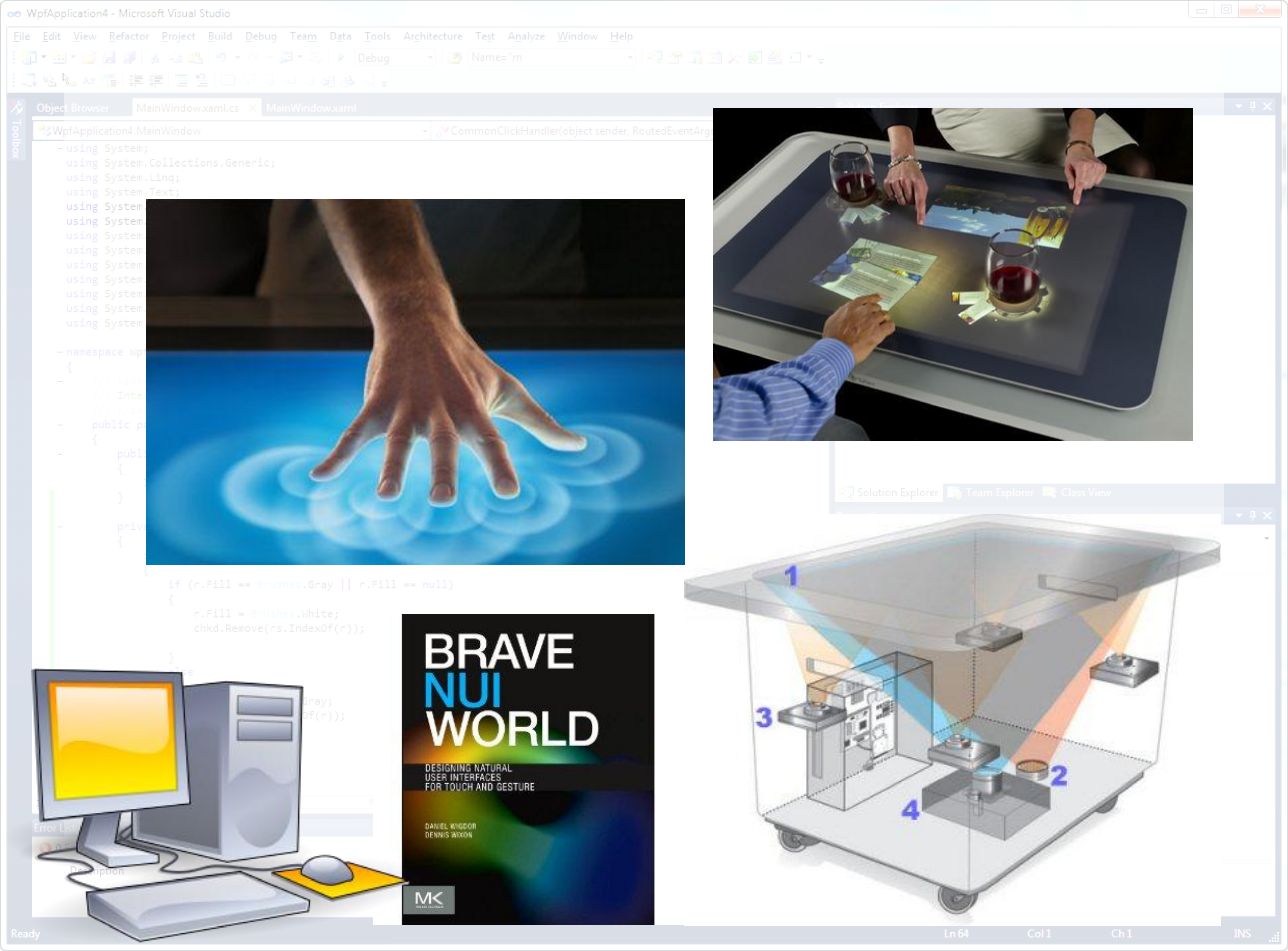
100 %

Error List

0 Errors 1 Warning

Description

Ready Ln 64 Col 1 Ch 1 INS



WpfApplication4 - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Team Data Tools Architecture Test Analyze Window Help
Debug
Name="m
Toolbox
Object Browser
MainWindow.xaml.cs
MainWindow.xaml
WpfApplication4.MainWindow
CommonClickHandler(object sender, RoutedEventArg
Solution Explorer
Team Explorer
Class View
Error List
Description
Ready
Ln 64
Col 1
Ch 1
INS
1
2
3
4
BRAVE
NUI
WORLD
DESIGNING NATURAL USER INTERFACES FOR TOUCH AND GESTURE
DANIEL WIGDOR
DENNIS WIXON
MK

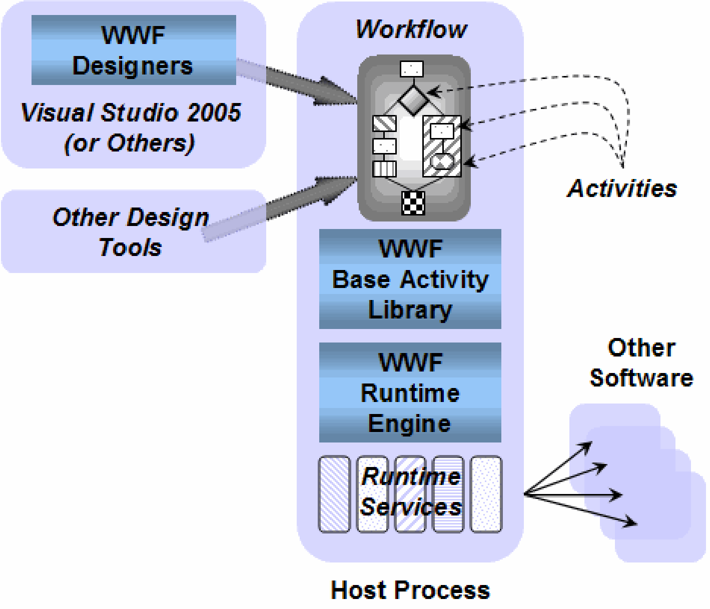
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.
using System.
using System.
using System.
using System.
using System.
using System.
using System.
using System.

namespace Wp
{
    /// sum
    /// Inte
    /// </su
    public pa
    {
        publ
        {
        }

        priv
        {

            if (r.Fill == Brushes.Gray || r.Fill == null)
            {
                r.Fill = Brushes.White;
                chkd.Remove(rs.IndexOf(r));
            }
            else
                Gray;
                Of(r));
```

Microsoft ASP.net

Microsoft Silverlight

Windows Azure AppFabric

Microsoft Sync Framework



WWF Designers

Visual Studio 2005 (or Others)

Other Design Tools

Workflow

Activities

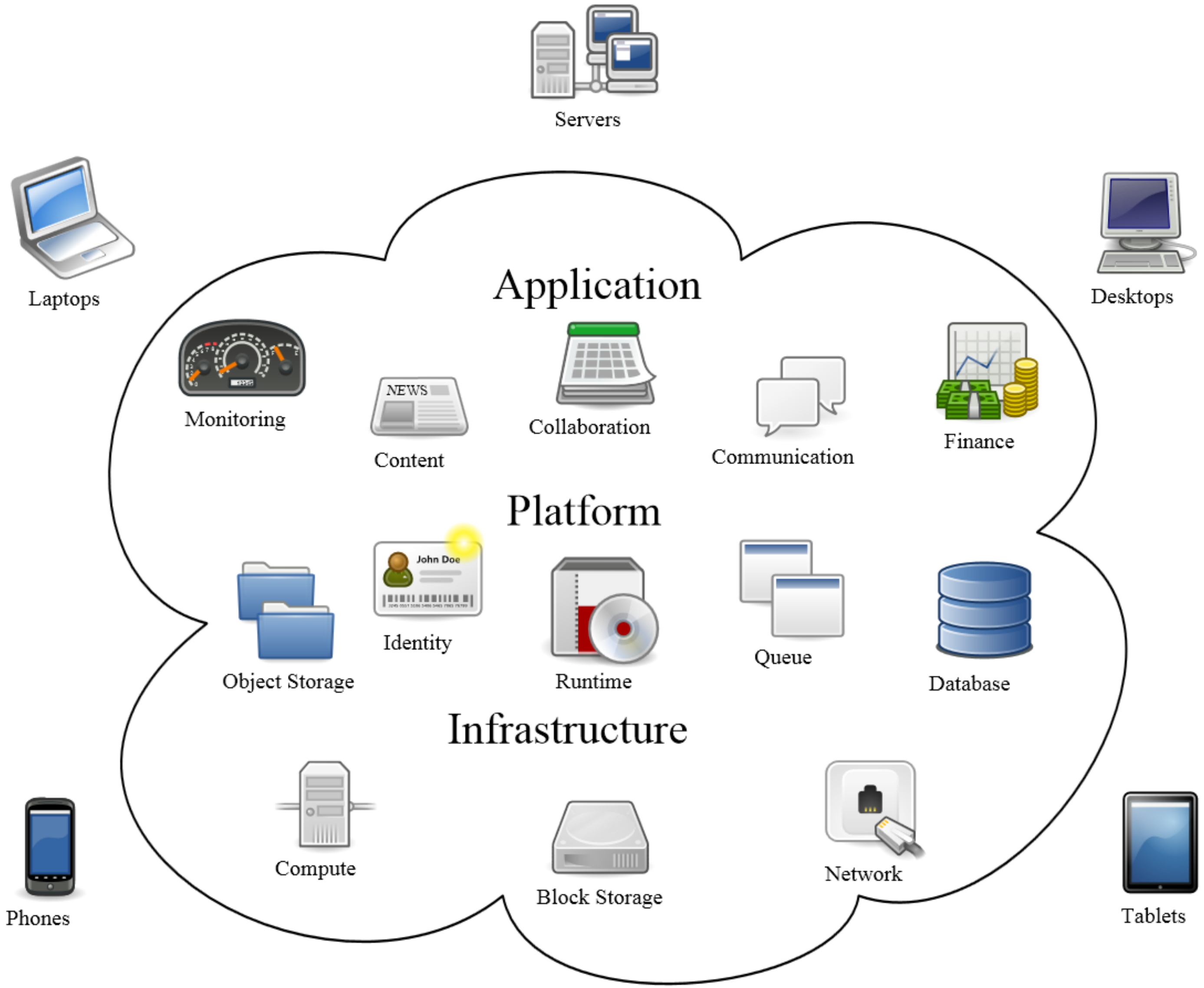
WWF Base Activity Library

WWF Runtime Engine

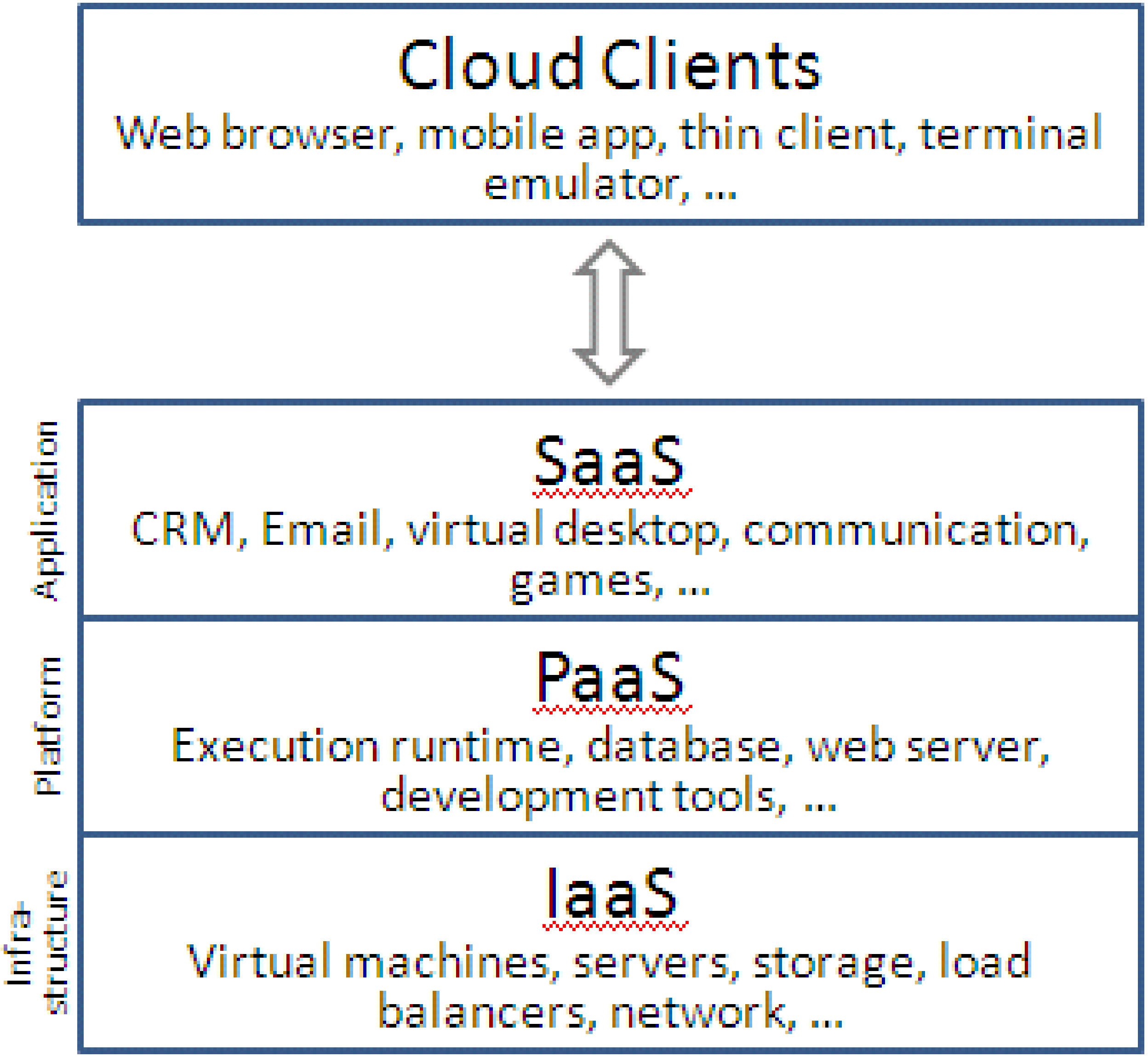
Runtime Services

Other Software

Host Process



Cloud Computing



Cloud Clients

Web browser, mobile app, thin client, terminal emulator, …

Application

SaaS

CRM, Email, virtual desktop, communication, games, …

Platform

PaaS

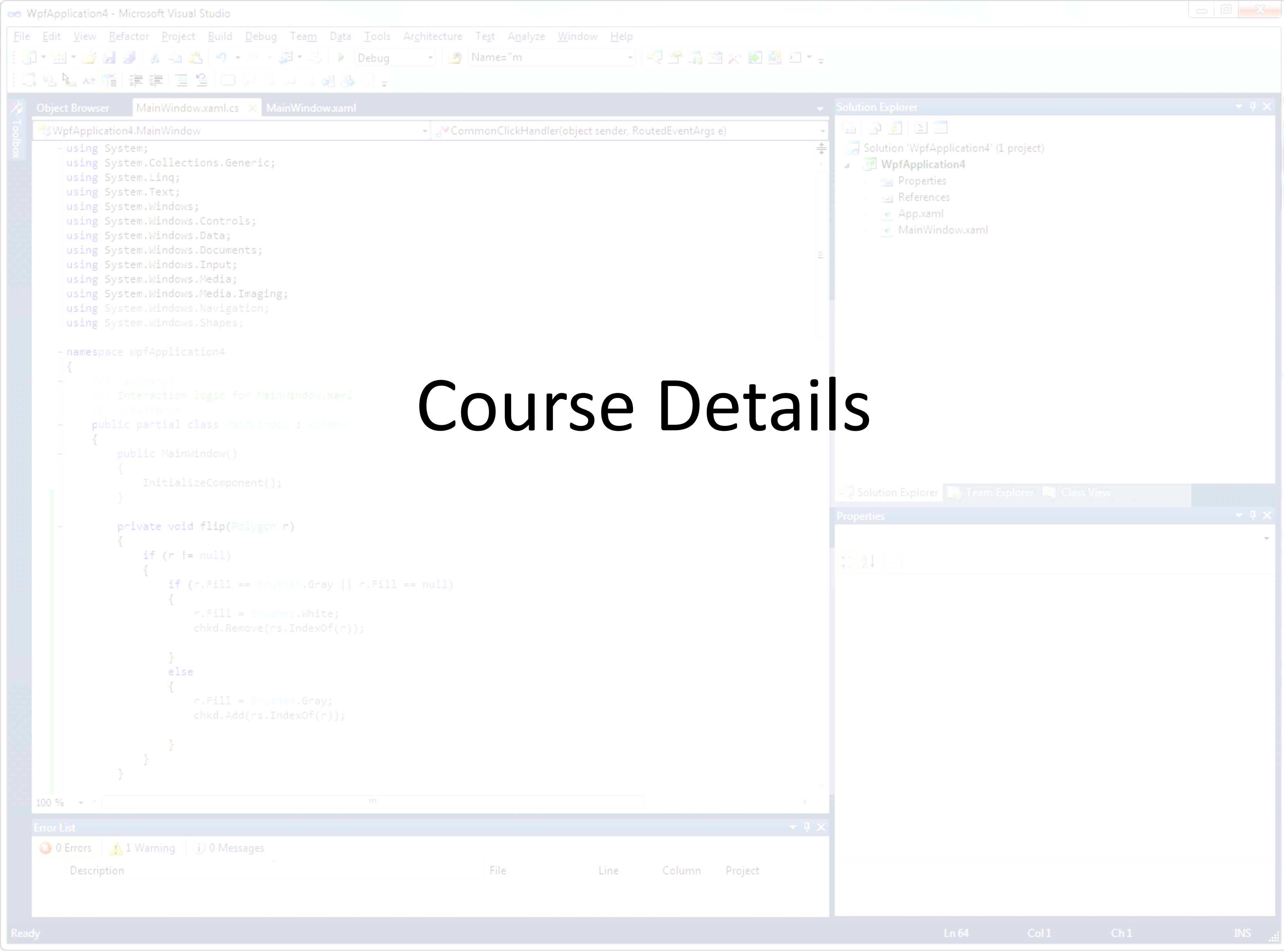
Execution runtime, database, web server, development tools, …

Infra-structure

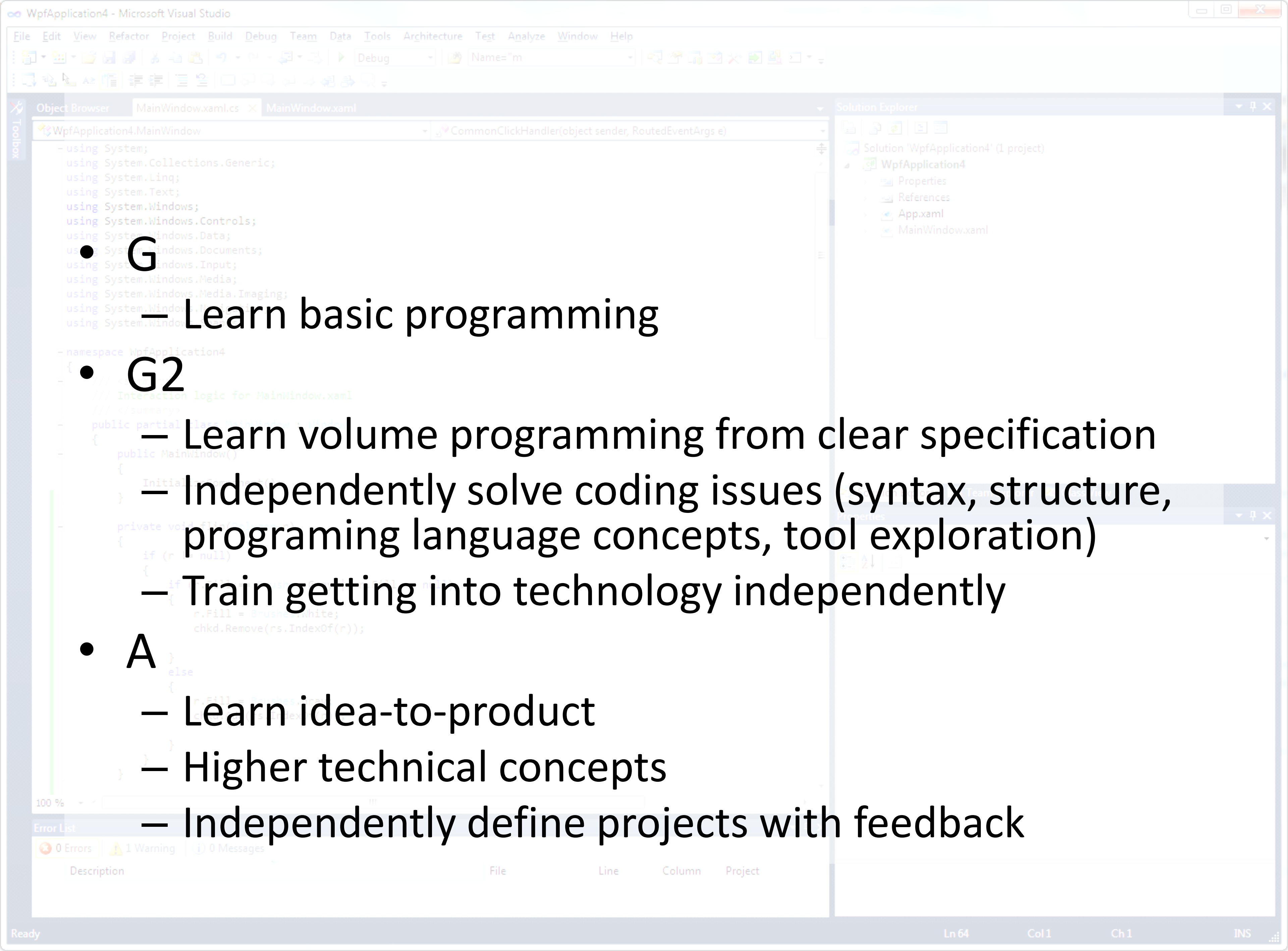
IaaS

Virtual machines, servers, storage, load balancers, network, …

Course Details

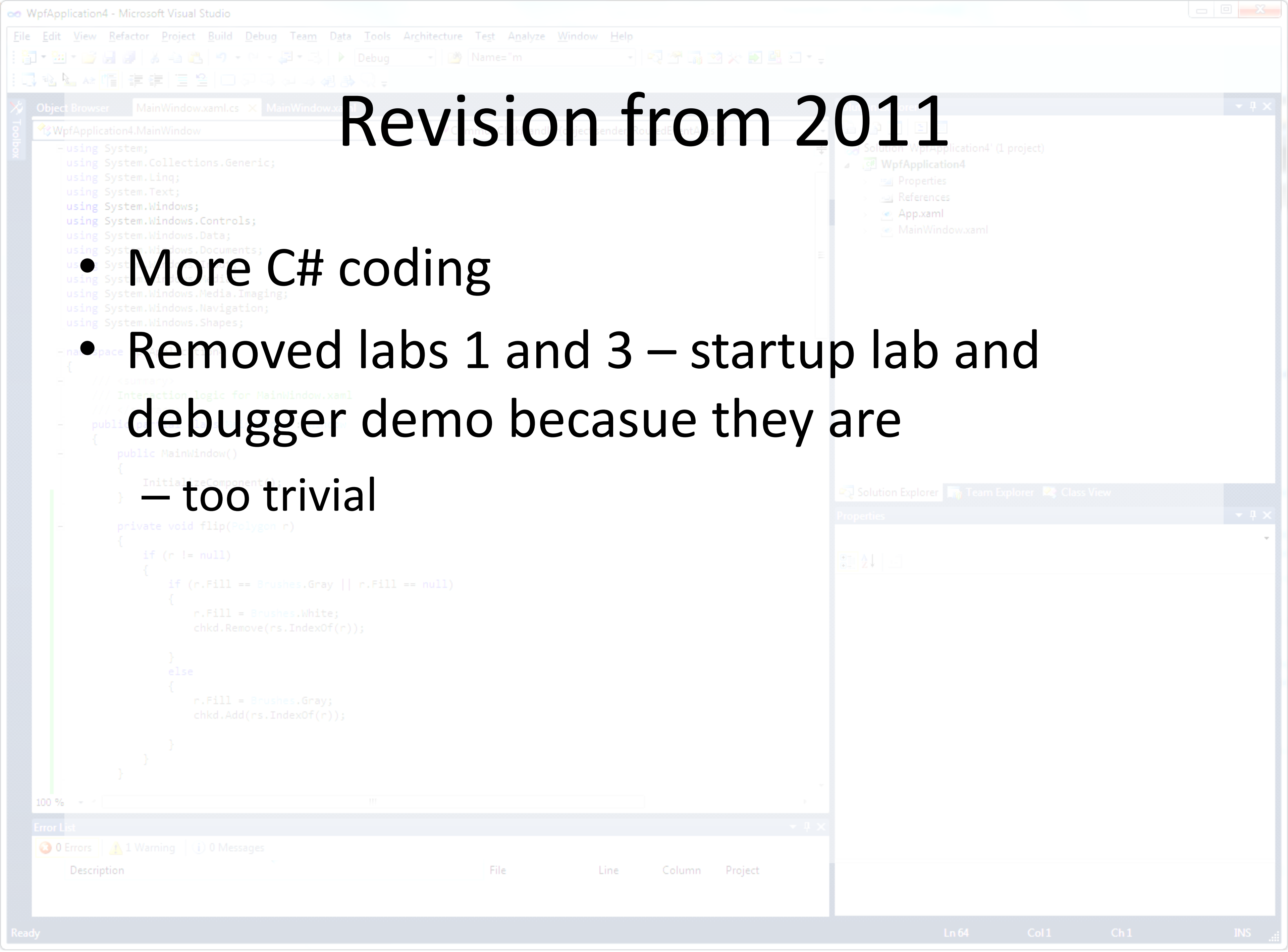


```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace WpfApplication4
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }

        private void flip(Polygon r)
        {
            if (r != null)
            {
                if (r.Fill == Brushes.Gray || r.Fill == null)
                {
                    r.Fill = Brushes.White;
                    chkd.Remove(rs.IndexOf(r));

                }
                else
                {
                    r.Fill = Brushes.Gray;
                    chkd.Add(rs.IndexOf(r));

                }
            }
        }
```


WpfApplication4 - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Team Data Tools Architecture Test Analyze Window Help
Debug
Name="m"
Object Browser
MainWindow.xaml.cs
MainWindow.xaml
WpfApplication4.MainWindow
CommonClickHandler(object sender, RoutedEventArgs e)
Solution Explorer
Solution 'WpfApplication4' (1 project)
WpfApplication4
Properties
References
App.xaml
MainWindow.xaml
Properties
Toolbox
100 %
Error List
0 Errors
1 Warning
0 Messages
Description
File
Line
Column
Project
Ready
Ln 64
Col 1
Ch 1
INS

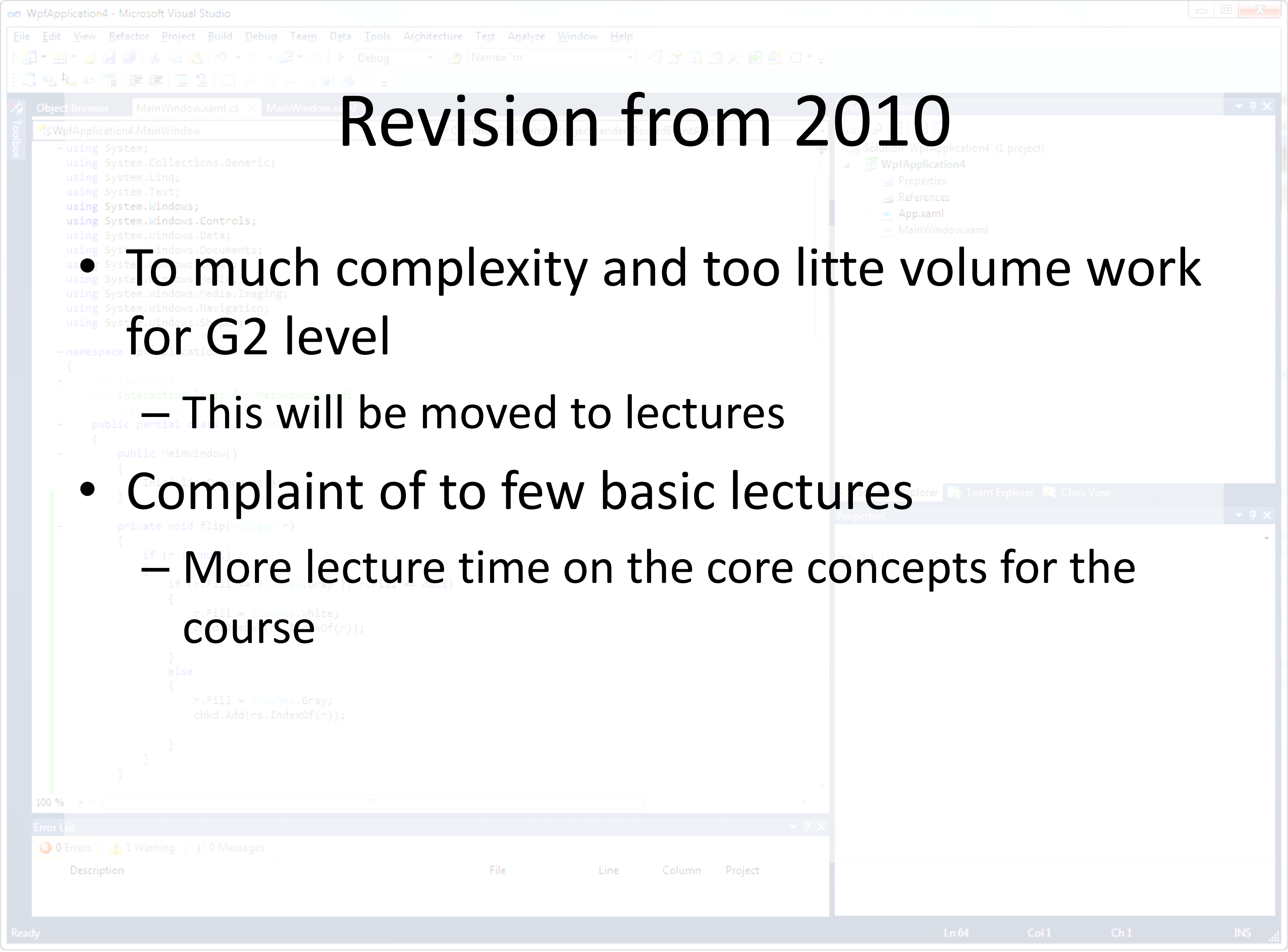
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
```

- G
  - Learn basic programming
- G2
  - Learn volume programming from clear specification
  - Independently solve coding issues (syntax, structure, programing language concepts, tool exploration)
  - Train getting into technology independently
- A
  - Learn idea-to-product
  - Higher technical concepts
  - Independently define projects with feedback



Revision from 2011

- More C# coding
- Removed labs 1 and 3 – startup lab and debugger demo becasue they are
  - too trivial

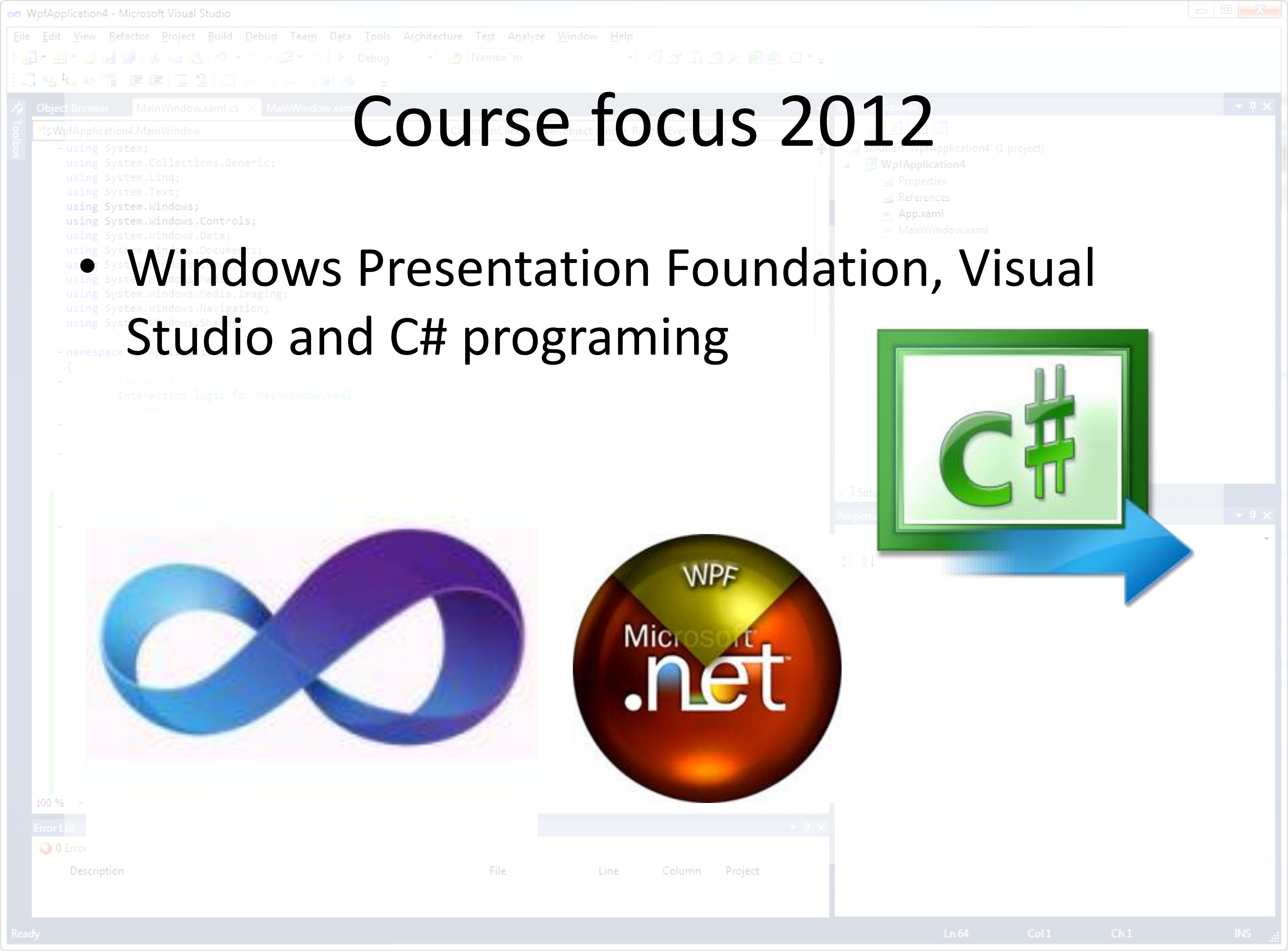


Revision from 2010

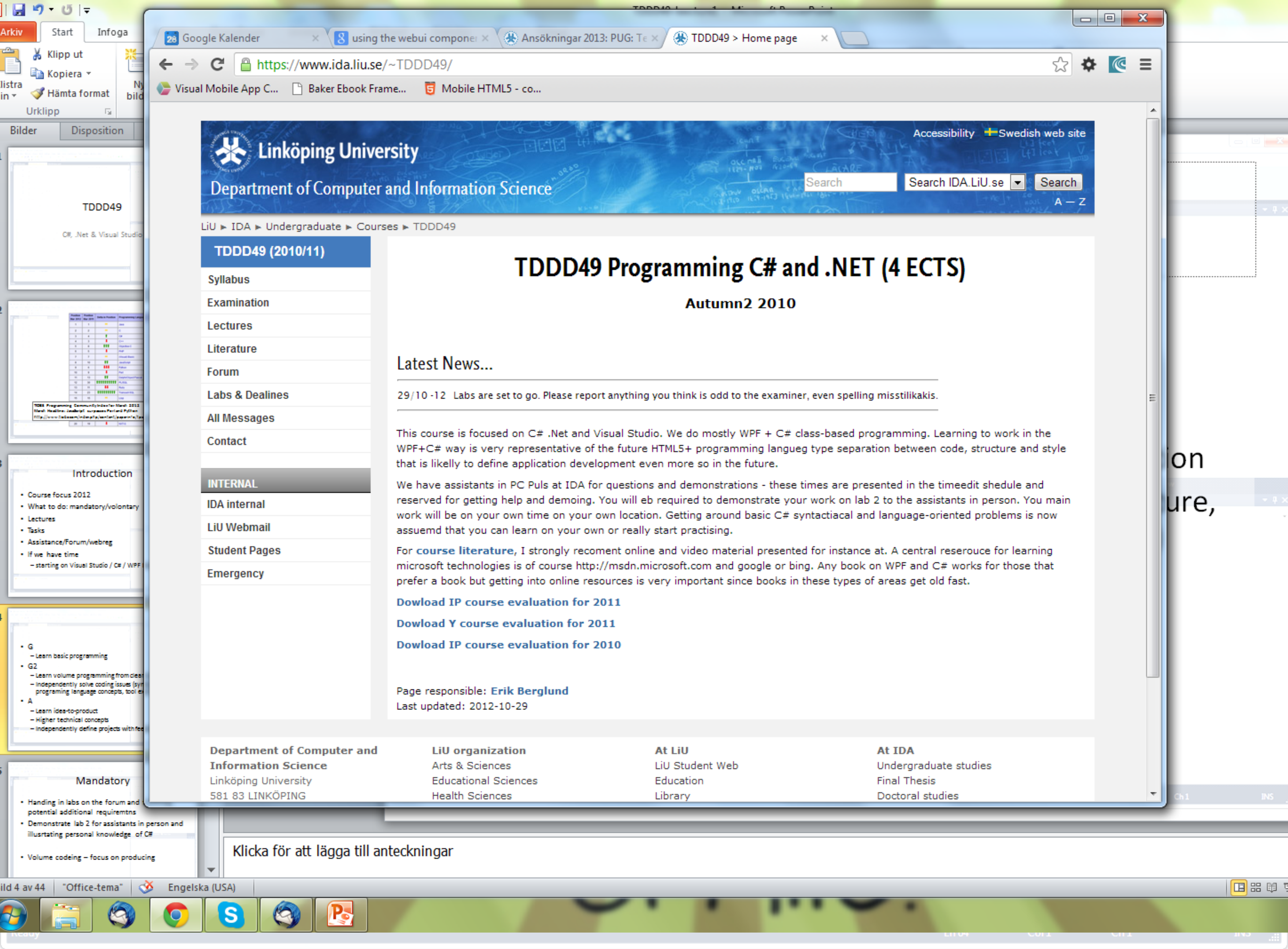
- To much complexity and too litte volume work for G2 level
  - This will be moved to lectures
- Complaint of to few basic lectures
  - More lecture time on the core concepts for the course

Course focus 2012

- Windows Presentation Foundation, Visual Studio and C# programing



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;

/// Interaction logic for MainWindow.xaml
```


Arkiv
Start
Infoga
Klipp ut
Kopiera
Hämta format
Urklipp
Bilder
Disposition
TDDD49
C#, .Net & Visual Studio
Introduction
Course focus 2012
What to do: mandatory/volontary
Lectures
Tasks
Assistance/Forum/webreg
If we have time
– starting on Visual Studio / C# / WPF
G
– Learn basic programming
G2
– Learn volume programming from clear
– Independently solve coding issues (sy
programing language concepts, tool e
A
– Learn idea-to-product
– Higher technical concepts
– Independently define projects with fe
Mandatory
Handing in labs on the forum and potential additional requiremtns
Demonstrate lab 2 for assistants in person and illusrtating personal knowledge of C#
Volume codeing – focus on producing
Google Kalender
using the webui compone
Ansökningar 2013: PUG: Te
TDDD49 > Home page
https://www.ida.liu.se/~TDDD49/
Visual Mobile App C...
Baker Ebook Frame...
Mobile HTML5 - co...
Linköping University
Department of Computer and Information Science
Accessibility
Swedish web site
Search
Search IDA.LiU.se
Search
A – Z
LiU ▸ IDA ▸ Undergraduate ▸ Courses ▸ TDDD49
TDDD49 (2010/11)
Syllabus
Examination
Lectures
Literature
Forum
Labs & Dealines
All Messages
Contact
INTERNAL
IDA internal
LiU Webmail
Student Pages
Emergency
ion
ure,
Klicka för att lägga till anteckningar
ild 4 av 44
"Office-tema"
Engelska (USA)

TDDD49 Programming C# and .NET (4 ECTS)

Autumn2 2010

Latest News...

29/10 -12 Labs are set to go. Please report anything you think is odd to the examiner, even spelling misstilikakis.

This course is focused on C# .Net and Visual Studio. We do mostly WPF + C# class-based programming. Learning to work in the WPF+C# way is very representative of the future HTML5+ programming langueg type separation between code, structure and style that is likely to define application development even more so in the future.

We have assistants in PC Puls at IDA for questions and demonstrations - these times are presented in the timeedit shedule and reserved for getting help and demoing. You will eb required to demonstrate your work on lab 2 to the assistants in person. You main work will be on your own time on your own location. Getting around basic C# syntactiacal and language-oriented problems is now assuemd that you can learn on your own or really start practising.

For **course literature**, I strongly recoment online and video material presented for instance at. A central reserouce for learning microsoft technologies is of course http://msdn.microsoft.com and google or bing. Any book on WPF and C# works for those that prefer a book but getting into online resources is very important since books in these types of areas get old fast.

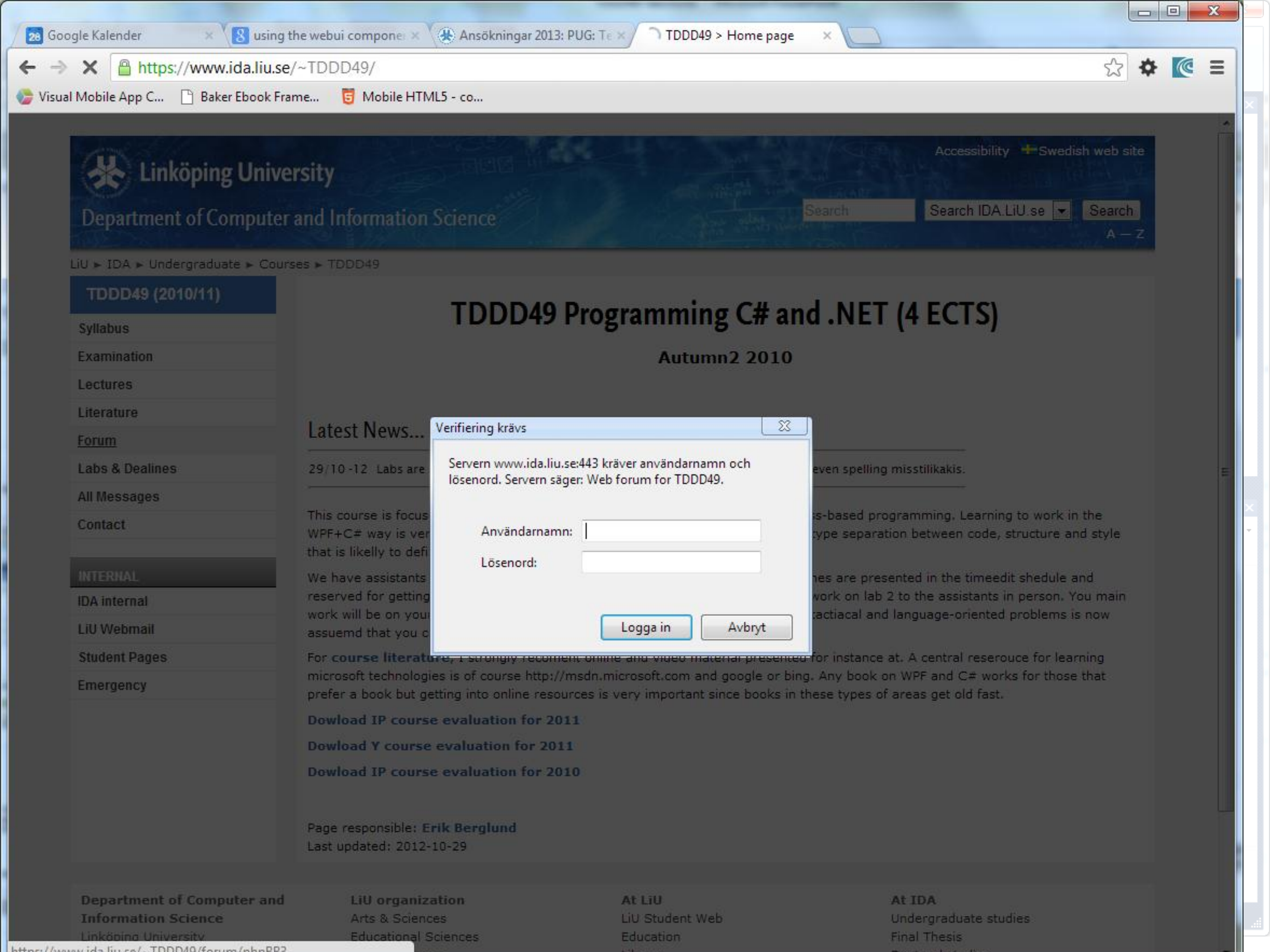
Dowload IP course evaluation for 2011

Dowload Y course evaluation for 2011

Dowload IP course evaluation for 2010

Page responsible: **Erik Berglund**
Last updated: 2012-10-29

Department of Computer and Information Science	LiU organization	At LiU	At IDA
Linköping University	Arts & Sciences	LiU Student Web	Undergraduate studies
581 83 LINKÖPING	Educational Sciences	Education	Final Thesis
	Health Sciences	Library	Doctoral studies



LiU ▸ IDA ▸ Undergraduate ▸ Courses ▸ TDDD49

TDDD49 Programming C# and .NET (4 ECTS)

Autumn2 2010

Latest News...

29/10 -12 Labs are … even spelling misstilikakis.

This course is focus… s-based programming. Learning to work in the WPF+C# way is ver… type separation between code, structure and style that is likely to defi…

We have assistants … es are presented in the timeedit shedule and reserved for getting … work on lab 2 to the assistants in person. You main work will be on you… actiacal and language-oriented problems is now assuemd that you c…

For **course literature**, I strongly recoment online and video material presented for instance at. A central reserouce for learning microsoft technologies is of course http://msdn.microsoft.com and google or bing. Any book on WPF and C# works for those that prefer a book but getting into online resources is very important since books in these types of areas get old fast.

Dowload IP course evaluation for 2011

Dowload Y course evaluation for 2011

Dowload IP course evaluation for 2010

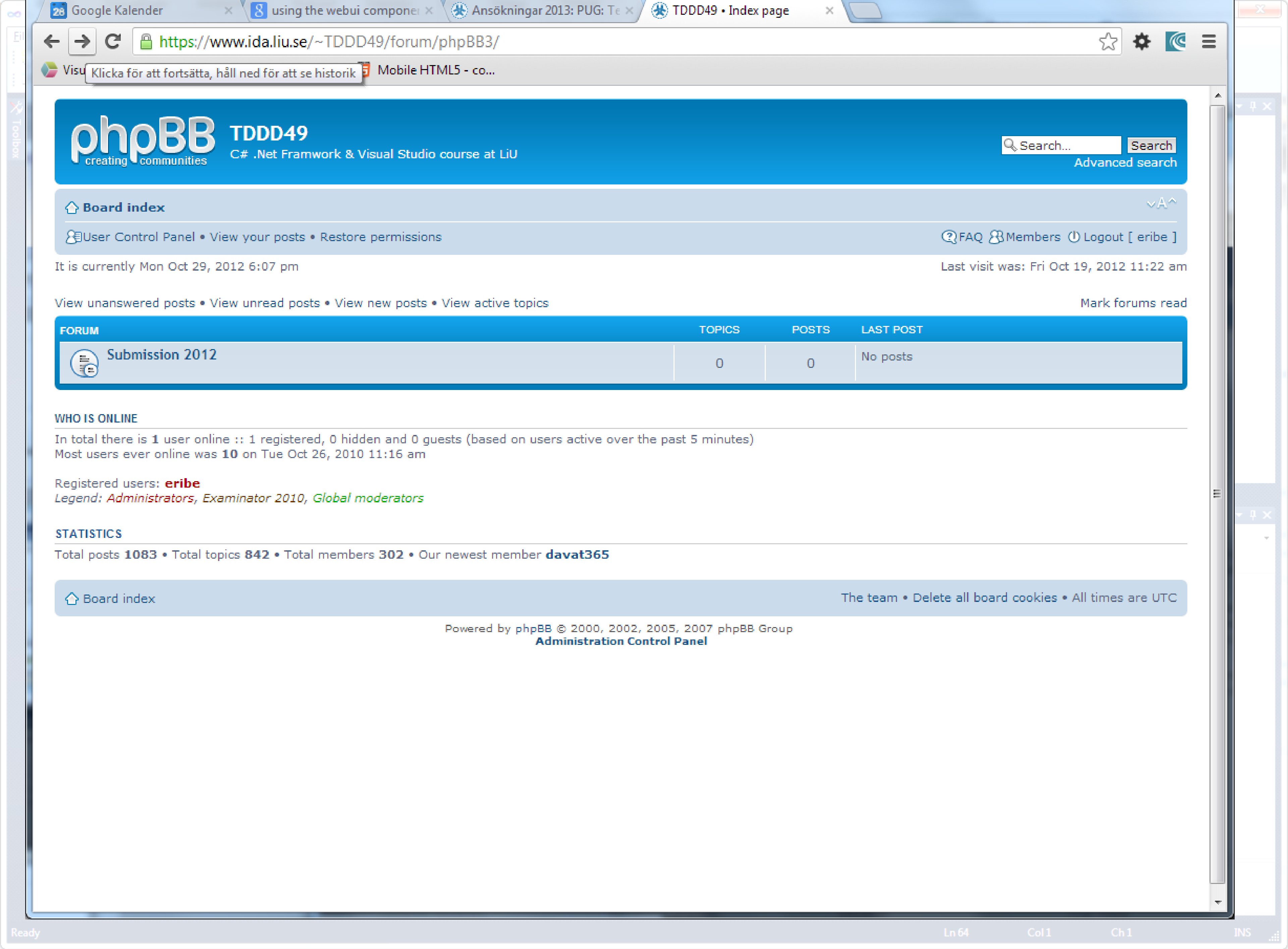
Page responsible: **Erik Berglund**
Last updated: 2012-10-29

Verifiering krävs

Servern www.ida.liu.se:443 kräver användarnamn och lösenord. Servern säger: Web forum for TDDD49.

Användarnamn:

Lösenord:



Google Kalender
using the webui componer
Ansökningar 2013: PUG: Te
TDDD49 • Index page
https://www.ida.liu.se/~TDDD49/forum/phpBB3/
Visu
Klicka för att fortsätta, håll ned för att se historik
Mobile HTML5 - co...
phpBB creating communities

TDDD49

C# .Net Framwork & Visual Studio course at LiU

Search... Search

Advanced search

Board index

User Control Panel • View your posts • Restore permissions

FAQ Members Logout [eribe]

It is currently Mon Oct 29, 2012 6:07 pm

Last visit was: Fri Oct 19, 2012 11:22 am

View unanswered posts • View unread posts • View new posts • View active topics

Mark forums read

FORUM	TOPICS	POSTS	LAST POST
Submission 2012	0	0	No posts

WHO IS ONLINE

In total there is **1** user online :: 1 registered, 0 hidden and 0 guests (based on users active over the past 5 minutes)
Most users ever online was **10** on Tue Oct 26, 2010 11:16 am

Registered users: **eribe**
Legend: Administrators, Examinator 2010, Global moderators

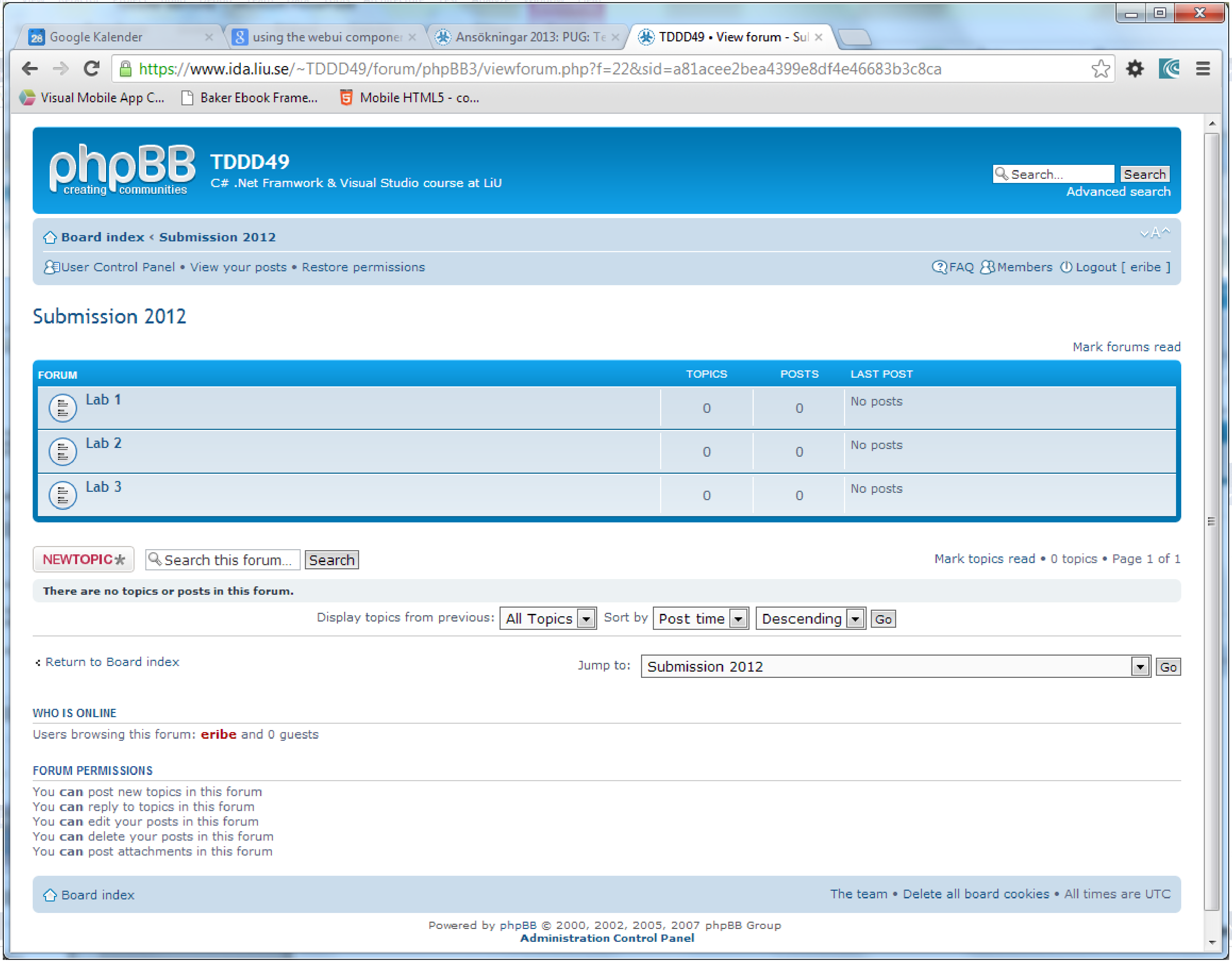
STATISTICS

Total posts **1083** • Total topics **842** • Total members **302** • Our newest member **davat365**

Board index

The team • Delete all board cookies • All times are UTC

Powered by phpBB © 2000, 2002, 2005, 2007 phpBB Group
Administration Control Panel



Google Kalender
using the webui compone
Ansökningar 2013: PUG: Te
TDDD49 • View forum - Sul
https://www.ida.liu.se/~TDDD49/forum/phpBB3/viewforum.php?f=22&sid=a81acee2bea4399e8df4e46683b3c8ca
Visual Mobile App C...
Baker Ebook Frame...
Mobile HTML5 - co...
phpBB creating communities

TDDD49

C# .Net Framwork & Visual Studio course at LiU

Search... Search

Advanced search

Board index ‹ Submission 2012

User Control Panel • View your posts • Restore permissions

FAQ Members Logout [eribe]

Submission 2012

Mark forums read

FORUM	TOPICS	POSTS	LAST POST
Lab 1	0	0	No posts
Lab 2	0	0	No posts
Lab 3	0	0	No posts

NEWTOPIC✱ Search this forum... Search

Mark topics read • 0 topics • Page 1 of 1

There are no topics or posts in this forum.

Display topics from previous: All Topics Sort by Post time Descending Go

‹ Return to Board index

Jump to: Submission 2012 Go

WHO IS ONLINE

Users browsing this forum: **eribe** and 0 guests

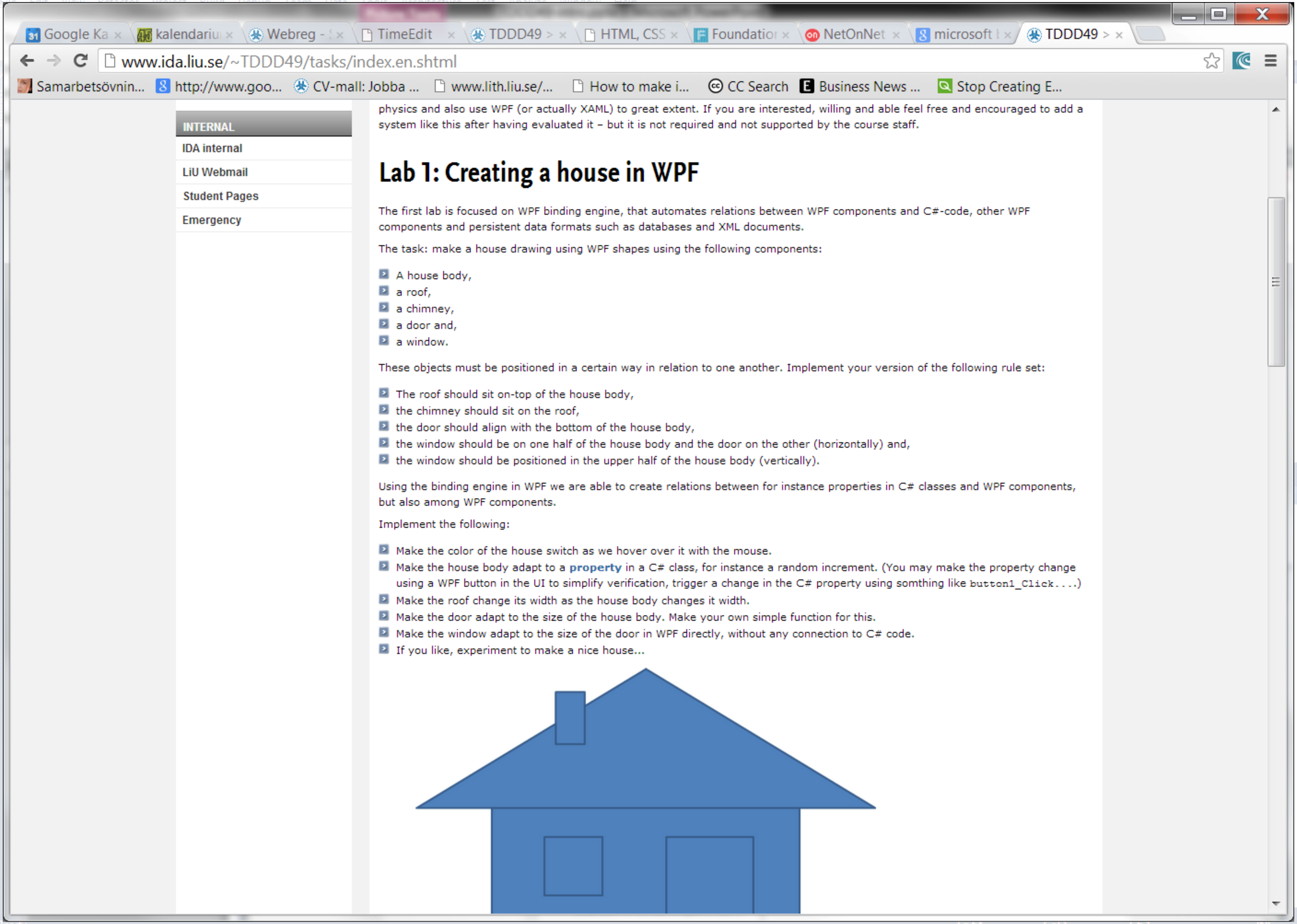
FORUM PERMISSIONS

You **can** post new topics in this forum
You **can** reply to topics in this forum
You **can** edit your posts in this forum
You **can** delete your posts in this forum
You **can** post attachments in this forum

Board index

The team • Delete all board cookies • All times are UTC

Powered by phpBB © 2000, 2002, 2005, 2007 phpBB Group
Administration Control Panel



physics and also use WPF (or actually XAML) to great extent. If you are interested, willing and able feel free and encouraged to add a system like this after having evaluated it – but it is not required and not supported by the course staff.

Lab 1: Creating a house in WPF

The first lab is focused on WPF binding engine, that automates relations between WPF components and C#-code, other WPF components and persistent data formats such as databases and XML documents.

The task: make a house drawing using WPF shapes using the following components:

- A house body,
- a roof,
- a chimney,
- a door and,
- a window.

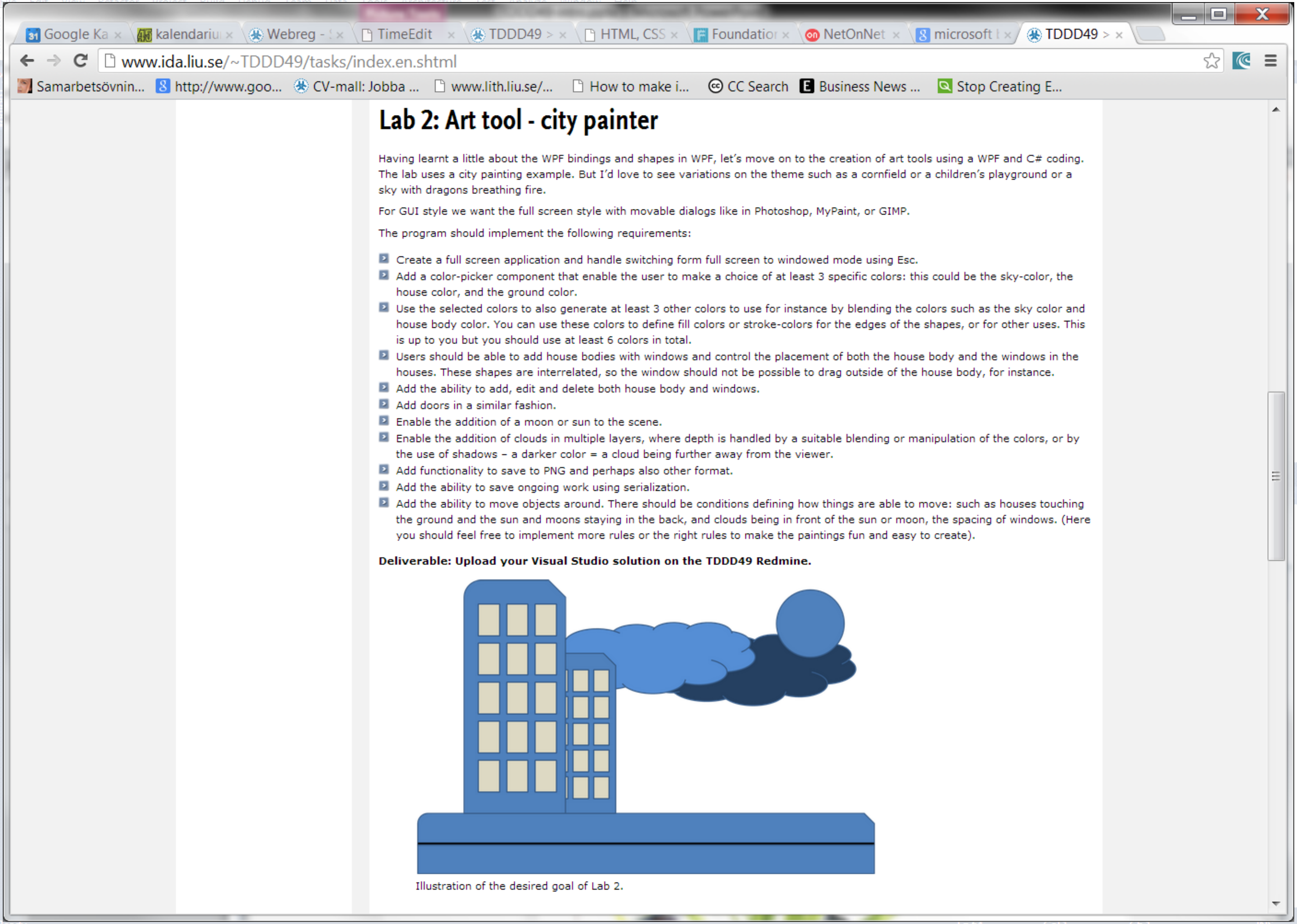
These objects must be positioned in a certain way in relation to one another. Implement your version of the following rule set:

- The roof should sit on-top of the house body,
- the chimney should sit on the roof,
- the door should align with the bottom of the house body,
- the window should be on one half of the house body and the door on the other (horizontally) and,
- the window should be positioned in the upper half of the house body (vertically).

Using the binding engine in WPF we are able to create relations between for instance properties in C# classes and WPF components, but also among WPF components.

Implement the following:

- Make the color of the house switch as we hover over it with the mouse.
- Make the house body adapt to a **property** in a C# class, for instance a random increment. (You may make the property change using a WPF button in the UI to simplify verification, trigger a change in the C# property using somthing like `button1_Click...`.)
- Make the roof change its width as the house body changes it width.
- Make the door adapt to the size of the house body. Make your own simple function for this.
- Make the window adapt to the size of the door in WPF directly, without any connection to C# code.
- If you like, experiment to make a nice house...



Lab 2: Art tool - city painter

Having learnt a little about the WPF bindings and shapes in WPF, let's move on to the creation of art tools using a WPF and C# coding. The lab uses a city painting example. But I'd love to see variations on the theme such as a cornfield or a children's playground or a sky with dragons breathing fire.

For GUI style we want the full screen style with movable dialogs like in Photoshop, MyPaint, or GIMP.

The program should implement the following requirements:

- Create a full screen application and handle switching form full screen to windowed mode using Esc.
- Add a color-picker component that enable the user to make a choice of at least 3 specific colors: this could be the sky-color, the house color, and the ground color.
- Use the selected colors to also generate at least 3 other colors to use for instance by blending the colors such as the sky color and house body color. You can use these colors to define fill colors or stroke-colors for the edges of the shapes, or for other uses. This is up to you but you should use at least 6 colors in total.
- Users should be able to add house bodies with windows and control the placement of both the house body and the windows in the houses. These shapes are interrelated, so the window should not be possible to drag outside of the house body, for instance.
- Add the ability to add, edit and delete both house body and windows.
- Add doors in a similar fashion.
- Enable the addition of a moon or sun to the scene.
- Enable the addition of clouds in multiple layers, where depth is handled by a suitable blending or manipulation of the colors, or by the use of shadows – a darker color = a cloud being further away from the viewer.
- Add functionality to save to PNG and perhaps also other format.
- Add the ability to save ongoing work using serialization.
- Add the ability to move objects around. There should be conditions defining how things are able to move: such as houses touching the ground and the sun and moons staying in the back, and clouds being in front of the sun or moon, the spacing of windows. (Here you should feel free to implement more rules or the right rules to make the paintings fun and easy to create).

Deliverable: Upload your Visual Studio solution on the TDDD49 Redmine.

Illustration of the desired goal of Lab 2.

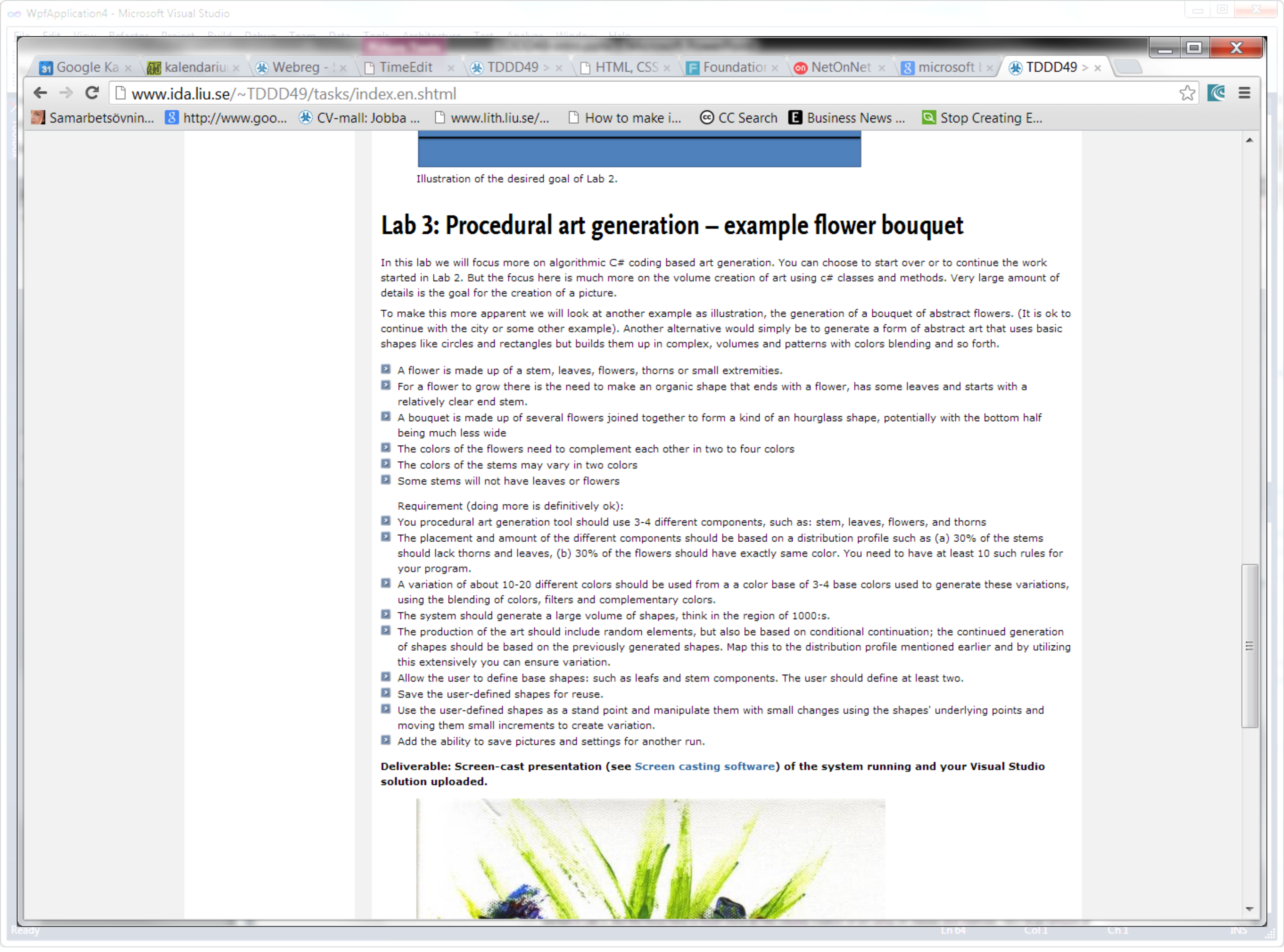


Illustration of the desired goal of Lab 2.

Lab 3: Procedural art generation – example flower bouquet

In this lab we will focus more on algorithmic C# coding based art generation. You can choose to start over or to continue the work started in Lab 2. But the focus here is much more on the volume creation of art using c# classes and methods. Very large amount of details is the goal for the creation of a picture.

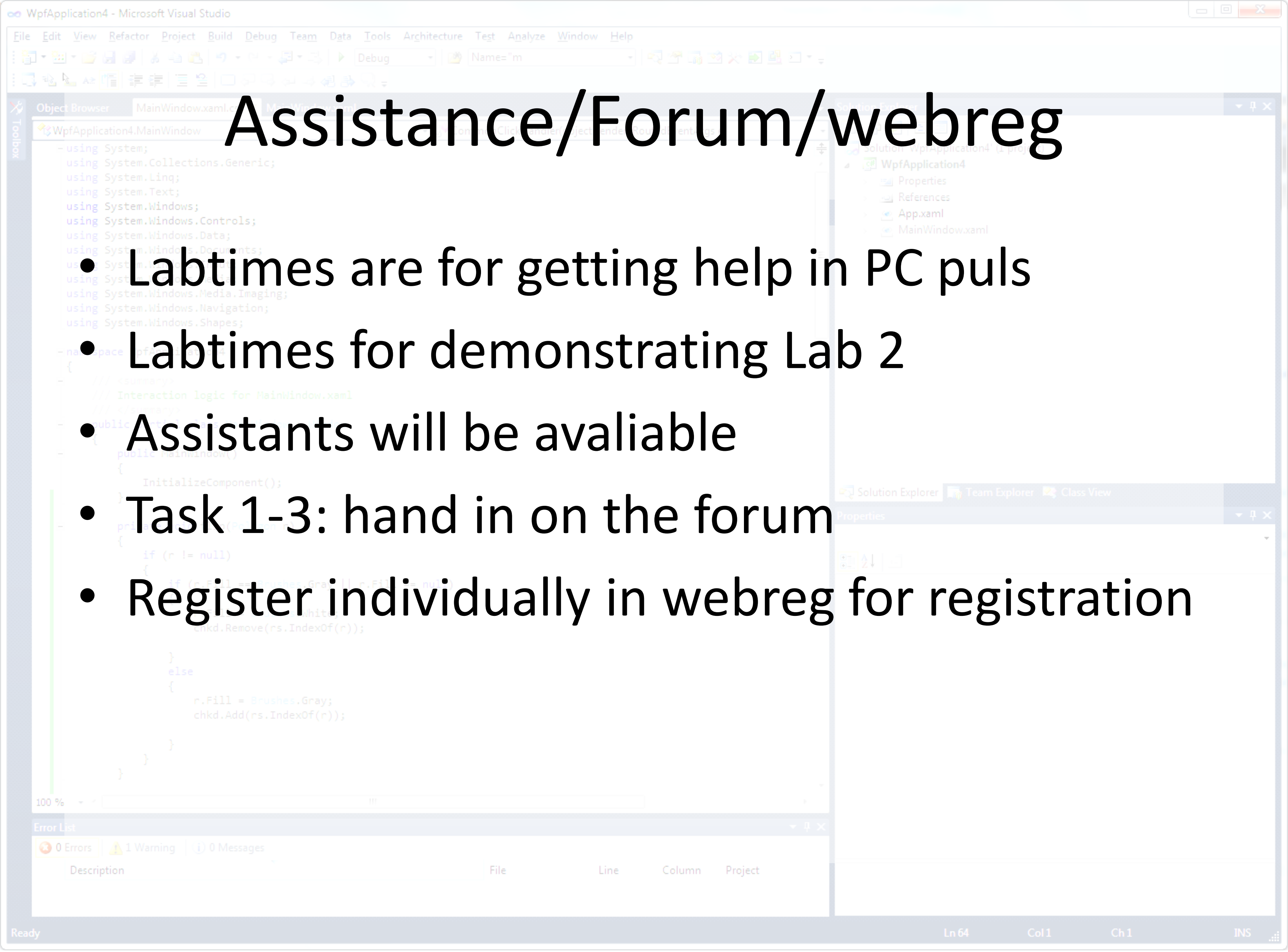
To make this more apparent we will look at another example as illustration, the generation of a bouquet of abstract flowers. (It is ok to continue with the city or some other example). Another alternative would simply be to generate a form of abstract art that uses basic shapes like circles and rectangles but builds them up in complex, volumes and patterns with colors blending and so forth.

- A flower is made up of a stem, leaves, flowers, thorns or small extremities.
- For a flower to grow there is the need to make an organic shape that ends with a flower, has some leaves and starts with a relatively clear end stem.
- A bouquet is made up of several flowers joined together to form a kind of an hourglass shape, potentially with the bottom half being much less wide
- The colors of the flowers need to complement each other in two to four colors
- The colors of the stems may vary in two colors
- Some stems will not have leaves or flowers

Requirement (doing more is definitively ok):

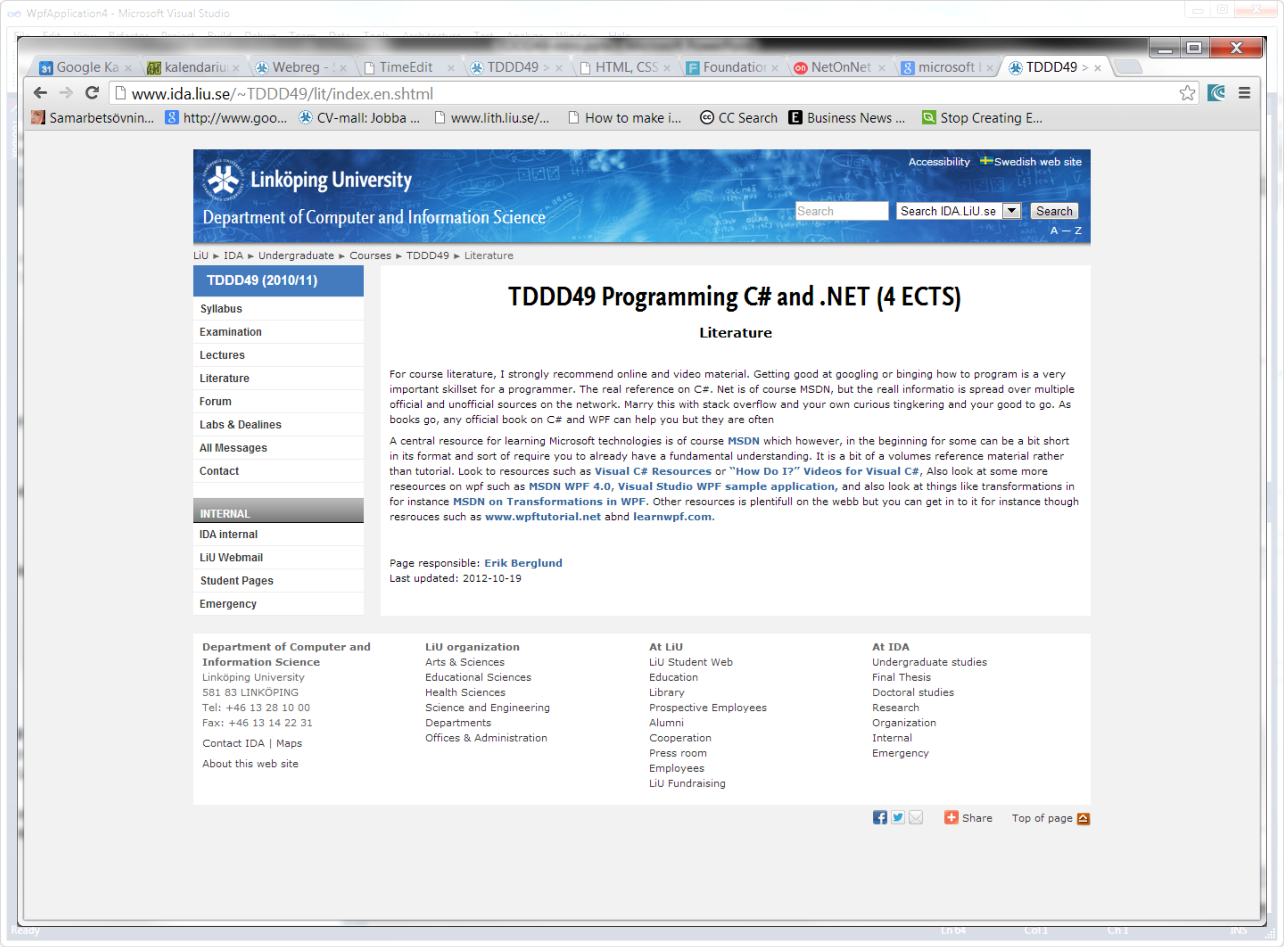
- You procedural art generation tool should use 3-4 different components, such as: stem, leaves, flowers, and thorns
- The placement and amount of the different components should be based on a distribution profile such as (a) 30% of the stems should lack thorns and leaves, (b) 30% of the flowers should have exactly same color. You need to have at least 10 such rules for your program.
- A variation of about 10-20 different colors should be used from a a color base of 3-4 base colors used to generate these variations, using the blending of colors, filters and complementary colors.
- The system should generate a large volume of shapes, think in the region of 1000:s.
- The production of the art should include random elements, but also be based on conditional continuation; the continued generation of shapes should be based on the previously generated shapes. Map this to the distribution profile mentioned earlier and by utilizing this extensively you can ensure variation.
- Allow the user to define base shapes: such as leafs and stem components. The user should define at least two.
- Save the user-defined shapes for reuse.
- Use the user-defined shapes as a stand point and manipulate them with small changes using the shapes' underlying points and moving them small increments to create variation.
- Add the ability to save pictures and settings for another run.

Deliverable: Screen-cast presentation (see Screen casting software) of the system running and your Visual Studio solution uploaded.



Assistance/Forum/webreg

- Labtimes are for getting help in PC puls
- Labtimes for demonstrating Lab 2
- Assistants will be avaliable
- Task 1-3: hand in on the forum
- Register individually in webreg for registration



Linköping University
Department of Computer and Information Science

LiU ▸ IDA ▸ Undergraduate ▸ Courses ▸ TDDD49 ▸ Literature

TDDD49 (2010/11)
- Syllabus
- Examination
- Lectures
- Literature
- Forum
- Labs & Dealines
- All Messages
- Contact

INTERNAL
- IDA internal
- LiU Webmail
- Student Pages
- Emergency

TDDD49 Programming C# and .NET (4 ECTS)

Literature

For course literature, I strongly recommend online and video material. Getting good at googling or binging how to program is a very important skillset for a programmer. The real reference on C#. Net is of course MSDN, but the reall informatio is spread over multiple official and unofficial sources on the network. Marry this with stack overflow and your own curious tingkering and your good to go. As books go, any official book on C# and WPF can help you but they are often

A central resource for learning Microsoft technologies is of course **MSDN** which however, in the beginning for some can be a bit short in its format and sort of require you to already have a fundamental understanding. It is a bit of a volumes reference material rather than tutorial. Look to resources such as **Visual C# Resources** or **"How Do I?" Videos for Visual C#**, Also look at some more reseources on wpf such as **MSDN WPF 4.0**, **Visual Studio WPF sample application**, and also look at things like transformations in for instance **MSDN on Transformations in WPF**. Other resources is plentifull on the webb but you can get in to it for instance though resrouces such as **www.wpftutorial.net** abnd **learnwpf.com**.

Page responsible: **Erik Berglund**
Last updated: 2012-10-19

Department of Computer and Information Science
Linköping University
581 83 LINKÖPING
Tel: +46 13 28 10 00
Fax: +46 13 14 22 31

Contact IDA | Maps

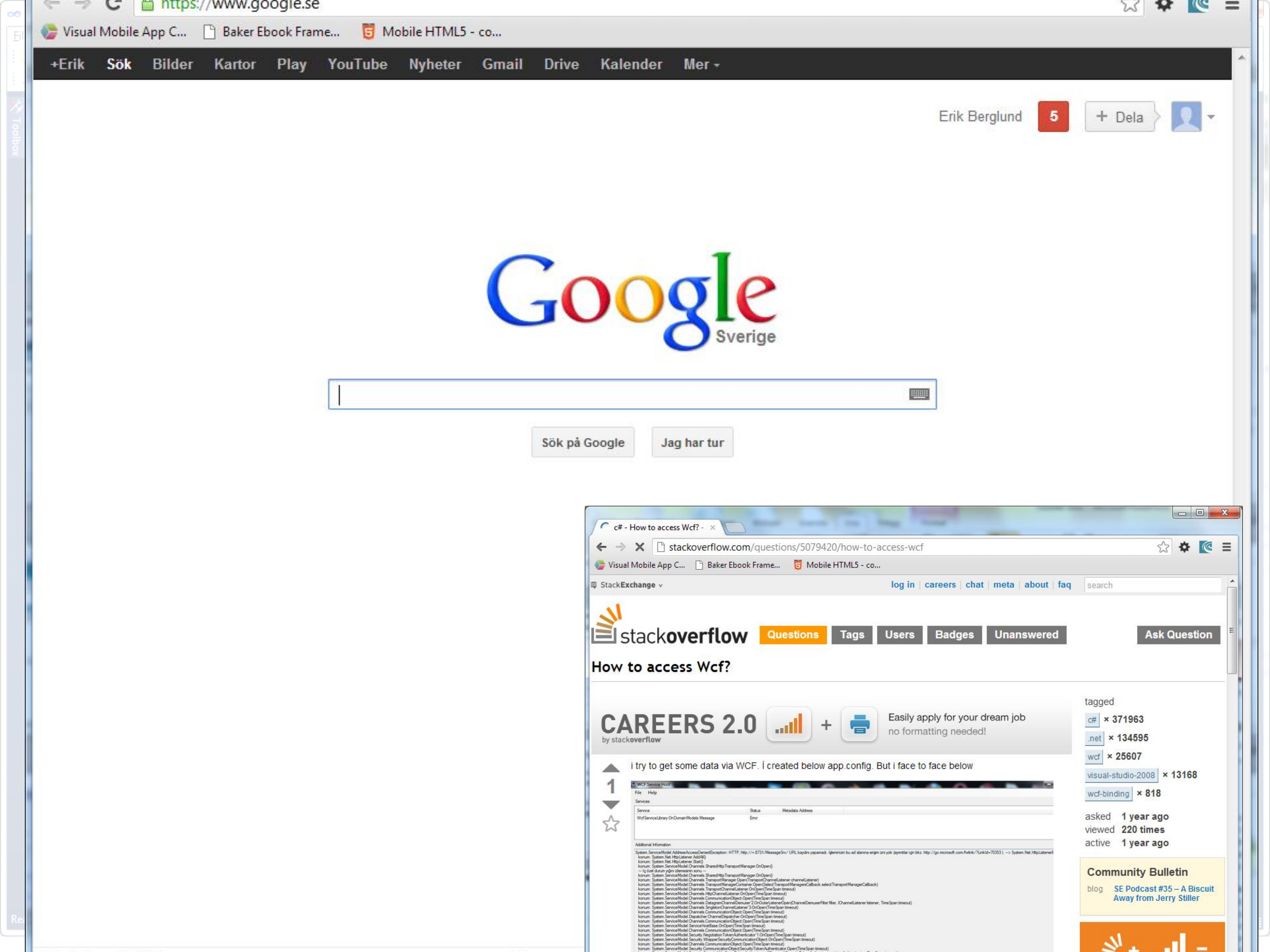
About this web site

LiU organization
Arts & Sciences
Educational Sciences
Health Sciences
Science and Engineering
Departments
Offices & Administration

At LiU
LiU Student Web
Education
Library
Prospective Employees
Alumni
Cooperation
Press room
Employees
LiU Fundraising

At IDA
Undergraduate studies
Final Thesis
Doctoral studies
Research
Organization
Internal
Emergency

Share   Top of page

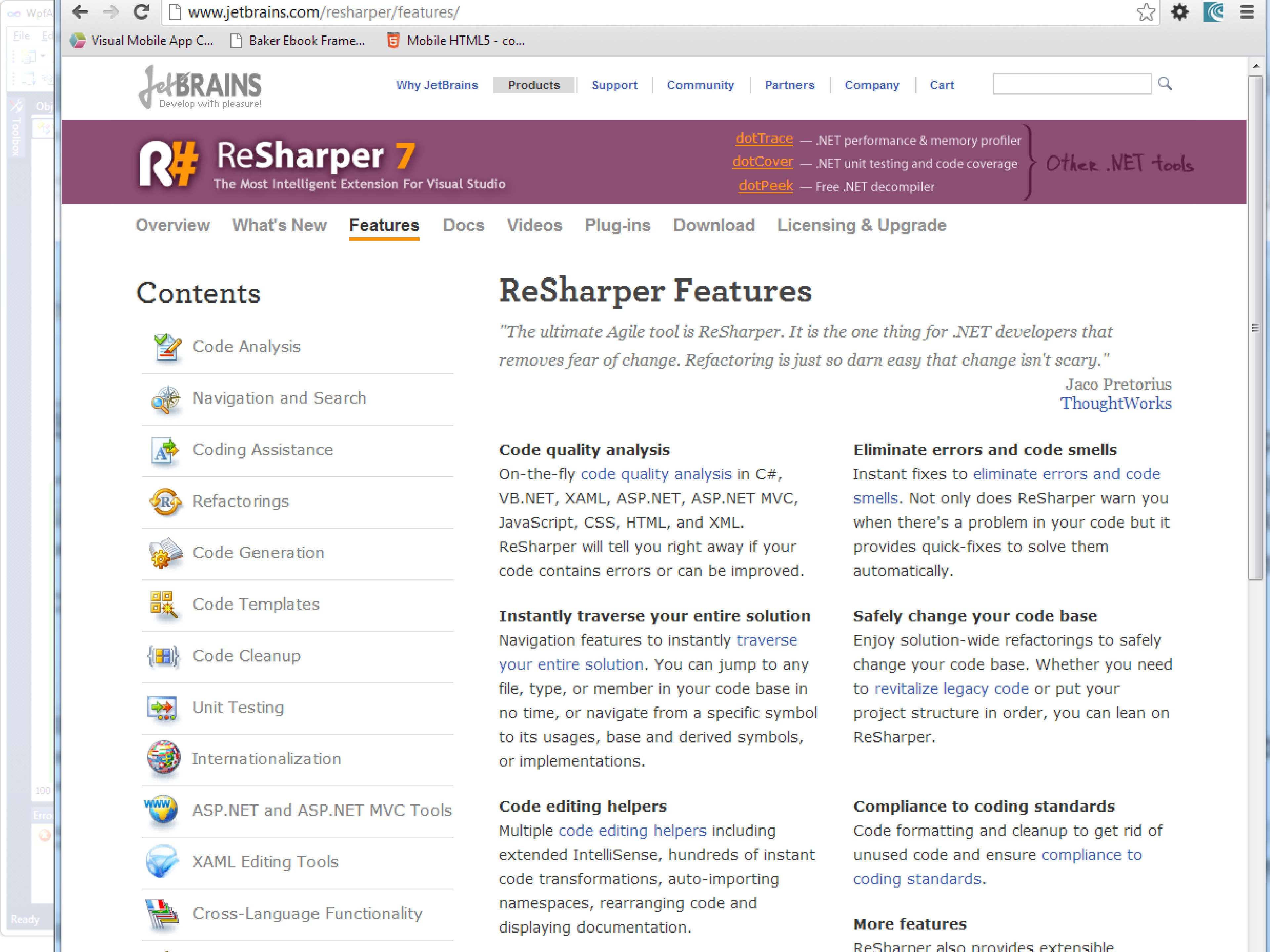


How to access Wcf?

i try to get some data via WCF. İ created below app.config. But i face to face below

Service	Status	Metadata Address
WcfServiceLibrary.OnDomainModels.Message	Error	

```
System.ServiceModel.AddressAccessDeniedException: HTTP. http://+:8731/MessageSrv/ URL kaydını yapamadı. İşleminizin bu ad alanına erişim izni yok (ayrıntılar için bkz. http://go.microsoft.com/fwlink/?LinkId=70353 ). ---> System.Net.HttpListenerE
   konum: System.Net.HttpListener.AddAll()
   konum: System.Net.HttpListener.Start()
   konum: System.ServiceModel.Channels.SharedHttpTransportManager.OnOpen()
   --- İç özel durum yığını izlemesinin sonu ---
   konum: System.ServiceModel.Channels.SharedHttpTransportManager.OnOpen()
   konum: System.ServiceModel.Channels.TransportManager.Open(TransportChannelListener channelListener)
   konum: System.ServiceModel.Channels.TransportManagerContainer.Open(SelectTransportManagersCallback selectTransportManagerCallback)
   konum: System.ServiceModel.Channels.TransportChannelListener.OnOpen(TimeSpan timeout)
   konum: System.ServiceModel.Channels.HttpChannelListener.OnOpen(TimeSpan timeout)
   konum: System.ServiceModel.Channels.CommunicationObject.Open(TimeSpan timeout)
   konum: System.ServiceModel.Channels.DatagramChannelDemuxer`2.OnOuterListenerOpen(ChannelDemuxerFilter filter, IChannelListener listener, TimeSpan timeout)
   konum: System.ServiceModel.Channels.SingletonChannelListener`3.OnOpen(TimeSpan timeout)
   konum: System.ServiceModel.Channels.CommunicationObject.Open(TimeSpan timeout)
   konum: System.ServiceModel.Dispatcher.ChannelDispatcher.OnOpen(TimeSpan timeout)
   konum: System.ServiceModel.Channels.CommunicationObject.Open(TimeSpan timeout)
   konum: System.ServiceModel.ServiceHostBase.OnOpen(TimeSpan timeout)
   konum: System.ServiceModel.Channels.CommunicationObject.Open(TimeSpan timeout)
   konum: System.ServiceModel.Security.NegotiationTokenAuthenticator`1.OnOpen(TimeSpan timeout)
   konum: System.ServiceModel.Security.WrapperSecurityCommunicationObject.OnOpen(TimeSpan timeout)
   konum: System.ServiceModel.Channels.CommunicationObject.Open(TimeSpan timeout)
   konum: System.ServiceModel.Security.CommunicationObjectSecurityTokenAuthenticator.Open(TimeSpan timeout)
```



ReSharper 7

The Most Intelligent Extension For Visual Studio

dotTrace — .NET performance & memory profiler
dotCover — .NET unit testing and code coverage
dotPeek — Free .NET decompiler

Other .NET tools

Contents

ReSharper Features

> *"The ultimate Agile tool is ReSharper. It is the one thing for .NET developers that removes fear of change. Refactoring is just so darn easy that change isn't scary."*
>
> Jaco Pretorius
> ThoughtWorks

Code quality analysis
On-the-fly code quality analysis in C#, VB.NET, XAML, ASP.NET, ASP.NET MVC, JavaScript, CSS, HTML, and XML. ReSharper will tell you right away if your code contains errors or can be improved.

Instantly traverse your entire solution
Navigation features to instantly traverse your entire solution. You can jump to any file, type, or member in your code base in no time, or navigate from a specific symbol to its usages, base and derived symbols, or implementations.

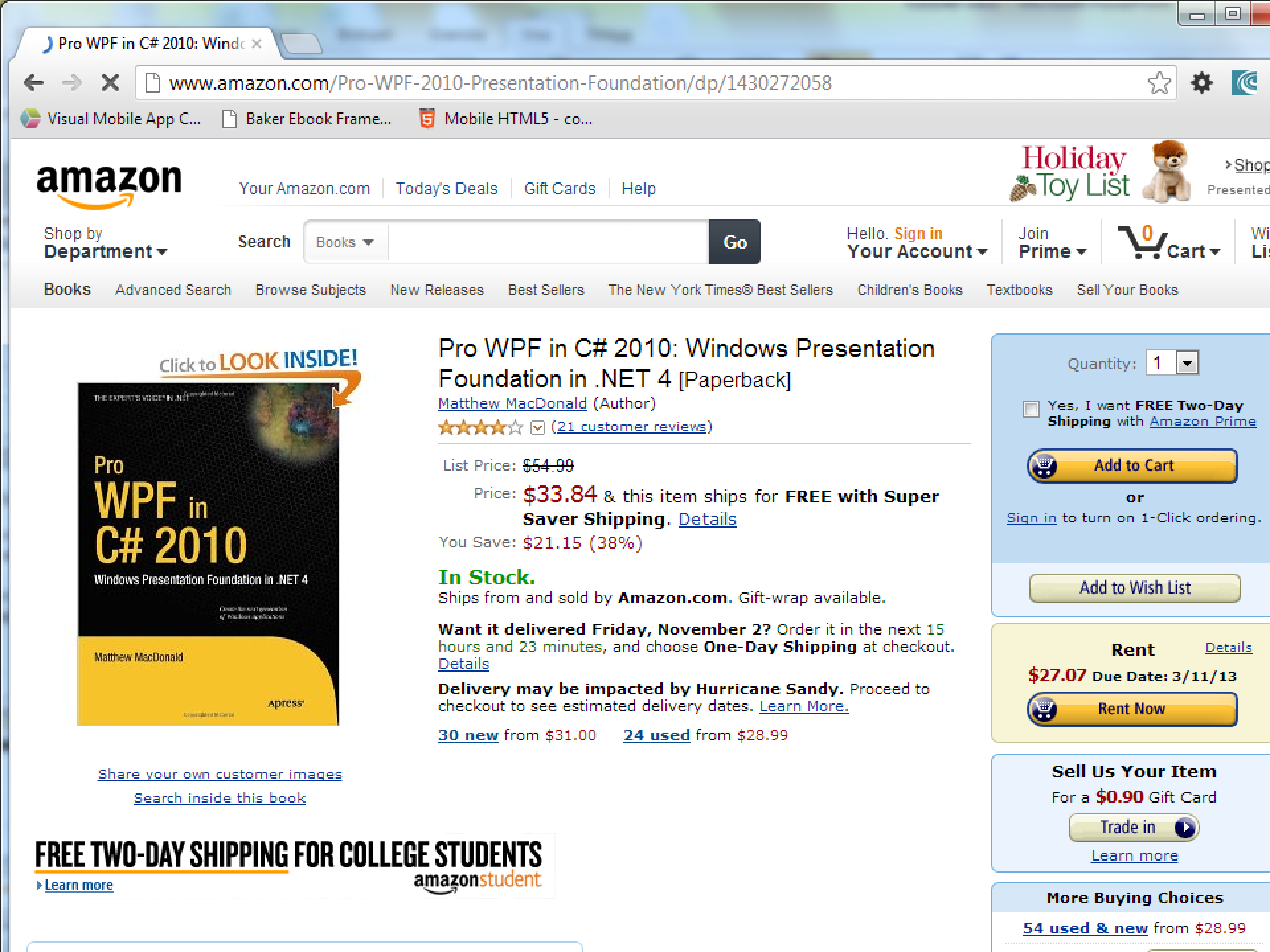
Code editing helpers
Multiple code editing helpers including extended IntelliSense, hundreds of instant code transformations, auto-importing namespaces, rearranging code and displaying documentation.

Eliminate errors and code smells
Instant fixes to eliminate errors and code smells. Not only does ReSharper warn you when there's a problem in your code but it provides quick-fixes to solve them automatically.

Safely change your code base
Enjoy solution-wide refactorings to safely change your code base. Whether you need to revitalize legacy code or put your project structure in order, you can lean on ReSharper.

Compliance to coding standards
Code formatting and cleanup to get rid of unused code and ensure compliance to coding standards.

More features
ReSharper also provides extensible



Pro WPF in C# 2010: Windows Presentation Foundation in .NET 4 [Paperback]

Matthew MacDonald (Author)

★★★★☆ (21 customer reviews)

List Price: ~~$54.99~~

Price: $33.84 & this item ships for **FREE with Super Saver Shipping**. Details

You Save: $21.15 (38%)

In Stock.

Ships from and sold by **Amazon.com**. Gift-wrap available.

Want it delivered Friday, November 2? Order it in the next 15 hours and 23 minutes, and choose **One-Day Shipping** at checkout. Details

Delivery may be impacted by Hurricane Sandy. Proceed to checkout to see estimated delivery dates. Learn More.

30 new from $31.00   24 used from $28.99

Share your own customer images

Search inside this book

FREE TWO-DAY SHIPPING FOR COLLEGE STUDENTS

amazonstudent

Learn more

Quantity: 1

□ Yes, I want **FREE Two-Day Shipping** with Amazon Prime

Add to Cart

or

Sign in to turn on 1-Click ordering.

Add to Wish List

Rent Details

$27.07 Due Date: 3/11/13

Rent Now

Sell Us Your Item

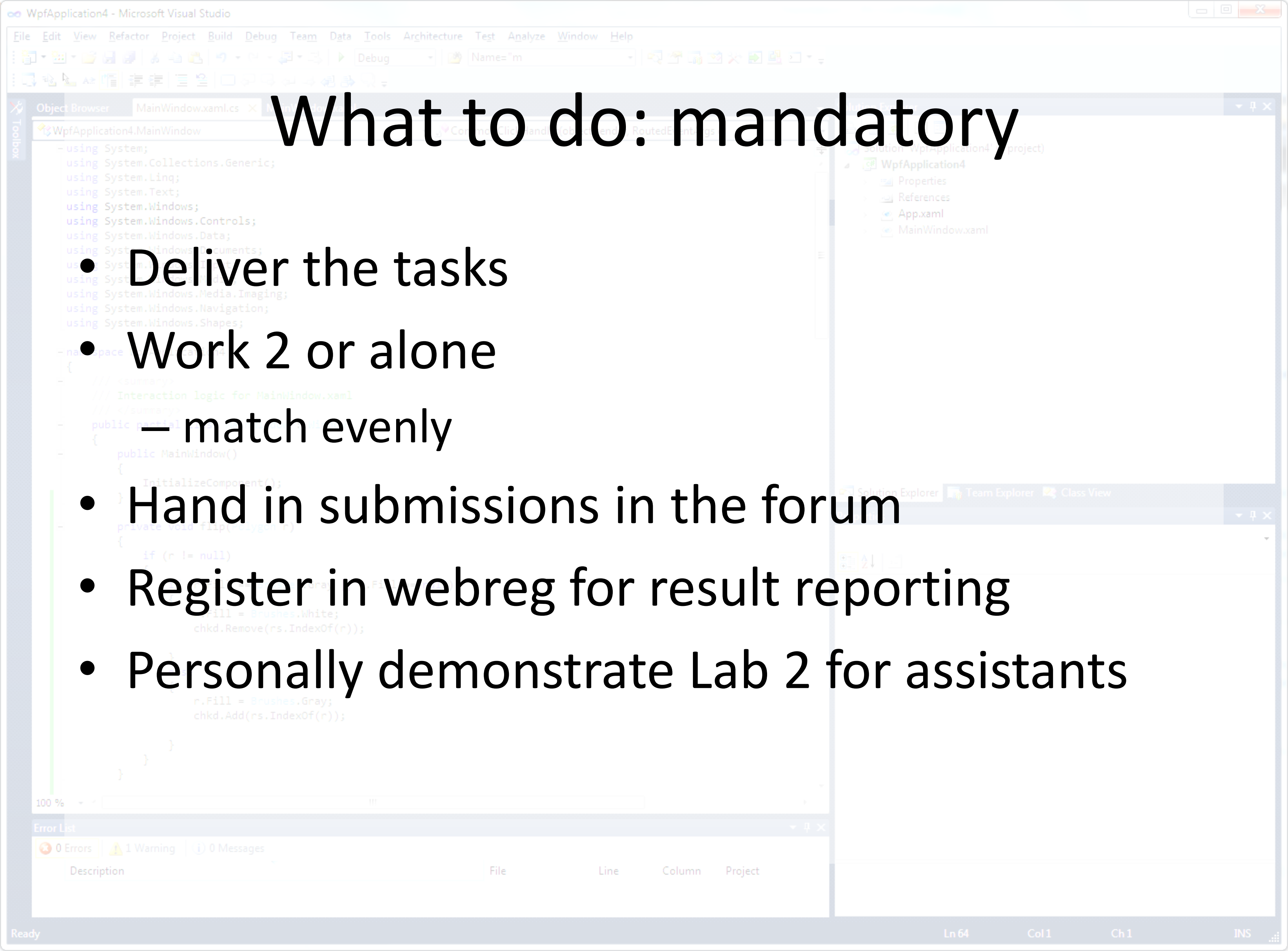
For a **$0.90** Gift Card

Trade in

Learn more

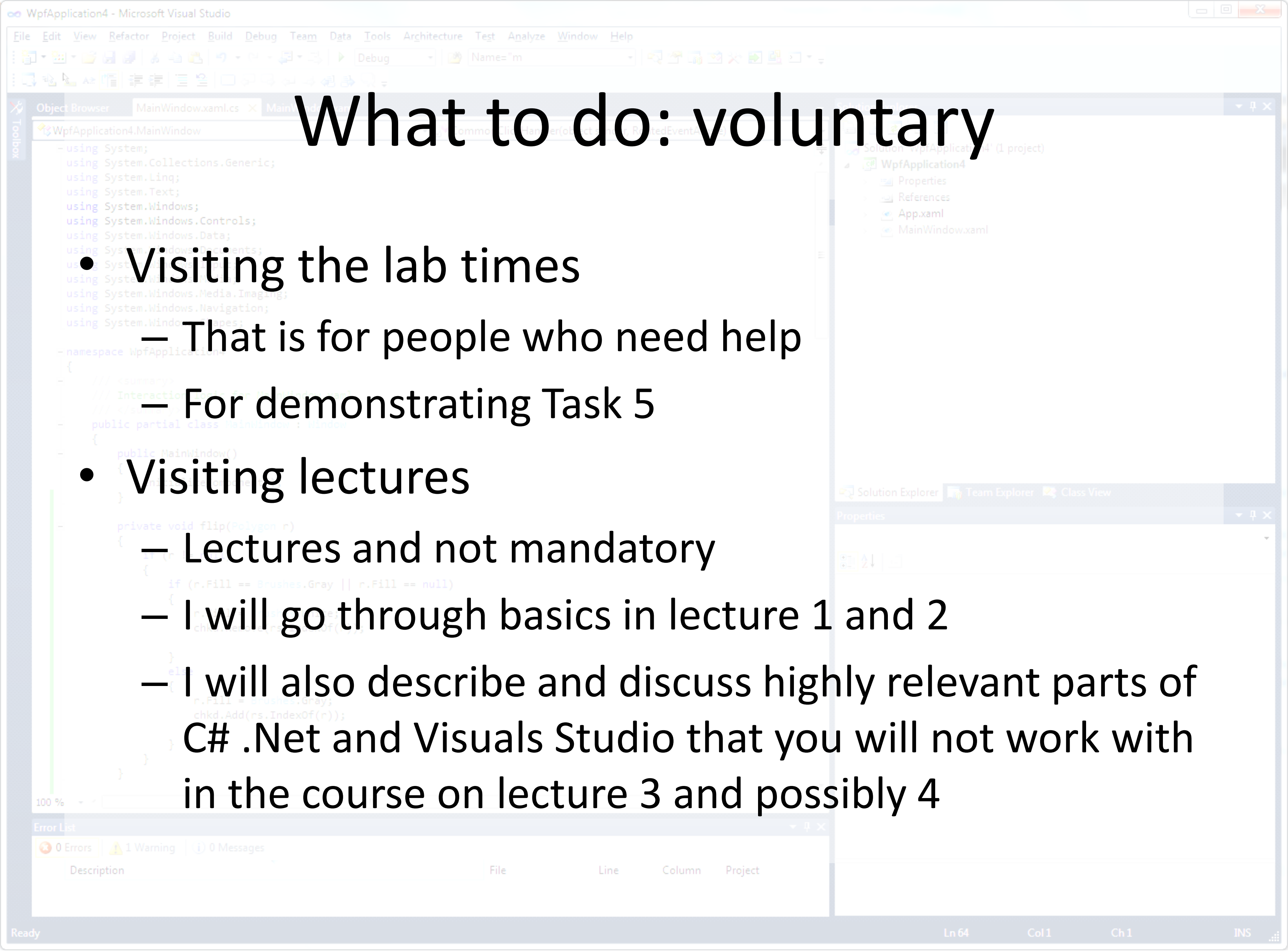
More Buying Choices

54 used & new from $28.99



What to do: mandatory

- Deliver the tasks
- Work 2 or alone
  - match evenly
- Hand in submissions in the forum
- Register in webreg for result reporting
- Personally demonstrate Lab 2 for assistants

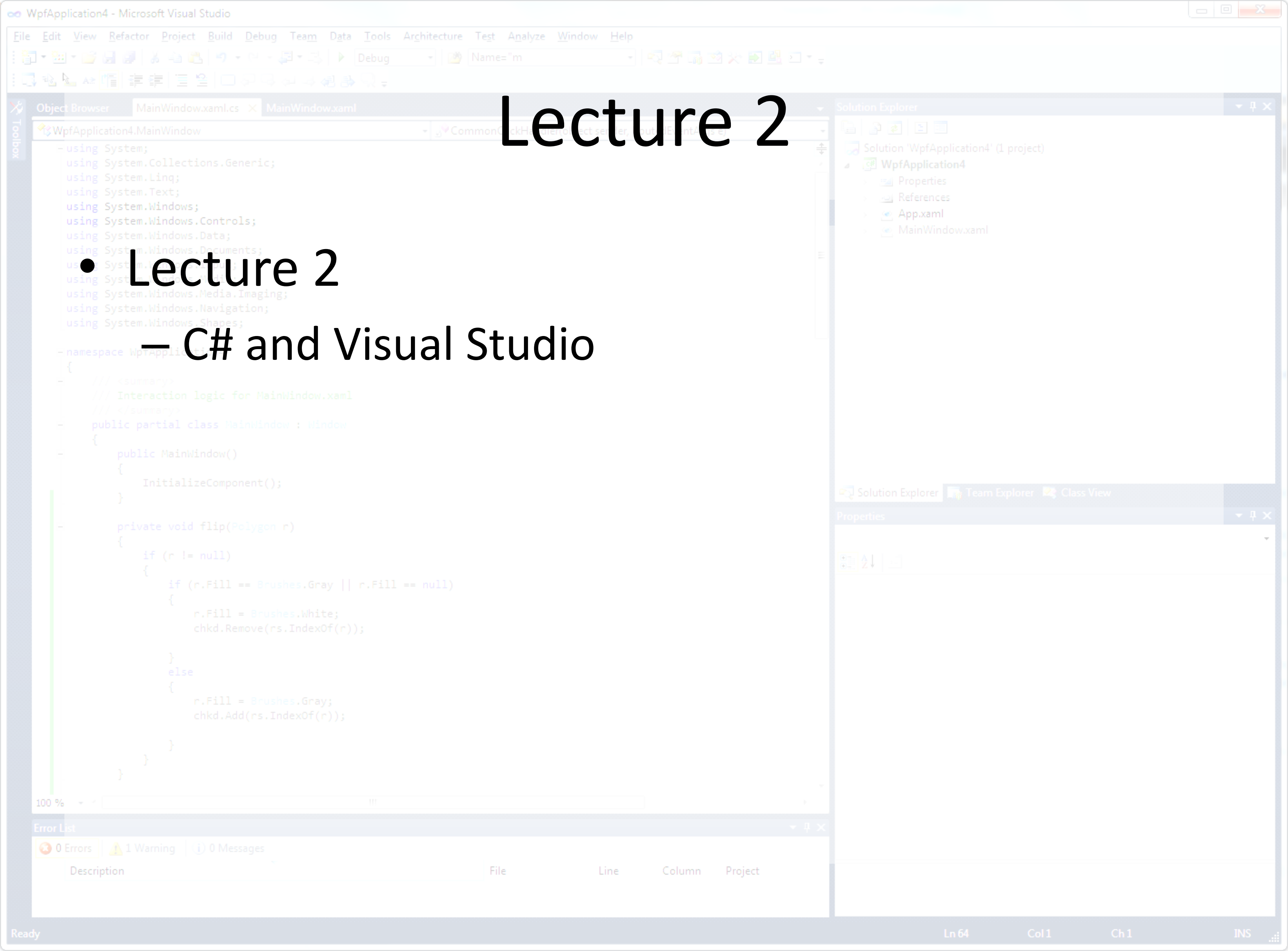


What to do: voluntary

- Visiting the lab times
  - That is for people who need help
  - For demonstrating Task 5
- Visiting lectures
  - Lectures and not mandatory
  - I will go through basics in lecture 1 and 2
  - I will also describe and discuss highly relevant parts of C# .Net and Visuals Studio that you will not work with in the course on lecture 3 and possibly 4

Lecture 2

- Lecture 2
  - C# and Visual Studio



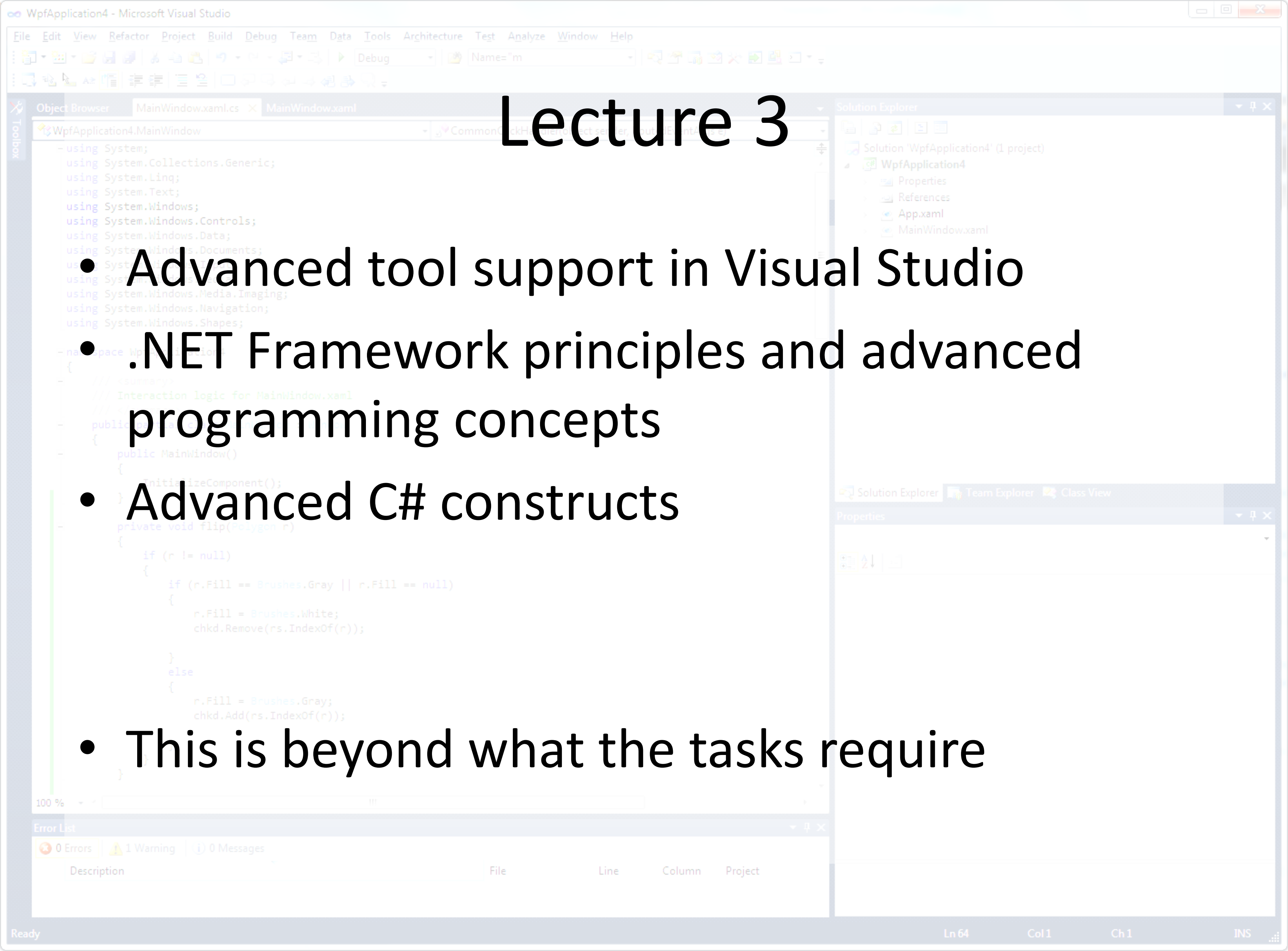
WpfApplication4 - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Team Data Tools Architecture Test Analyze Window Help
Debug
Name="m
Object Browser
MainWindow.xaml.cs
MainWindow.xaml
WpfApplication4.MainWindow
Toolbox

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
[illegible]
[illegible]
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace WpfApplication4
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }

        private void flip(Polygon r)
        {
            if (r != null)
            {
                if (r.Fill == Brushes.Gray || r.Fill == null)
                {
                    r.Fill = Brushes.White;
                    chkd.Remove(rs.IndexOf(r));

                }
                else
                {
                    r.Fill = Brushes.Gray;
                    chkd.Add(rs.IndexOf(r));

                }
            }
        }
```

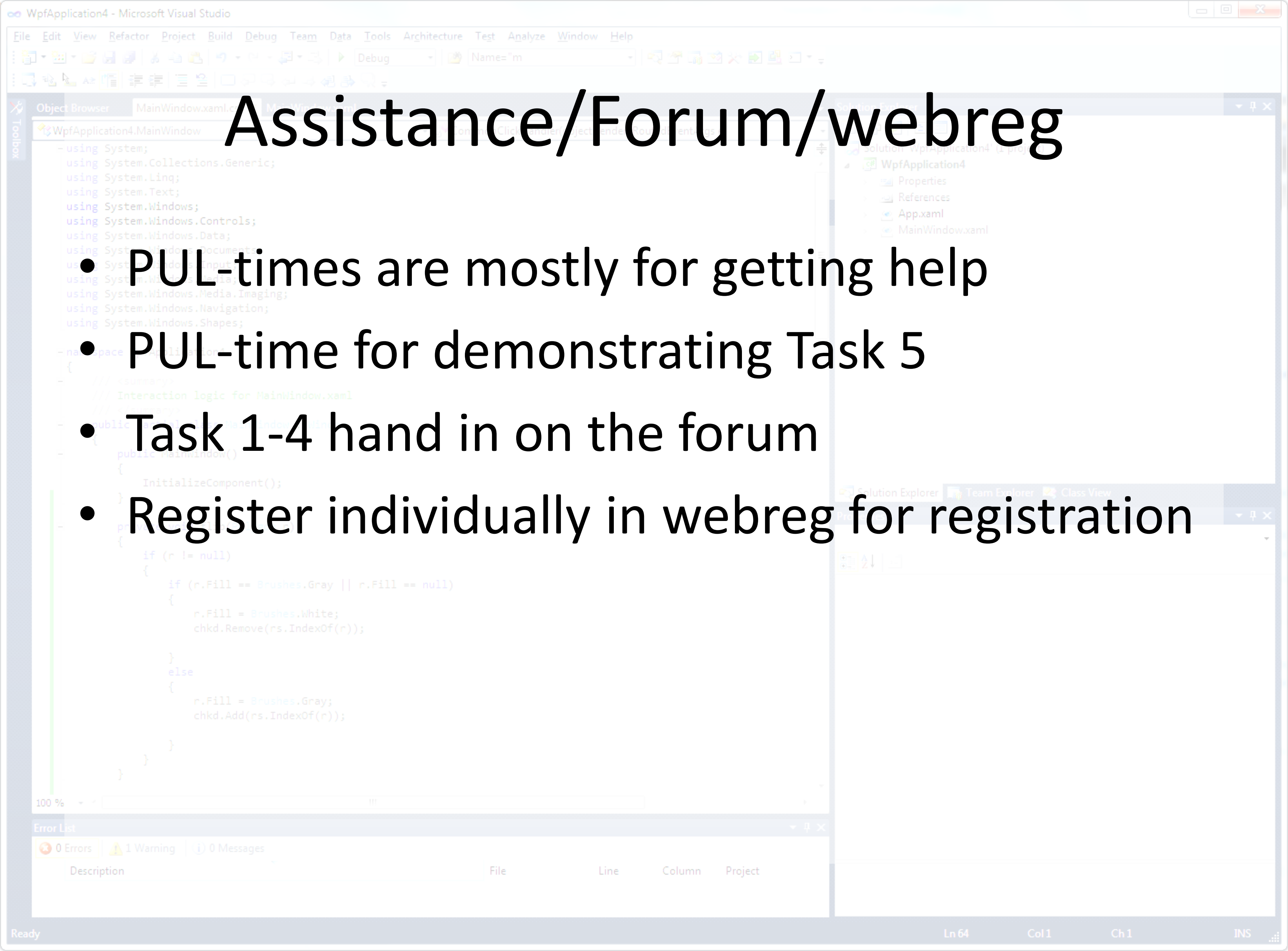
Lecture 3

- Advanced tool support in Visual Studio
- .NET Framework principles and advanced programming concepts
- Advanced C# constructs

- This is beyond what the tasks require



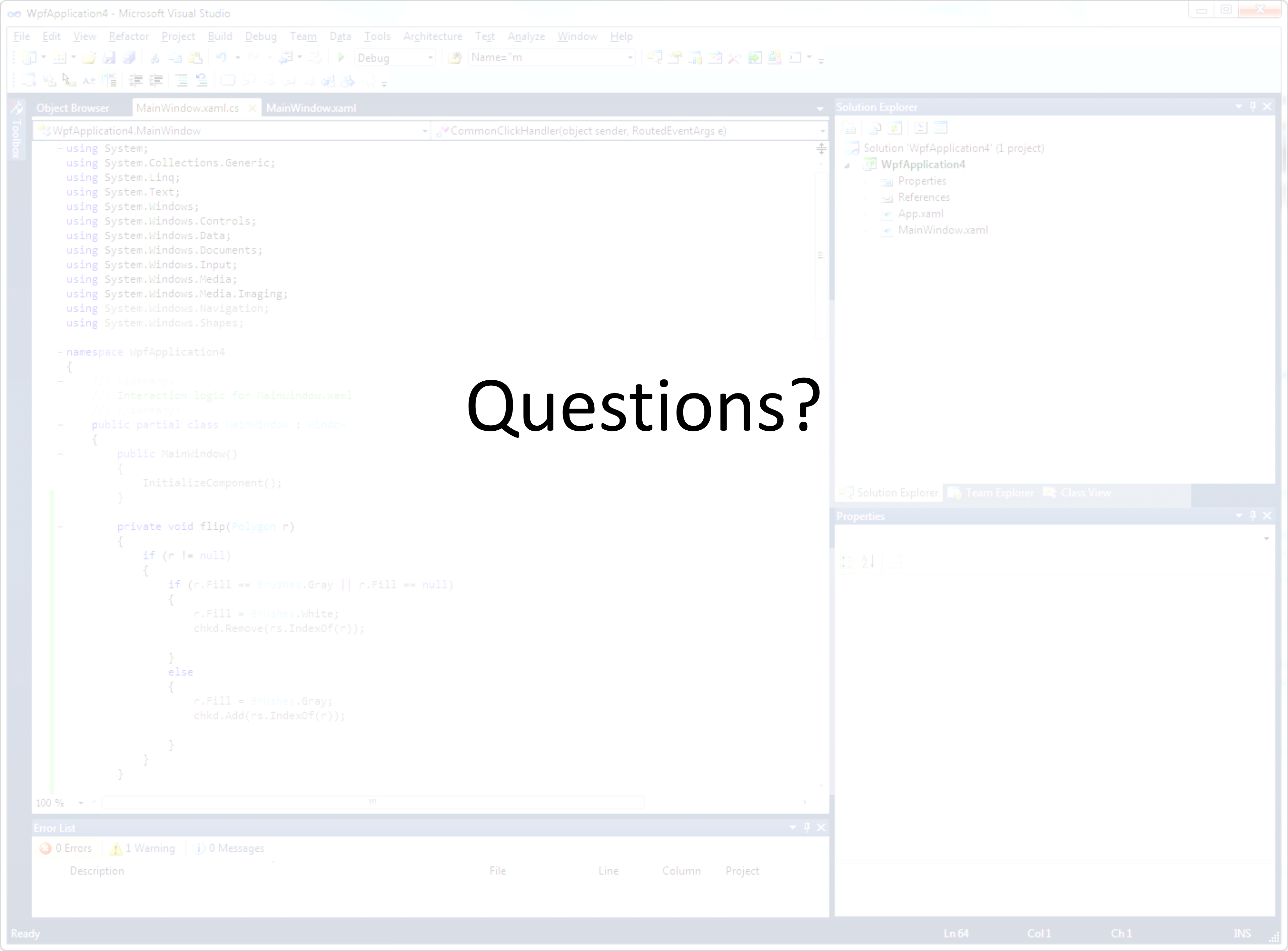
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace WpfApplication4
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }

        private void flip(Polygon r)
        {
            if (r != null)
            {
                if (r.Fill == Brushes.Gray || r.Fill == null)
                {
                    r.Fill = Brushes.White;
                    chkd.Remove(rs.IndexOf(r));

                }
                else
                {
                    r.Fill = Brushes.Gray;
                    chkd.Add(rs.IndexOf(r));


            }
        }
```



Assistance/Forum/webreg

- PUL-times are mostly for getting help
- PUL-time for demonstrating Task 5
- Task 1-4 hand in on the forum
- Register individually in webreg for registration

Questions?



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace WpfApplication4
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }

        private void flip(Polygon r)
        {
            if (r != null)
            {
                if (r.Fill == Brushes.Gray || r.Fill == null)
                {
                    r.Fill = Brushes.White;
                    chkd.Remove(rs.IndexOf(r));

                }
                else
                {
                    r.Fill = Brushes.Gray;
                    chkd.Add(rs.IndexOf(r));

                }
            }
        }
```