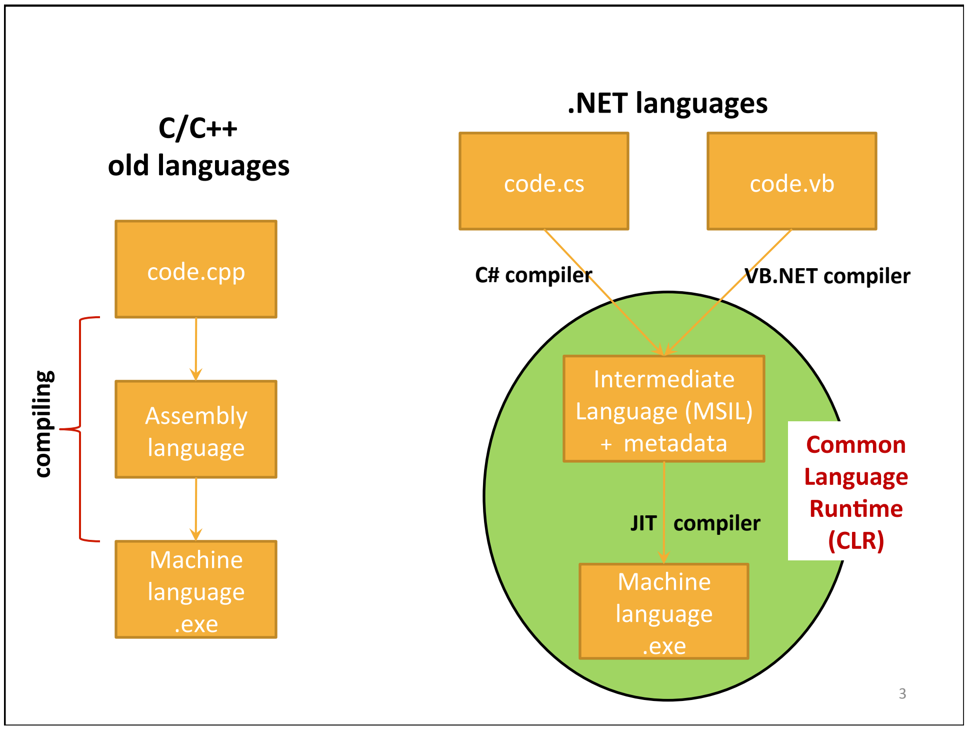
Review of C#

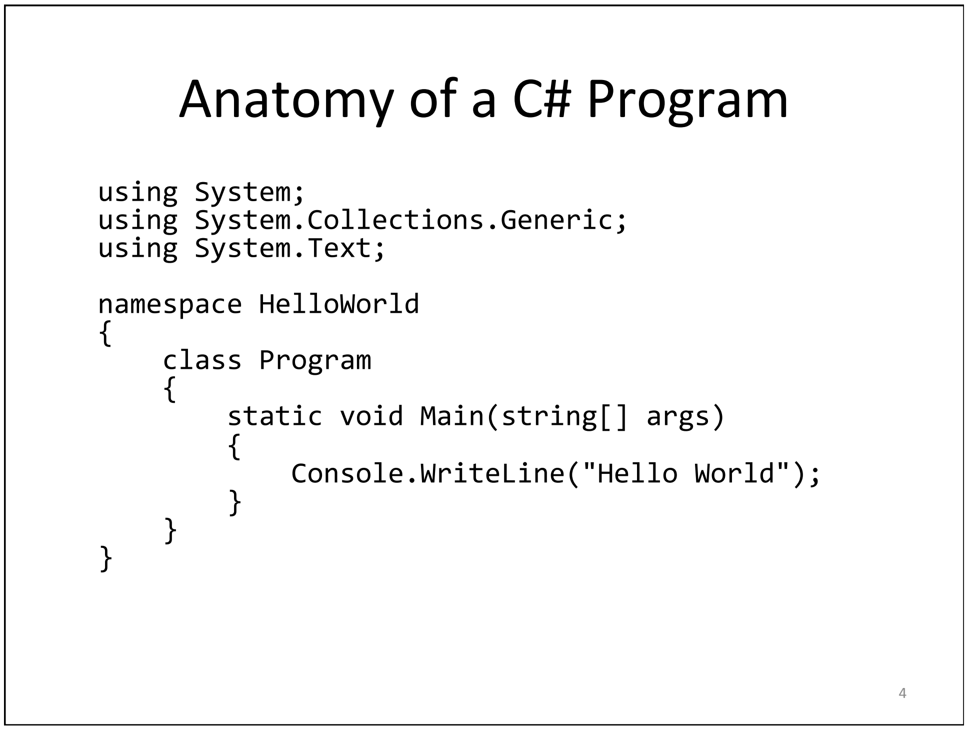
Dominic Duggan
Stevens Institute of Technology

1

C# BASICS

2





Anatomy of a C# Program

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

namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World");
        }
    }
}
```

4

Using …

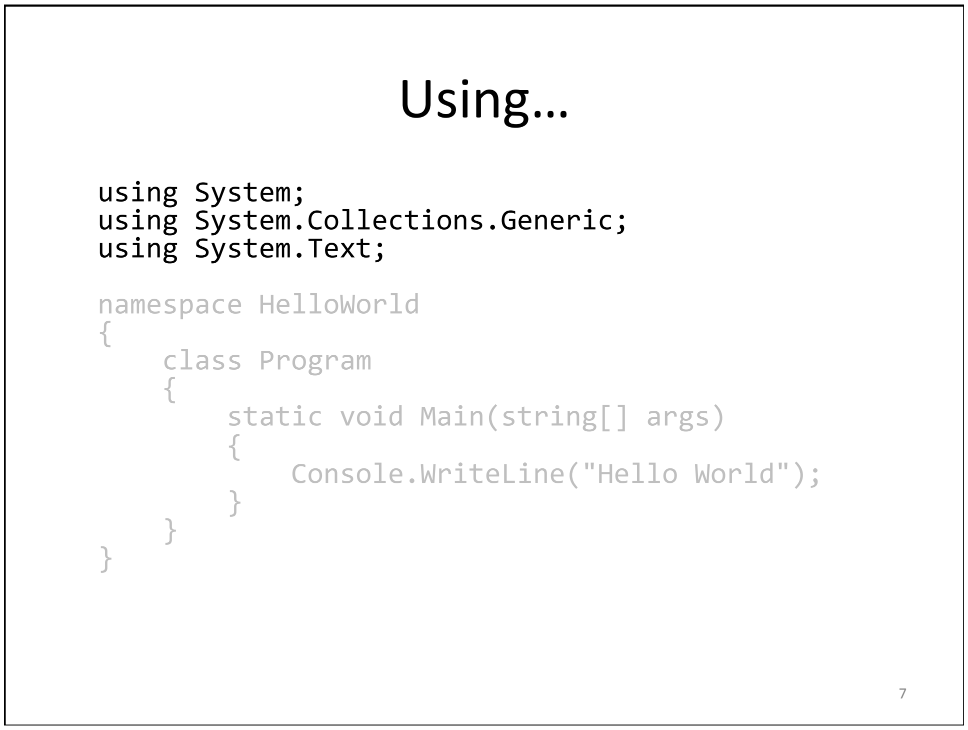
```
using System.Collections.Generic;
using System.Text;

namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            System.Console.WriteLine("Hello World");
        }
    }
}
```

5

Using …

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

namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World");
        }
    }
}
```

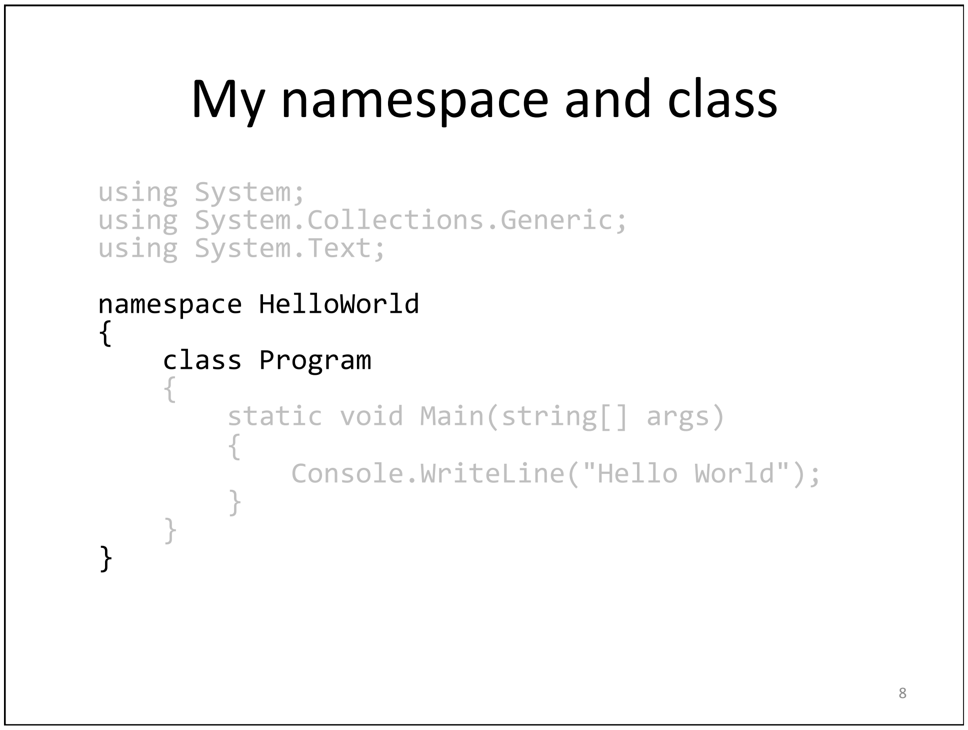
6



Using…

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

namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World");
        }
    }
}
```

7



My namespace and class

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

namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World");
        }
    }
}
```

8

The main program

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

namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World”);
        }
    }
}
```

9

The main program

```
using System.Windows.Forms;
using System.Collections.Generic;
using System.Text;

namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            MessageBox.Show("Hello World");
        }
    }
}
```

10

C# Data Types

- There are 15 data types in C#
- Eight of them represent integers:
  - `byte, sbyte, short, ushort, int, uint, long, ulong`
- Two of them represent floating point numbers
  - `float, double`
- One of them represents decimals:
  - `decimal`
- One of them represents boolean values:
  - `bool`
- One of them represents characters:
  - `char`
- One of them represents strings:
  - `string`
- One of them represents objects:
  - `object`

11

Numeric Data Types

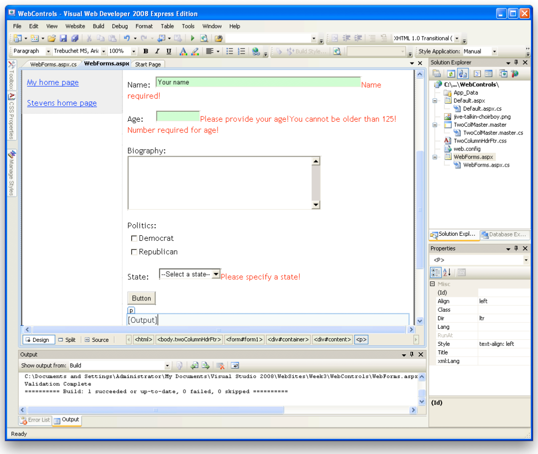
- The difference between the various numeric types is their size, and therefore the values they can store:

Type	Storage	Range
byte	8 bits	0 - 255
sbyte	8 bits	-128 - 127
short	16 bits	-32,768 - 32767
ushort	16 bits	0 - 65537
int	32 bits	-2,147,483,648 – 2,147,483,647
uint	32 bits	0 – 4,294,967,295
long	64 bits	-9×10^{18} to 9×10^{18}
ulong	64 bits	$0 – 1.8\times10^{19}$
decimal	128 bits	$\pm1.0\times10^{-28}$; $\pm7.9\times10^{28}$ with 28-29 significant digits
float	32 bits	$\pm1.5\times10^{-45}$; $\pm3.4\times10^{38}$ with 7 significant digits
double	64 bits	$\pm5.0\times10^{-324}$; $\pm1.7\times10^{308}$ with 15-16 significant digits

12

OBJECTS

Objects



My home page

Stevens home page

Name: Your name Name required!

Age: Please provide your age!You cannot be older than 125! Number required for age!

Biography:

Politics:

☐ Democrat

☐ Republican

State: --Select a state-- Please specify a state!

Button

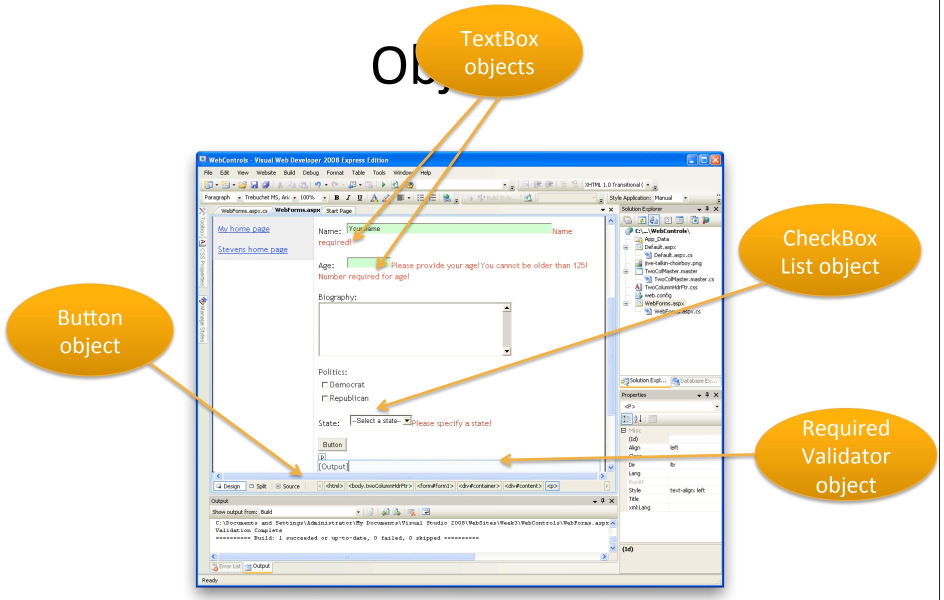
[Output]

Misc	
(Id)	
Align	left
Class	
Dir	ltr
Lang	
RunAt	
Style	text-align: left
Title	
xml:Lang	

(Id)

```
C:\Documents and Settings\Administrator\My Documents\Visual Studio 2008\WebSites\Week3\WebControls\WebForms.aspx
Validation Complete
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
```

Ob


TextBox objects
CheckBox List object
Button object
Required Validator object
WebControls - Visual Web Developer 2008 Express Edition
File Edit View Website Build Debug Format Table Tools Window Help
WebForms.aspx.cs WebForms.aspx Start Page
Solution Explorer
Properties
Design Split Source
Error List Output
Ready

My home page
Stevens home page

Name: Your name Name required!

Age: Please provide your age!You cannot be older than 125! Number required for age!

Biography:

Politics:
- Democrat
- Republican

State: --Select a state-- Please specify a state!

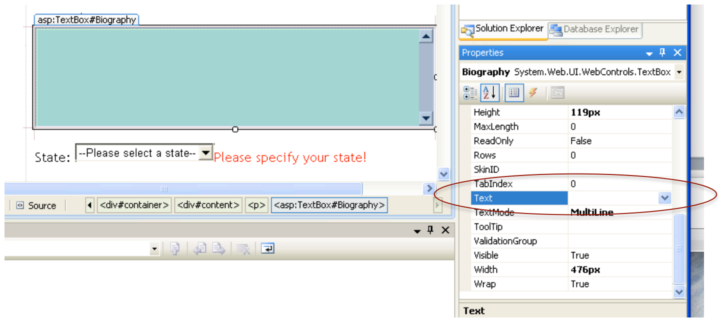
Button

[Output]

```
C:\Documents and Settings\Administrator\My Documents\Visual Studio 2008\WebSites\Week3\WebControls\WebForms.aspx
Validation Complete
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
```

What’s an Object?

- Objects were invented to model the real world!
- They have memory (fields and properties)



State: --Please select a state-- Please specify your state!

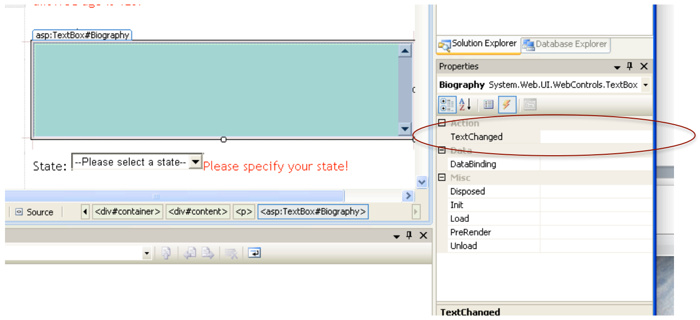
Biography System.Web.UI.WebControls.TextBox

Height	119px
MaxLength	0
ReadOnly	False
Rows	0
SkinID	
TabIndex	0
Text	
TextMode	MultiLine
ToolTip	
ValidationGroup	
Visible	True
Width	476px
Wrap	True

Text

What's an Object?

- Objects were invented to model the real world!
- They understand certain messages



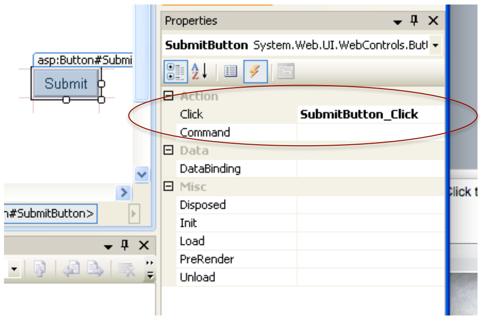
Biography System.Web.UI.WebControls.TextBox

TextChanged	
DataBinding	
Disposed	
Init	
Load	
PreRender	
Unload	

17

What's an Object?

- Objects were invented to model the real world!
- They understand certain messages



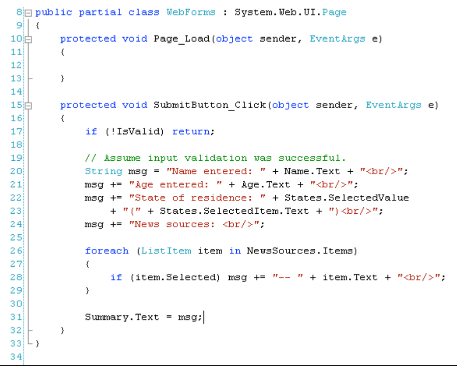
SubmitButton System.Web.UI.WebControls.Butt

Click	SubmitButton_Click
Command	
DataBinding	
Disposed	
Init	
Load	
PreRender	
Unload	

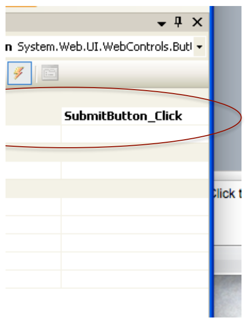
18

What's an Object?

- Objects were invented to model the real world!
- Method invocation on message receipt



```
public partial class WebForms : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void SubmitButton_Click(object sender, EventArgs e)
    {
        if (!IsValid) return;

        // Assume input validation was successful.
        String msg = "Name entered: " + Name.Text + "<br/>";
        msg += "Age entered: " + Age.Text + "<br/>";
        msg += "State of residence: " + States.SelectedValue
            + "(" + States.SelectedItem.Text + ")<br/>";
        msg += "News sources: <br/>";

        foreach (ListItem item in NewsSources.Items)
        {
            if (item.Selected) msg += "-- " + item.Text + "<br/>";
        }

        Summary.Text = msg;
    }
}
```



19

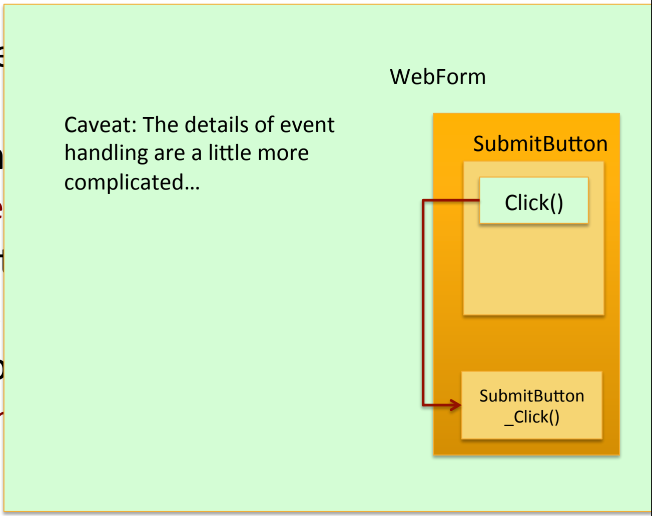
What's an Object?

- Objects were invented to model the real world!
- They have memory (fields and properties)
  - TextBox: Text property
- They understand certain messages
  - Button: Click event
- Method invocation on message receipt
  - Invoke SubmitButton_Click event handler on Click event

20

What's an Object?

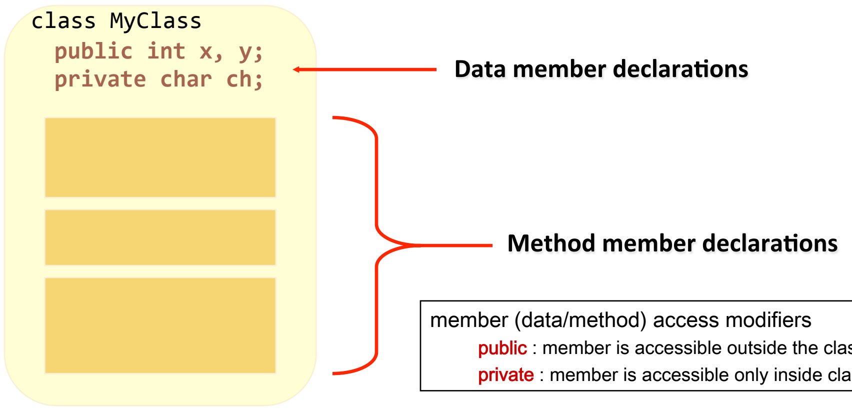
- Objects were
  world!
- They have m
  - TextBox: Te
- They underst
  - Button: Clic
- Method invo
  - Invoke Subr
    event



Caveat: The details of event handling are a little more complicated…

21

Defining Classes



```
class MyClass
  public int x, y;
  private char ch;
```

member (data/method) access modifiers
- public : member is accessible outside the class
- private : member is accessible only inside class

22

Example: Declaring a counter

```
class Counter {

   private int internalCtr;
   public Counter () { internalCtr = 0; }
   public Counter (int init) { internalCtr = init; }

   public void click () {
        internalCtr++;
   }

   public int value () {
        return internalCtr;
   }
}
```

Example: Using a counter

```
class Test {

   public void Main (String[] args) {

        Counter ctr = new Counter(100);

        while (…) {
                …
                ctr.click ();
                …
        }

        Console.WriteLine (“Loop executed {0} times”,
                                ctr.value ());
   }
}
```

Properties: “Smart Fields”

```
public class Button: Control
{
   private string caption;

   public string Caption {
       get {
          return caption;
       }
       set {
          caption = value;
          Repaint();
       }
   }
}
```

```
Button b = new Button();
b.Caption = "OK";
String s = b.Caption;
```

25

INDEXERS

26

Indexers (“Smart arrays”)

```
public class ListBox : Control
{
   private string[] items;

   public string this [int index] {
      get {
         return items [index];
      }
      set {
       items [index] = value;
         Repaint();
      }
   }
}
```

```
ListBox listBox = new ListBox();
listBox[0] = "hello";
Console.WriteLine(listBox[0]);
```

Indexers (“Smart dictionaries”)

```
public class MyCookie
{
   private Dictionary<String,String> items;

   public string this [string key] {
       get {
           return items.Get (key);
       }

       set {
           items.Add (key, value);
       }
   }
}
```

```
MyCookie cookie = new MyCookie();
cookie[“Name”] = Name.Text;
…
string name = cookie[“Name”];
```

DELEGATES AND EVENT HANDLING

Delegates

- Basis for event handling
  - Stand-alone method (C: function pointer)
  - “Wrap” an existing method as a delegate

```
delegate double Func(double x);

Func func = new Func(Math.Sin);
double x = func(1.0);
```

Event Handlers

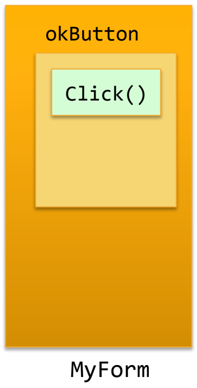
- Basis for event handling
  - Every event has an associated delegate

```
delegate void EventHandler(object x,
                           EventArgs xs);

class Button {
  event EventHandler Click;
  void OnClick (EventArgs e) {
     if (Click != null) Click (this, e);
  }
}
```

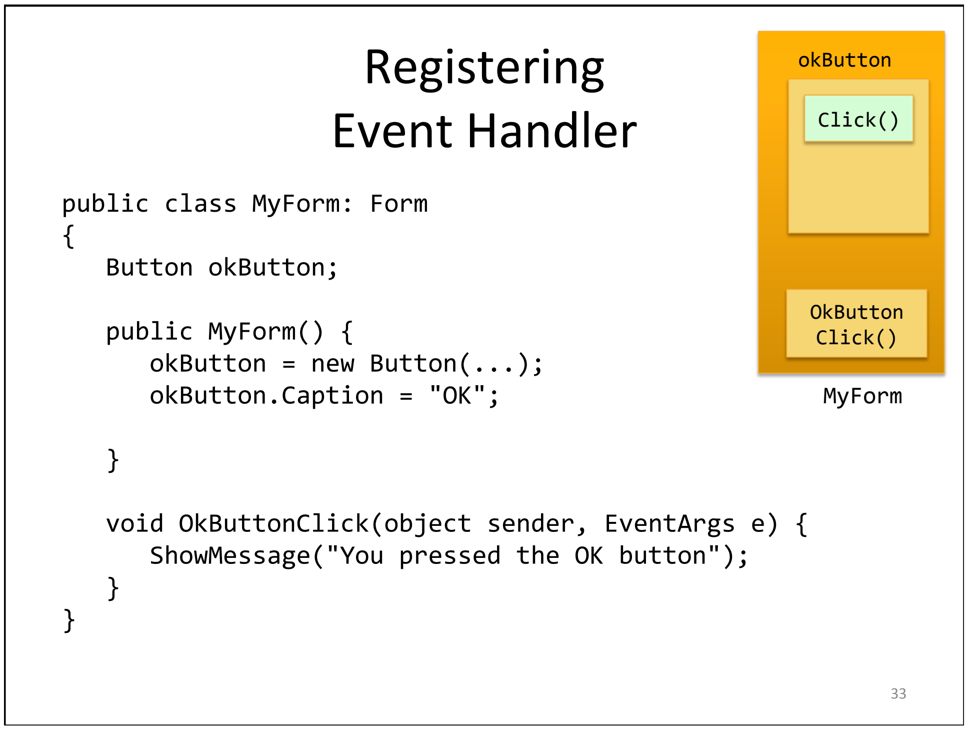
31

Registering Event Handler

```
public class MyForm: Form
{
   Button okButton;

   public MyForm() {
      okButton = new Button(...);
      okButton.Caption = "OK";

   }



}
```

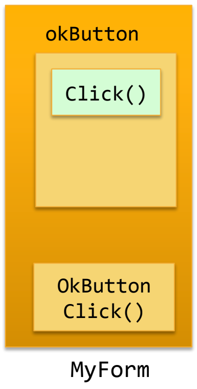


32

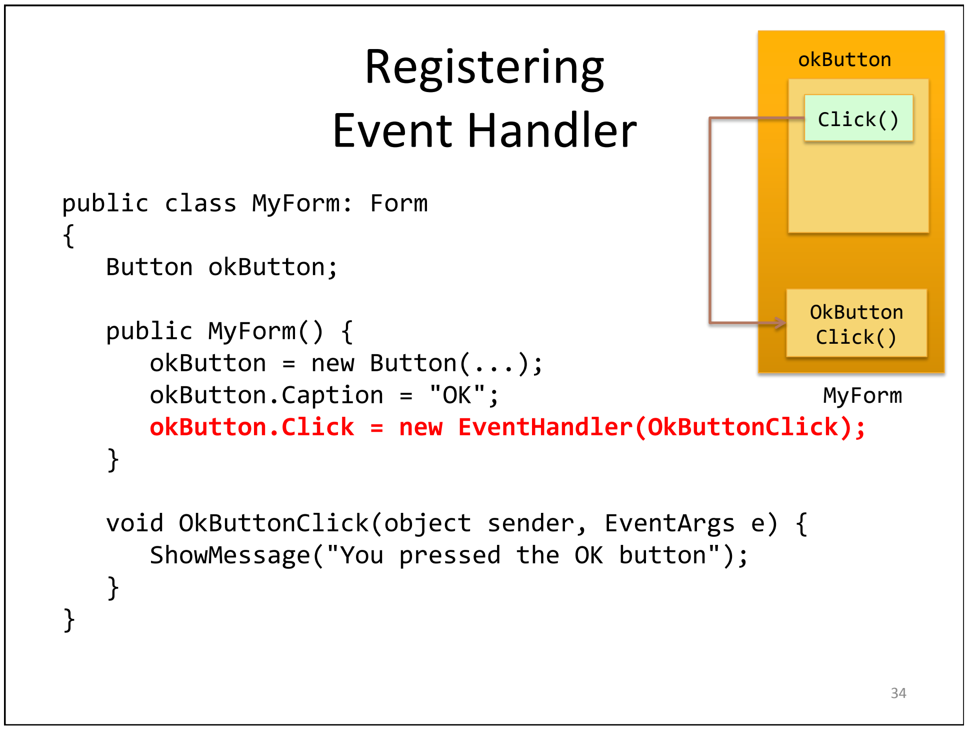


Registering Event Handler

```
public class MyForm: Form
{
   Button okButton;

   public MyForm() {
      okButton = new Button(...);
      okButton.Caption = "OK";

   }

   void OkButtonClick(object sender, EventArgs e) {
      ShowMessage("You pressed the OK button");
   }
}
```

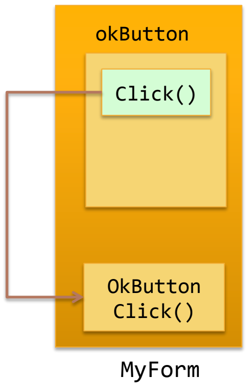


33



Registering Event Handler

```
public class MyForm: Form
{
   Button okButton;

   public MyForm() {
      okButton = new Button(...);
      okButton.Caption = "OK";
      okButton.Click = new EventHandler(OkButtonClick);
   }

   void OkButtonClick(object sender, EventArgs e) {
      ShowMessage("You pressed the OK button");
   }
}
```



34