

CO453 Application Programming

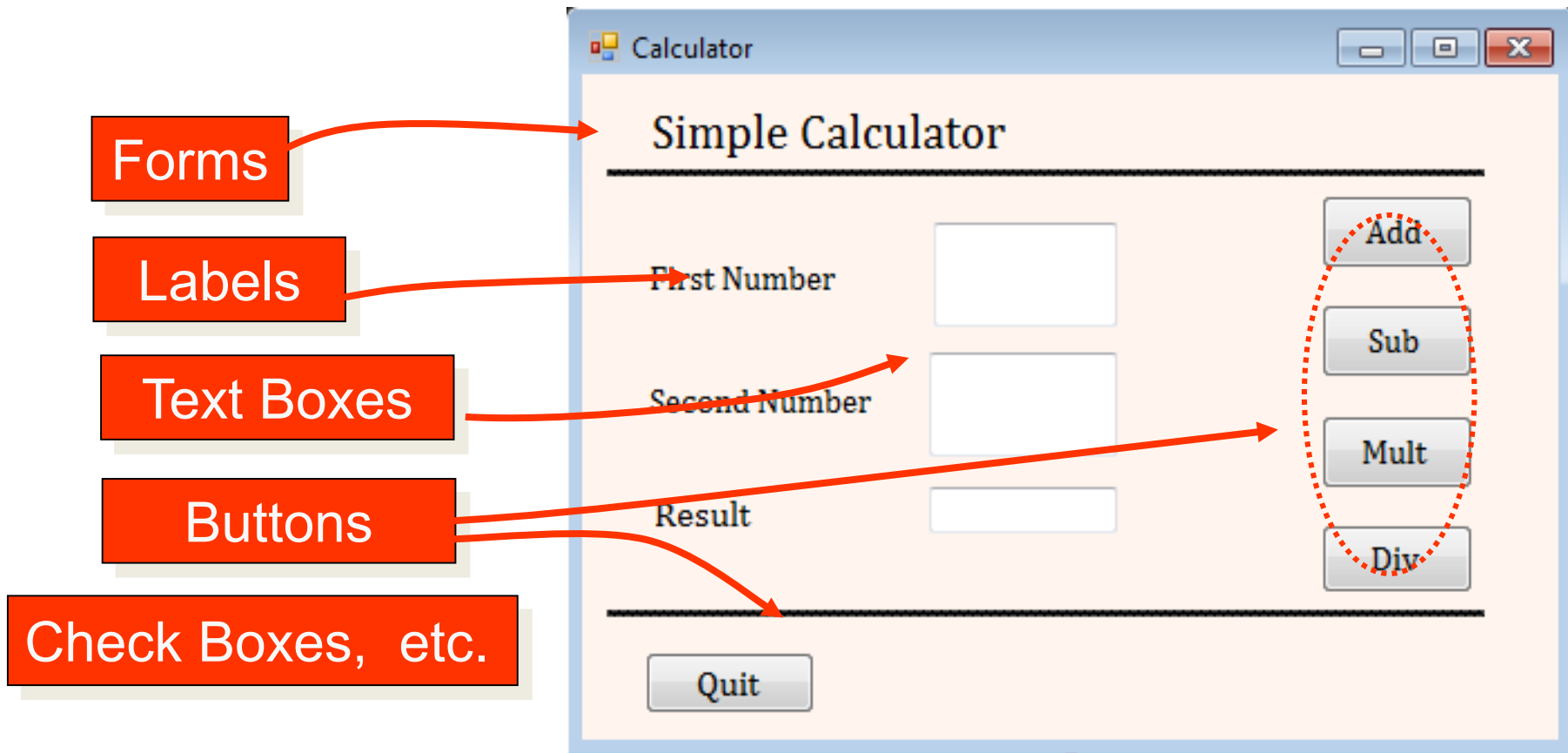
Week 5 – C# .NET part 1

Introduction to forms, and splash
screens

.NET C# is VISUAL

.NET programs run on MS Windows

.NET provides ready-made objects to create an easy-to-use visual user interface

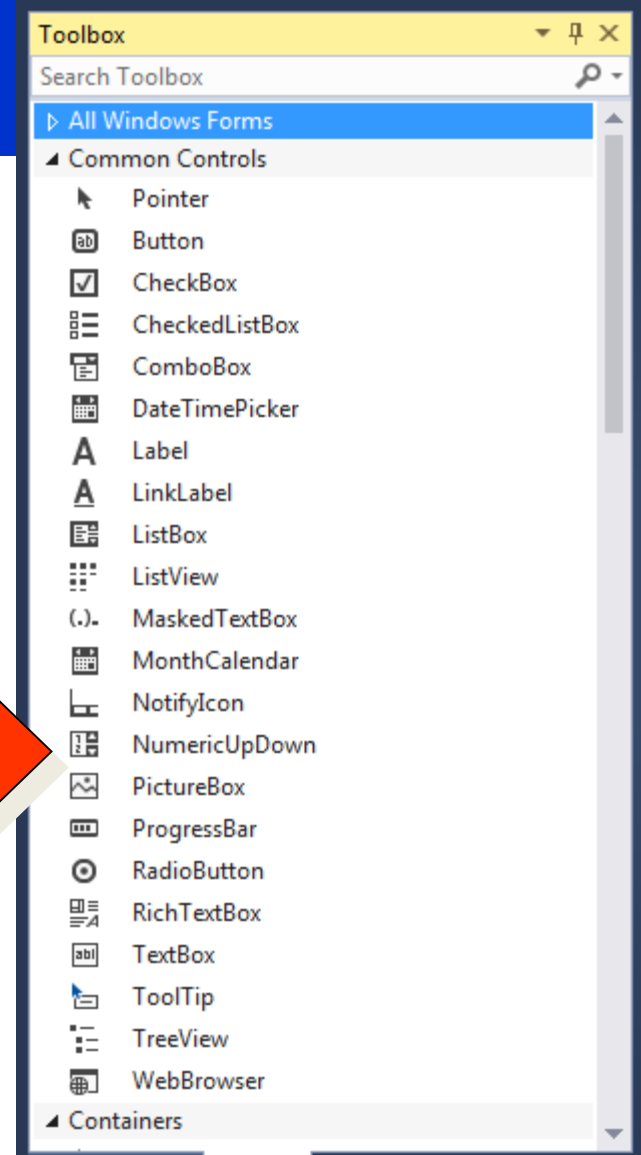


.NET Objects have PROPERTIES

We can change the properties of any object:

Name
BackColor
BackgroundImage
BorderStyle
Text
Font
AutoSize
etc.

Select any object
Then press the F4 key
to show its
Properties Window



C# .NET is Event-Driven

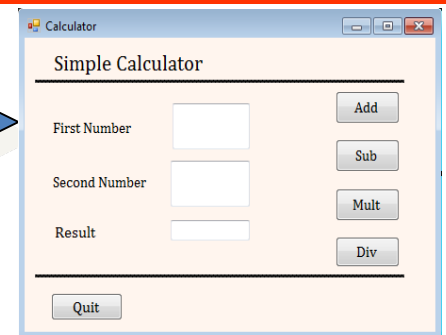


.NET programs respond to Events
a mouse-click
a key press
a form loading, etc.

.NET programs can be constructed differently:

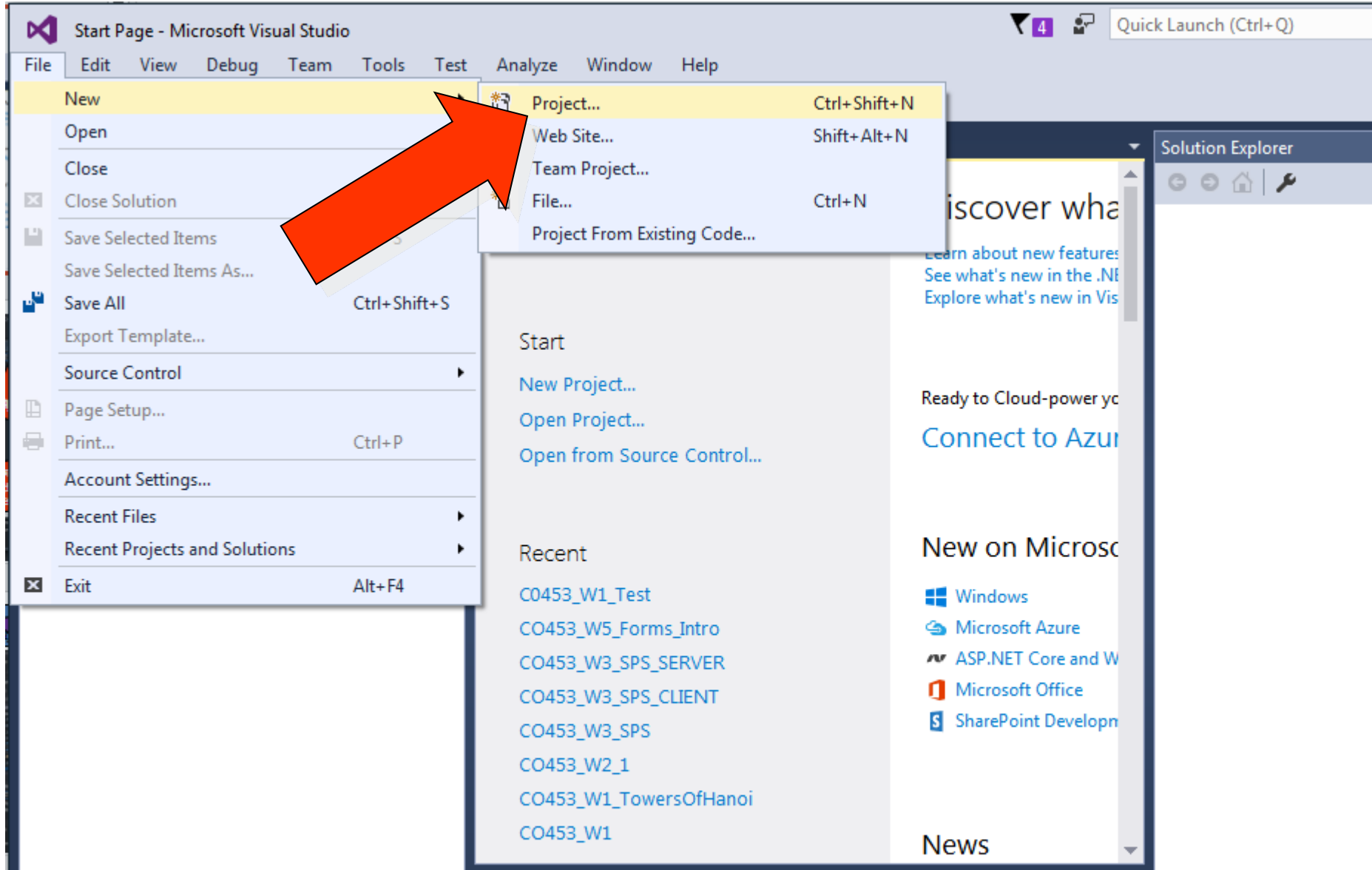
- A project can consist of several forms with objects on them
- objects have built-in properties and methods
- program code is attached to the various object methods.

The objects together make up the interface
with the user



Starting a Windows Project

Create a New Project



The screenshot shows the Microsoft Visual Studio interface. The 'File' menu is open, and the 'New' option is selected. The 'Project...' submenu is also open, and a red arrow points to the 'Project...' option. The 'Project...' submenu includes the following options:

- Project... (Ctrl+Shift+N)
- Web Site... (Shift+Alt+N)
- Team Project...
- File... (Ctrl+N)
- Project From Existing Code...

The main window displays the 'Start Page' with the following sections:

- Start**
 - [New Project...](#)
 - [Open Project...](#)
 - [Open from Source Control...](#)
- Recent**
 - CO453_W1_Test
 - CO453_W5_Forms_Intro
 - CO453_W3_SPS_SERVER
 - CO453_W3_SPS_CLIENT
 - CO453_W3_SPS
 - CO453_W2_1
 - CO453_W1_TowersOfHanoi
 - CO453_W1
- News**

The Solution Explorer on the right shows a project named 'Discover what's new in the .NET Core 2.0'.

New Project

.NET Framework 4.5.2 Sort by: Default

- Recent
- Installed
 - Visual C#**
 - Windows Forms Application
 - WPF Application
 - Application
 - Application for iOS, Android a...
 - Visual C#
 - Visual C#
 - Visual C#
 - Class Library (Portable)
 - Silverlight Application
 - Visual Basic
 - Visual F#
 - Visual C++
- Online

Click here to go online and find templates

Name: Greetings
 Location: \\buc01fs\homes\mr\n\nday01\visual studio 2015\Projects
 Solution name: Greetings

Create directory for solution
 Add to Source Control

OK Cancel

Visual C#

Select Visual C#
NOT Visual Basic

Choose a
Windows Forms
 Application

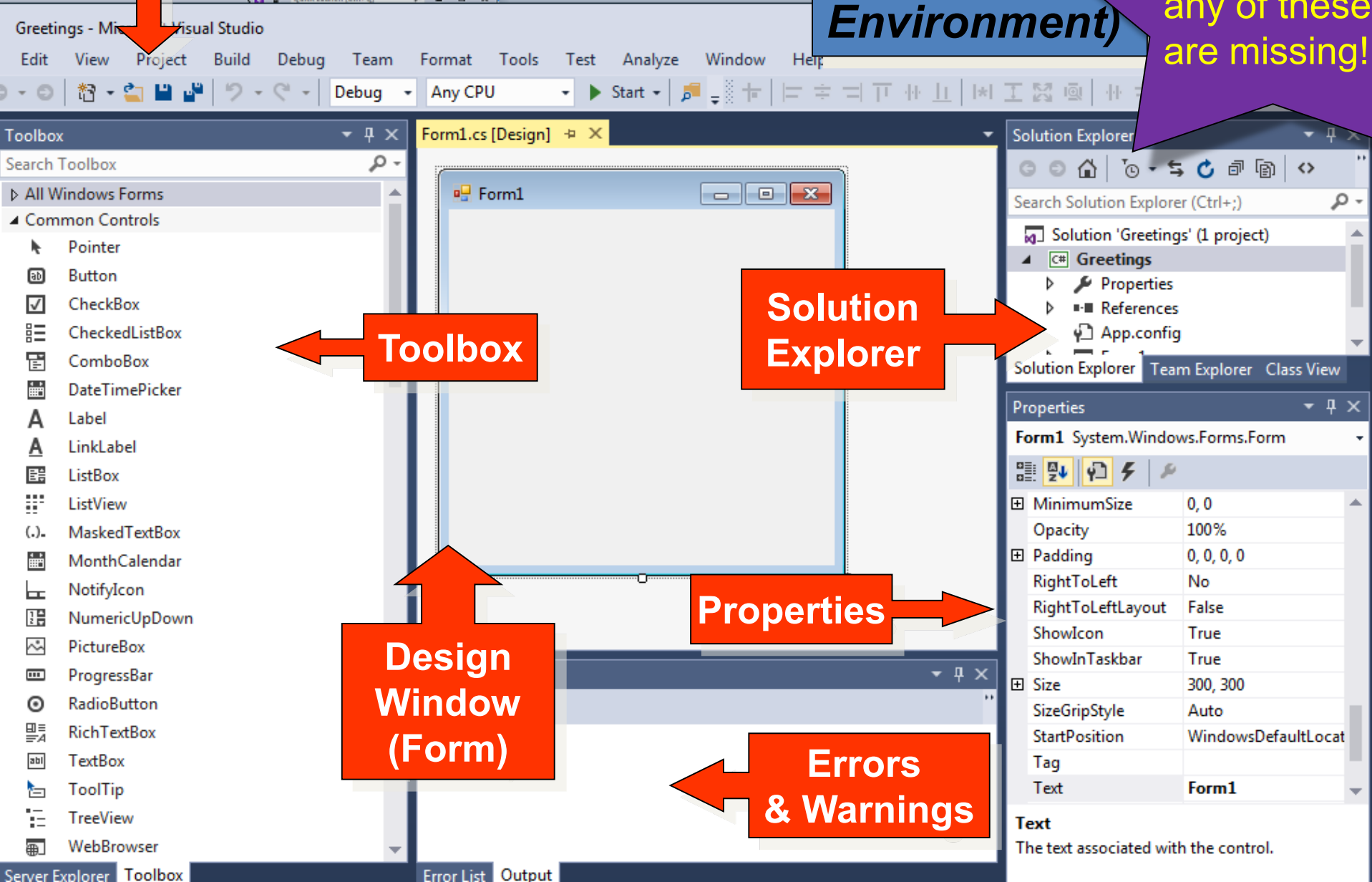
Type a Name
 and Location
 Then click **OK**

Menu

The C# .NET IDE

(Integrated Development Environment)

Use View if any of these are missing!

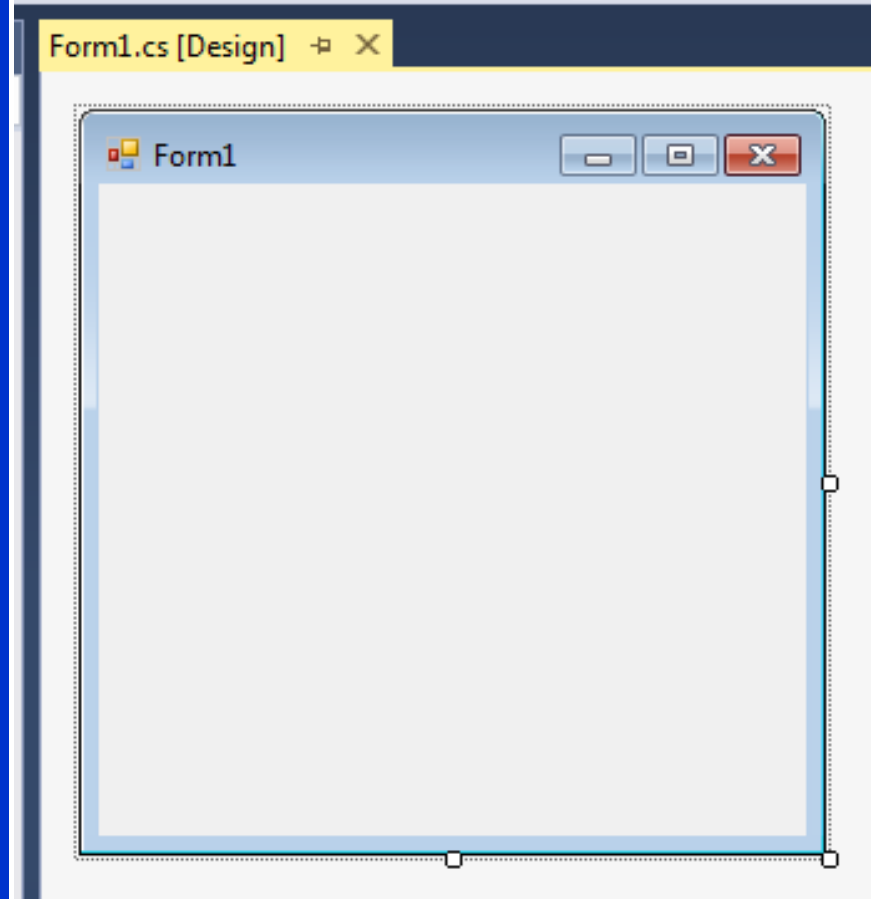


**Building
A simple
Windows project
(Name Entry and
Message Form)**

Building a Simple Project

1: The Form

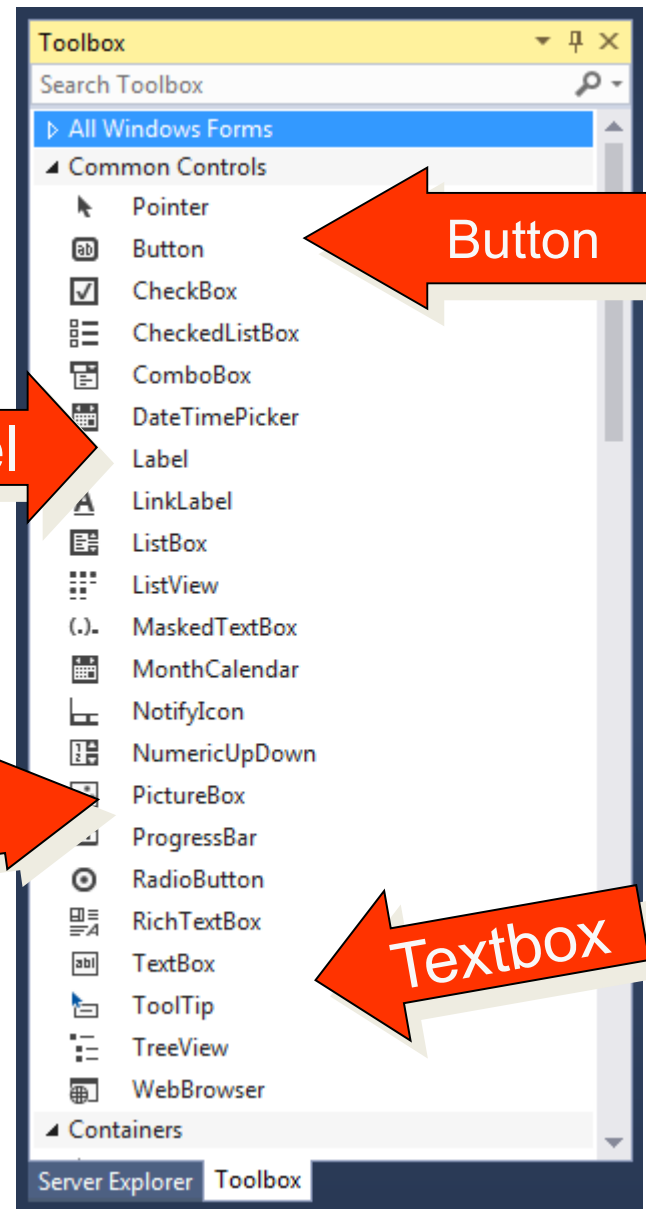
- An empty form is always the start object for your .NET C# project
- You can change the form properties
- You can add other objects (buttons, textboxes, etc) to the form
- Then change their properties
- Then you can add code to object methods to make them do something useful



Building a Project

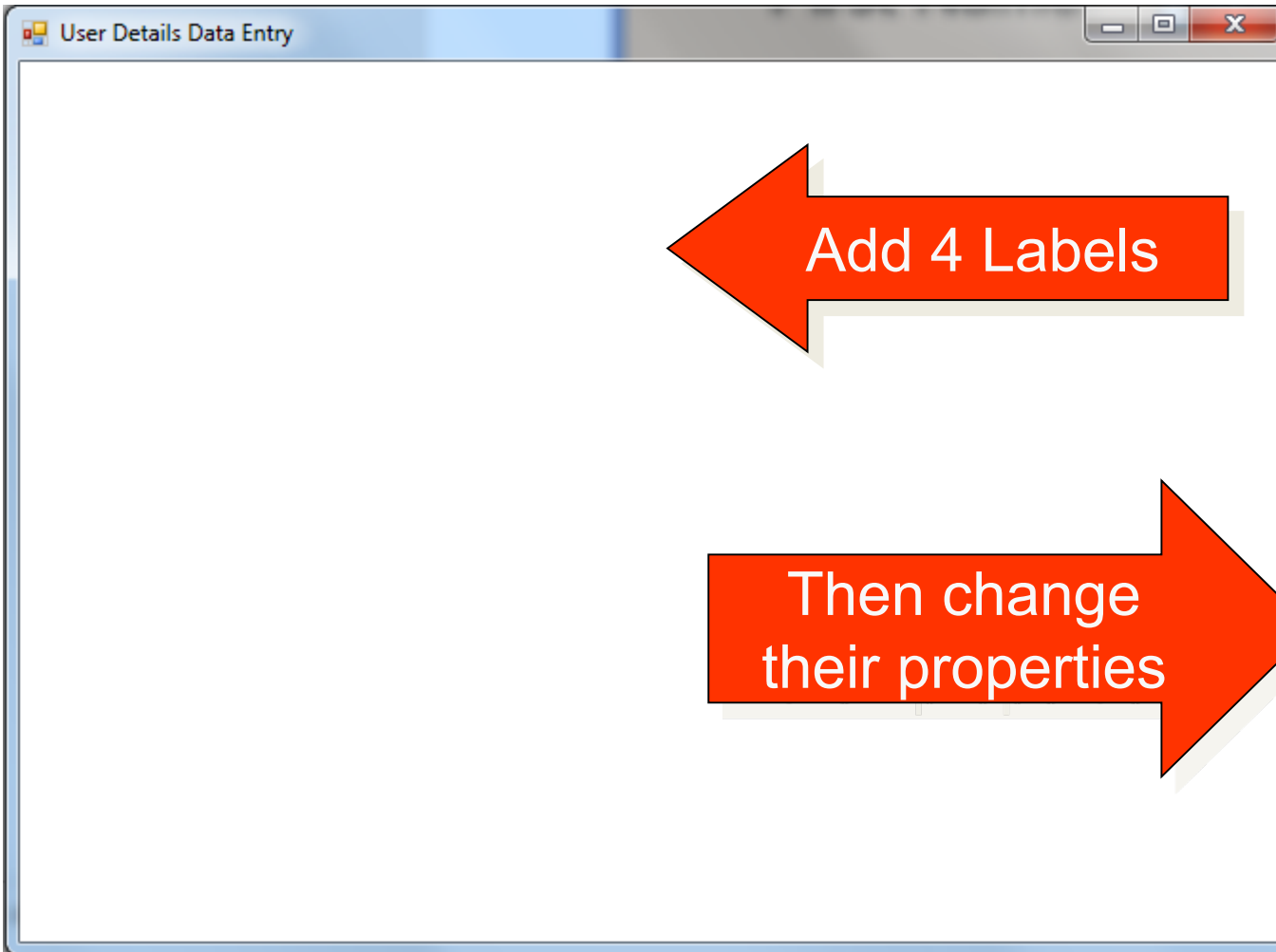
2 : Use the Toolbox

- The Toolbox provides a selection of objects (also called Controls) that can be added to a form
- Either double-click the appropriate tool on the toolbox
or
click PictureBox
tool and drag open an appropriate size box on the form



Building a Simple Project

3 : Add Some Labels



Properties

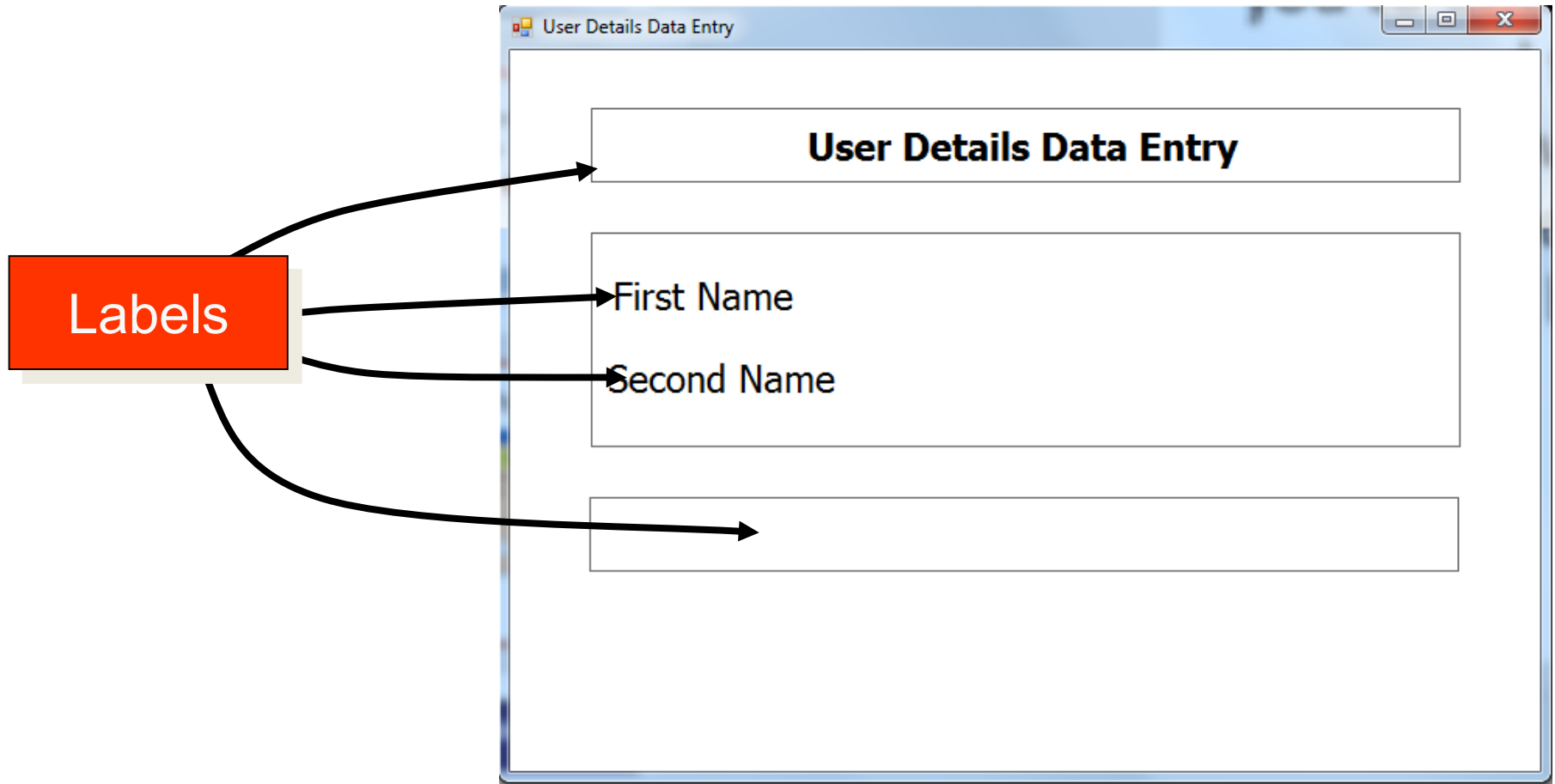
label1 System.Windows.Forms.Label

ImageIndex	<input type="checkbox"/> (none)
ImageKey	<input type="checkbox"/> (none)
ImageList	(none)
Location	22, 26
Locked	False
Margin	3, 0, 3, 0
MaximumSize	0, 0
MinimumSize	0, 0
Modifiers	Private
Padding	0, 0, 0, 0
RightToLeft	No
Size	35, 13
TabIndex	0
Tag	
Text	label1
TextAlign	TopLeft
UseCompatibleTextR	False
UseMnemonic	True
UseWaitCursor	False
Visible	True

Text
The text associated with the control.

↑ Publish ▲

Building a Simple Project 4 : Change the Properties

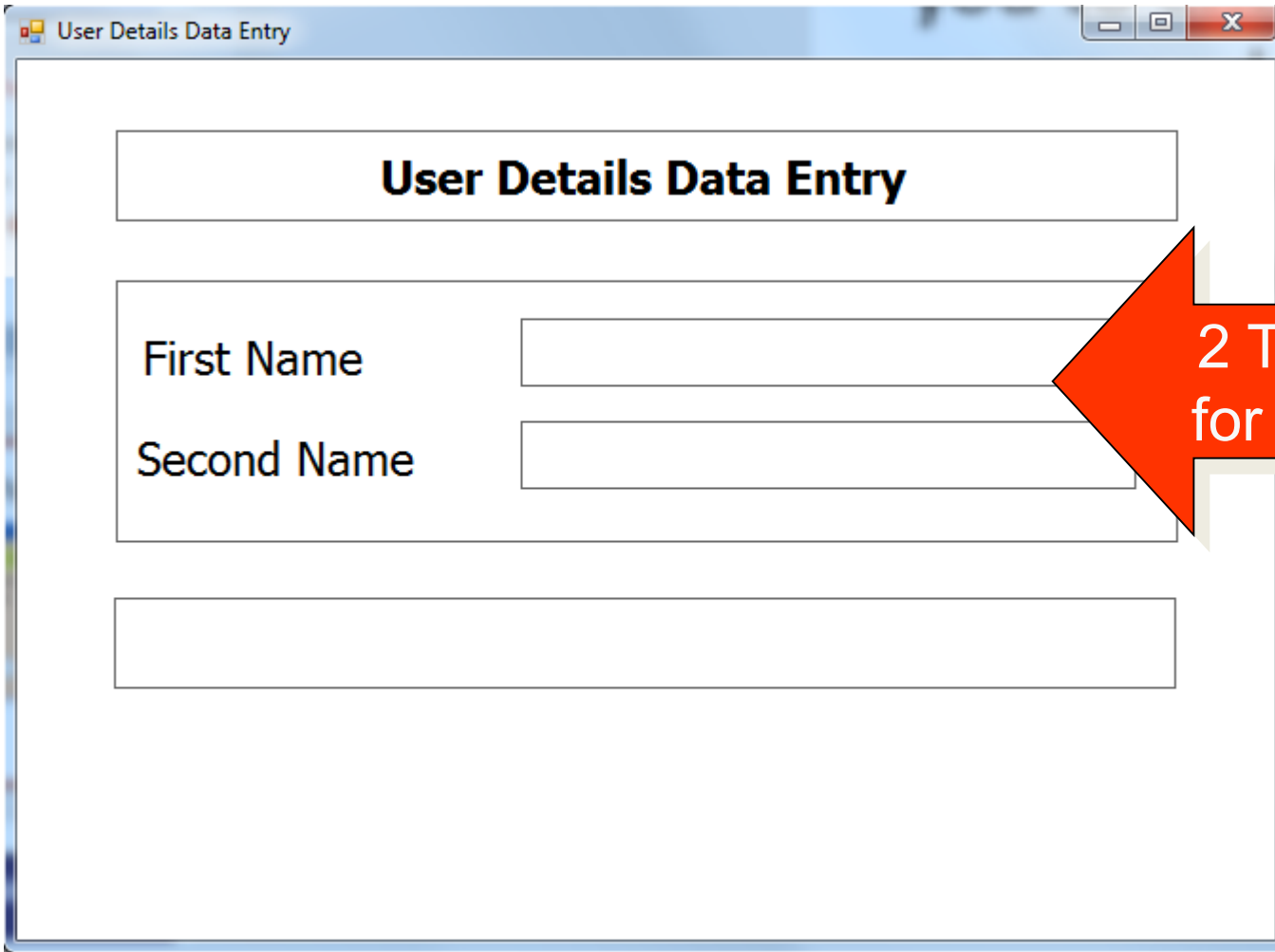


**Here we have changed the Font, Font Size, Text, BorderStyle
AutoSize and BackColor properties for the 4 labels**

Building a Simple Project

5 : Add Text Boxes

and change their properties



The screenshot shows a Windows application window titled "User Details Data Entry". The window contains a form with the following elements:

- A title bar with the text "User Details Data Entry" and standard Windows window controls (minimize, maximize, close).
- A large rectangular box containing the text "User Details Data Entry" in bold black font.
- Two rows of text labels with corresponding text boxes:
 - "First Name" followed by a text box.
 - "Second Name" followed by a text box.
- A large empty rectangular box at the bottom of the form.

2 Text Boxes
for data entry

Building a Simple Project

6 : Add Buttons

And change their properties

The screenshot shows a Java Swing window titled "User Details Data Entry". The window contains a title bar with standard minimize, maximize, and close buttons. The main content area has a title "User Details Data Entry" centered at the top. Below the title, there are two text input fields. The first field is labeled "First Name" and the second is labeled "Second Name". At the bottom of the window, there are three buttons: "Message", "Clear", and "Exit". The "Clear" button is highlighted with a double border.

3 Buttons

Building a Simple Project

7 : Add Code to Objects

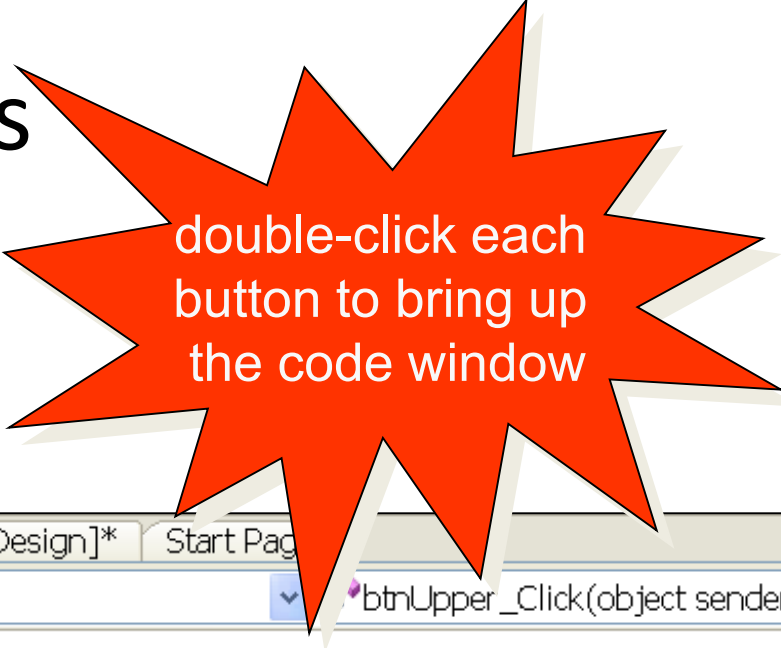
Double-Click objects to bring up the Code Window

Objects have built-in templates for methods e.g. `btnQuit_Click()`

Add code to the appropriate method template

```
Form1.cs [Design]*   Form1.Designer.cs   Form1.cs*  X
C0453_W1_Test
7   using System.Text;
8   using System.Threading.Tasks;
9   using System.Windows.Forms;
10
11  namespace C0453_W1_Test
12  {
13      3 references
14      public partial class Form1 : Form
15      {
16          1 reference
17          public Form1()
18          {
19              InitializeComponent();
20          }
21          1 reference
22          private void btnMessage_Click(object sender, EventArgs e)
23          {
24              // add new code here
25          }
26      }
27
28
29
30
```


Adding Code to Buttons

A screenshot of a Windows application window titled "User Details Data Entry". The window contains a title bar, a header area with the text "User Details Data Entry", two text input fields labeled "First Name" and "Second Name", and three buttons at the bottom labeled "Message", "Clear", and "Exit".

```
Form1.cs*  Form1.cs [Design]*  Start Page
DataEntry.frmDataEntry  btnUpper_Click(object sender, EventArgs e)

private void btnMessage_Click(object sender, EventArgs e)
{
    lblMessage.Text = "Hello " + txtFName.Text + " " + txtSName.Text;
}

private void btnQuit_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void btnClear_Click(object sender, EventArgs e)
{
    lblMessage.Text = "";
}
```

Building a Simple Project

8 : Run it

The screenshot shows a Java Swing window titled "User Details Data Entry". The window contains a form with two input fields: "First Name" with the text "Nick" and "Second Name" with the text "Day". Below the form, a message box displays "Hi Nick Day!". At the bottom of the window, there are three buttons: "Message", "Clear", and "Exit".

Enter text here

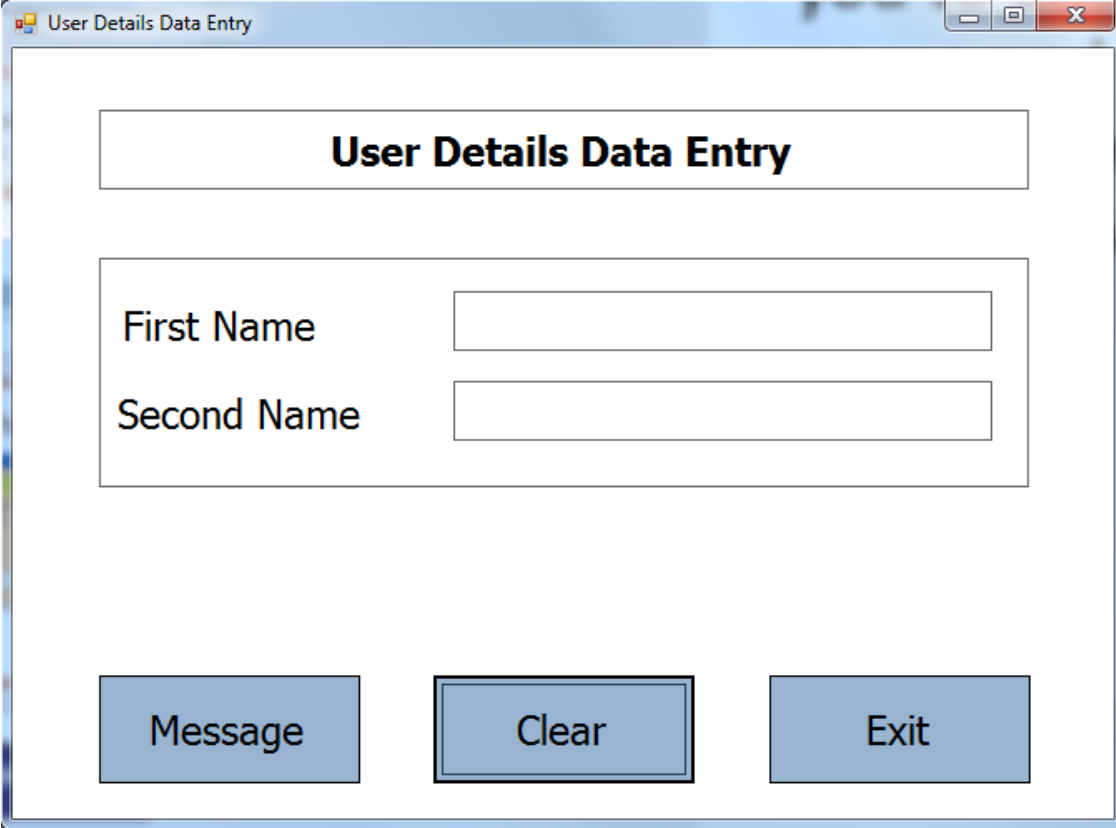
click here to clear
the message box

Click to Finish

Then click here
to see message

Activity

- Try and produce a similar form. The form should allow you to enter your name and then output it when clicking the 'message' button (Task 1.4)

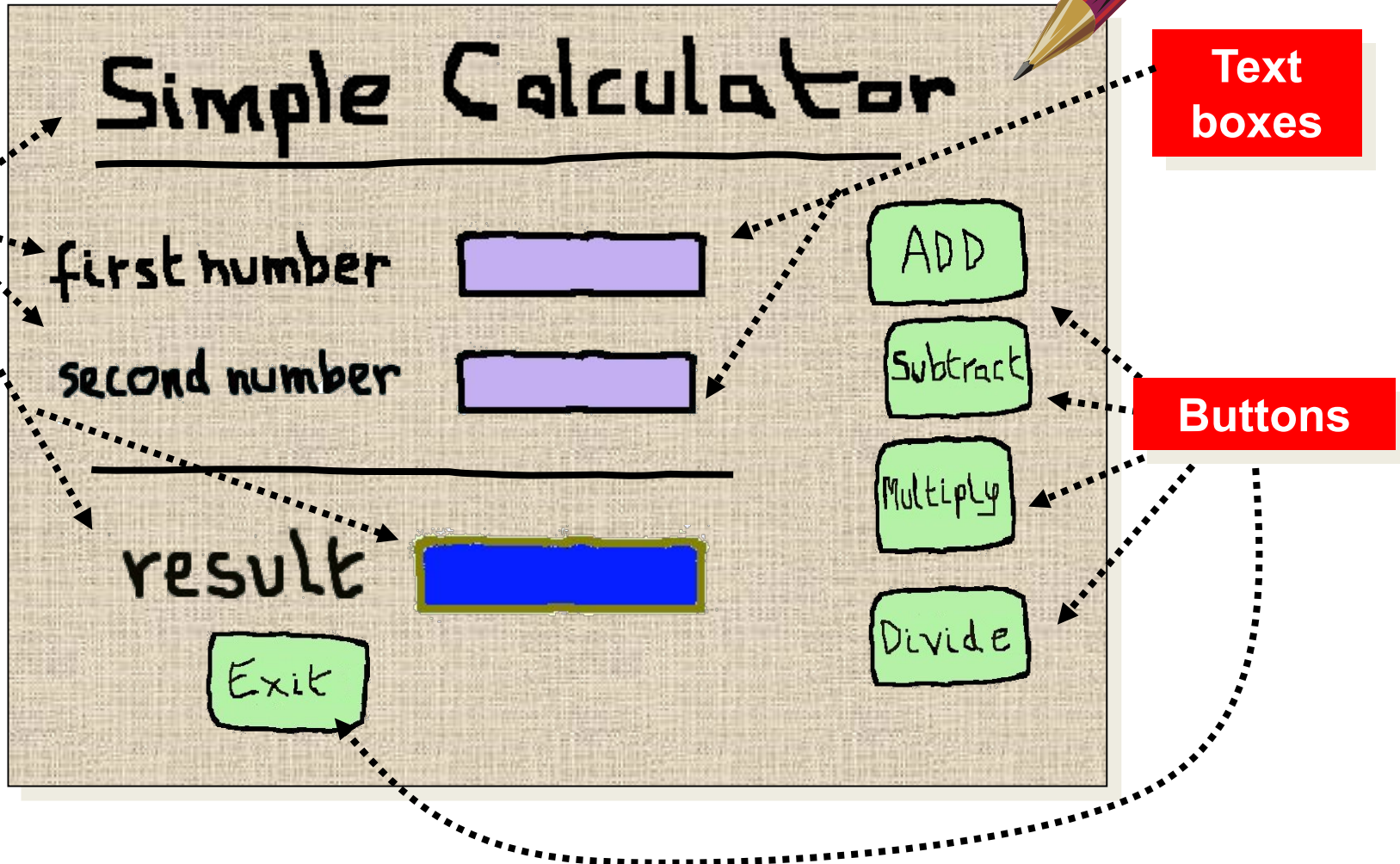


The screenshot shows a Windows application window titled "User Details Data Entry". The window contains a form with the following elements:

- A title bar with the text "User Details Data Entry" and standard Windows window controls (minimize, maximize, close).
- A large rectangular box containing the text "User Details Data Entry" in bold.
- Two input fields: "First Name" and "Second Name", each with a corresponding text box to its right.
- Three buttons at the bottom: "Message", "Clear", and "Exit".

**Designing
a
C# Windows project
(A simple calculator)**

1: Make a Sketch



2a: List Objects Needed and give them names

Objects Needed

- 1 form (frmCalc)
- 5 labels
- 2 text boxes
- 5 buttons

Label Names

lblTitle
lblFirstNum
lblSecNum
lblResult
lblResultLabel

Button Names

btnAdd
btnSub
btnMult
btnDiv
btnQuit

Text Box Names

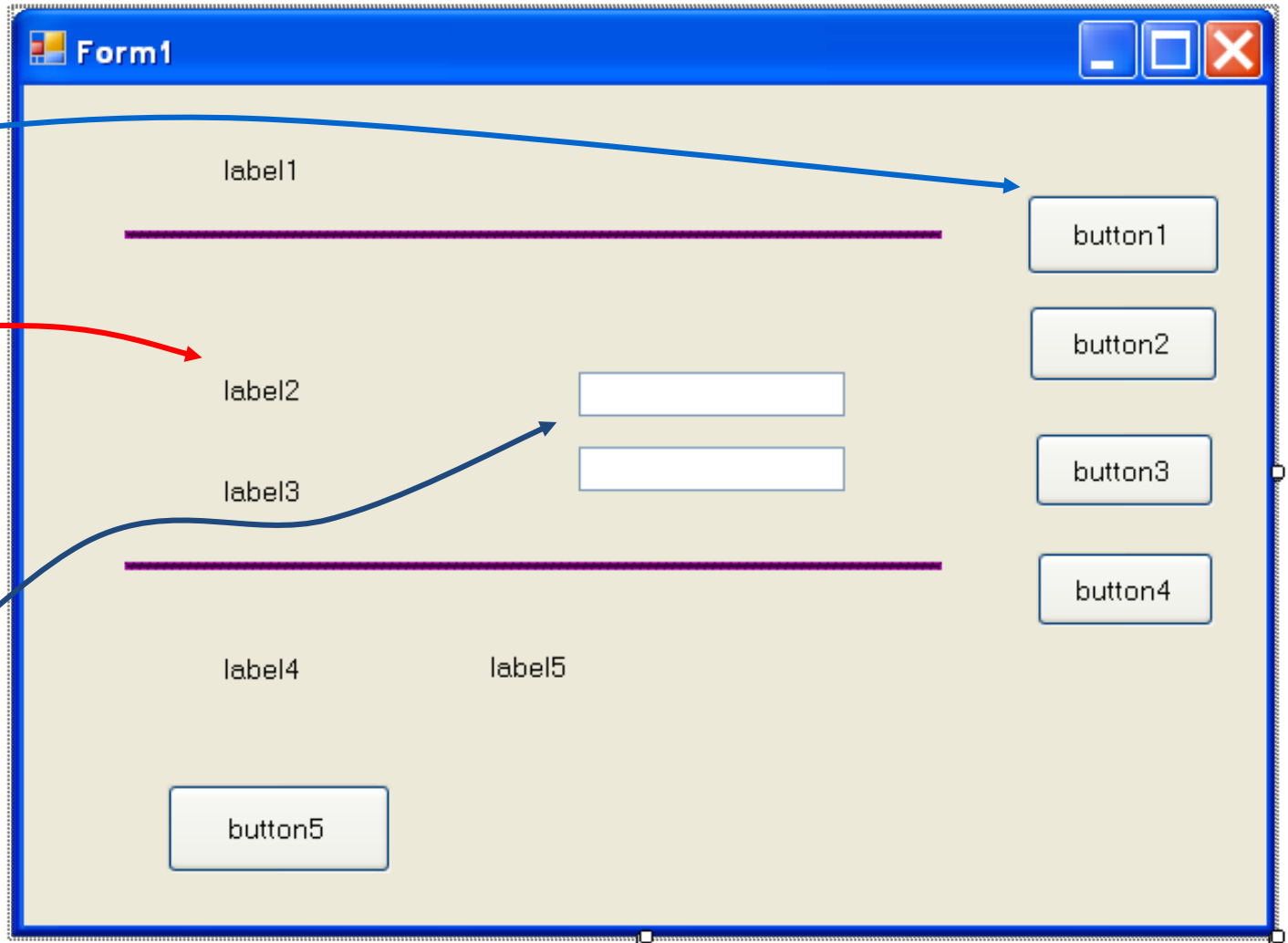
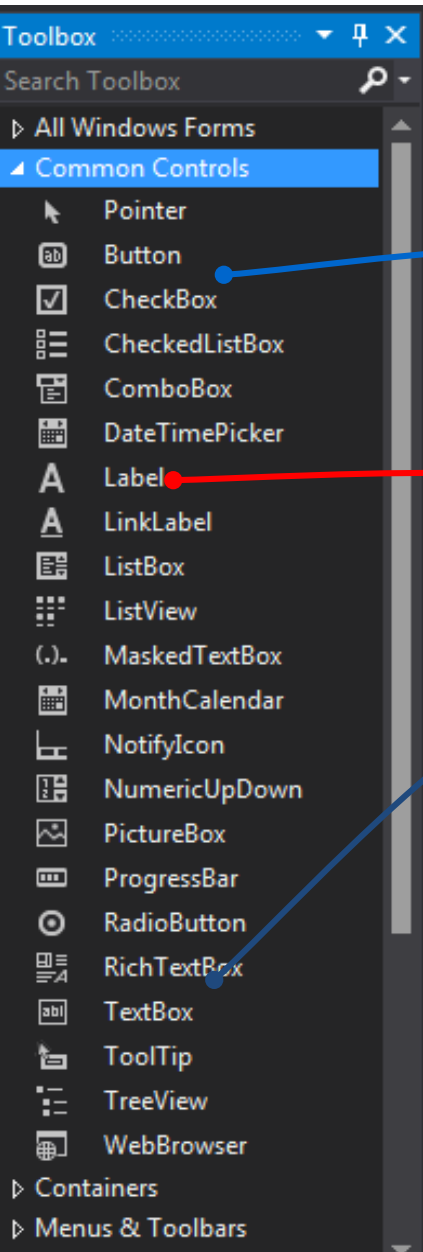
txtFirstNum
txtSecNum

2b: TOE Chart

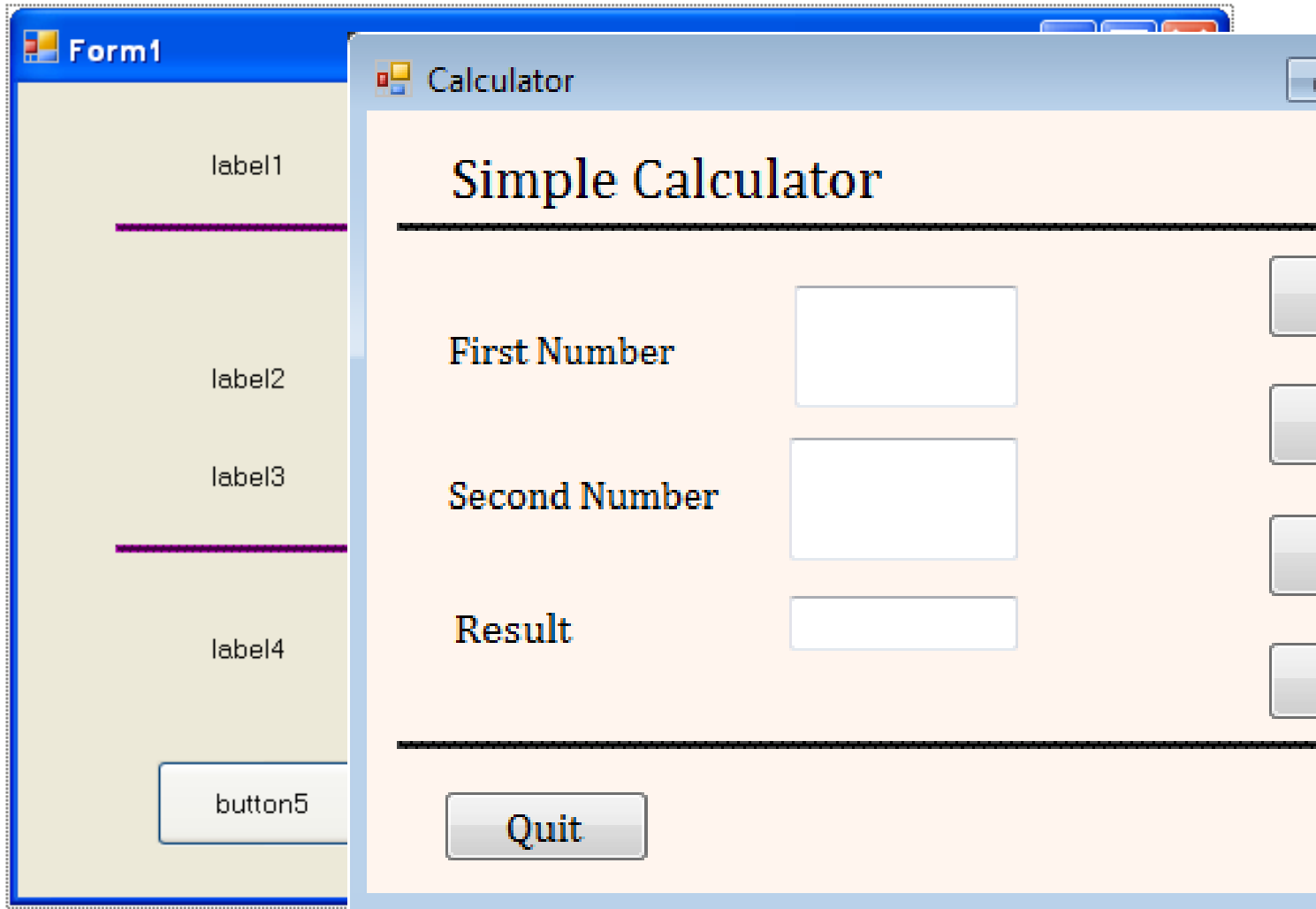
	Task	Object	Event
1	Display Interface	frmCalc	
2	Enter first number	txtFirstNum	
3	Enter second number	txtSecNum	
4	Perform addition	btnAdd	Click
5	Perform subtraction	btnSub	Click
6	Perform multiplication	btnMult	Click
7	Perform division	btnDiv	Click
8	Display Result	lblResult	
9	Quit Program	btnQuit	Click

Other objects on the form have no active role (decoration only)

3: Put Objects on Form



4: Change Object Properties



5: Add Code to Buttons

The image shows a Visual Studio window with the following tabs: Form1.cs*, Form1.cs [Design]*, and Start Page. The active window is titled "Calculator.Form1" and shows the following C# code:

```
private void btnAdd_Click(object sender, EventArgs e)
{
}

private void btnSub_Click(object sender, EventArgs e)
{
}

private void btnMult_Click(object sender, EventArgs e)
{
}

private void btnDiv_Click(object sender, EventArgs e)
{
}

private void btnQuit_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

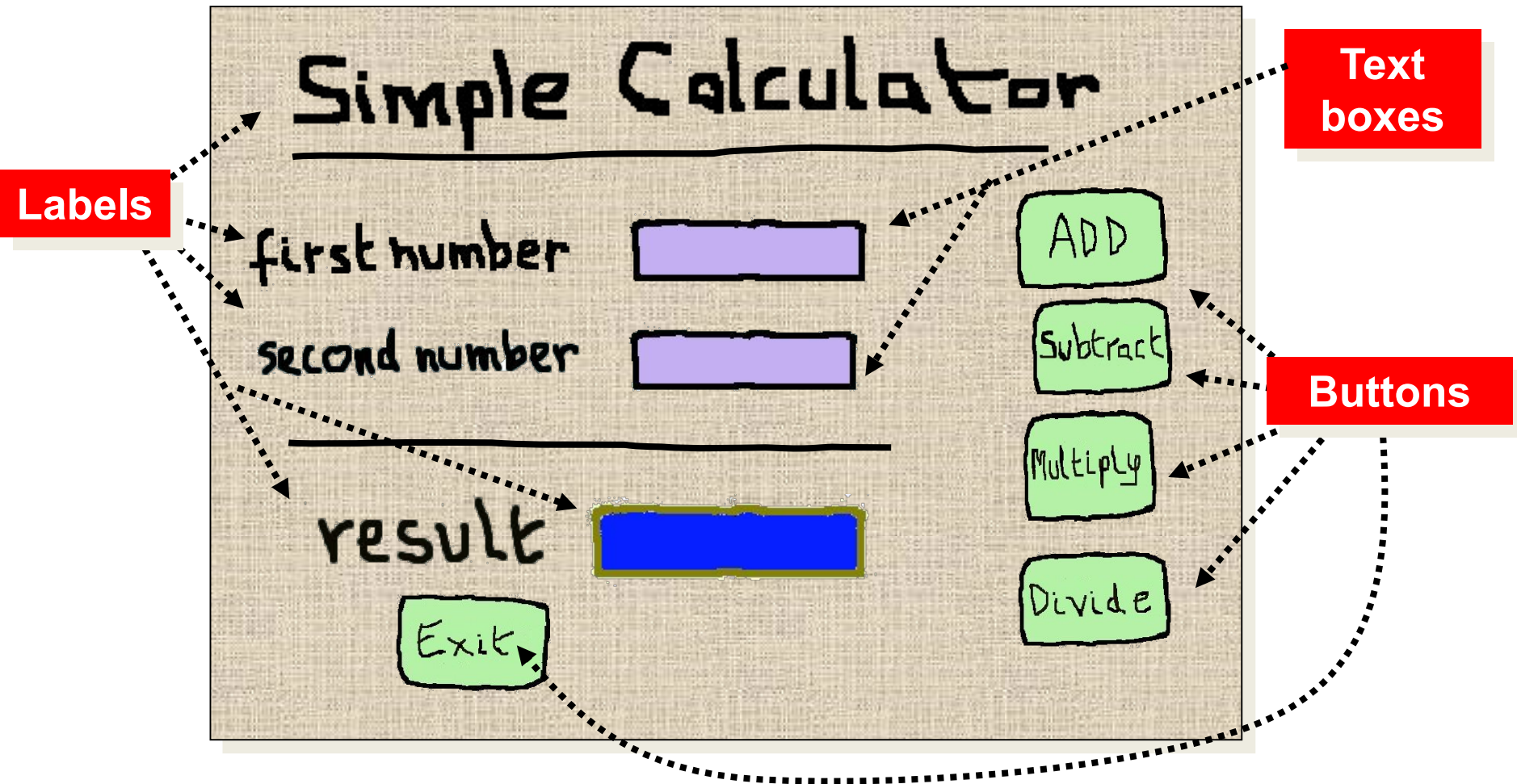
On the left side, there are five buttons: "Add", "Subtract", "Multiply", "Divide", and "Quit". The "Add", "Subtract", "Multiply", and "Divide" buttons are grouped together in a yellow box. The "Quit" button is in a separate yellow box. Red arrows point from each button to its corresponding code block in the C# code.

The Add button Code

```
private void btnAdd_Click(object sender, EventArgs e)
{
    // define 3 variables for decimal numbers
    double n1, n2, answer;
    // convert strings in text boxes to numbers
    n1 = Convert.ToDouble(txtFirstNum.Text);
    n2 = Convert.ToDouble(txtSecNum.Text);
    answer = n1 + n2;
    // convert answer to a string and put into result label
    lblResult.Text = answer.ToString();
}
```

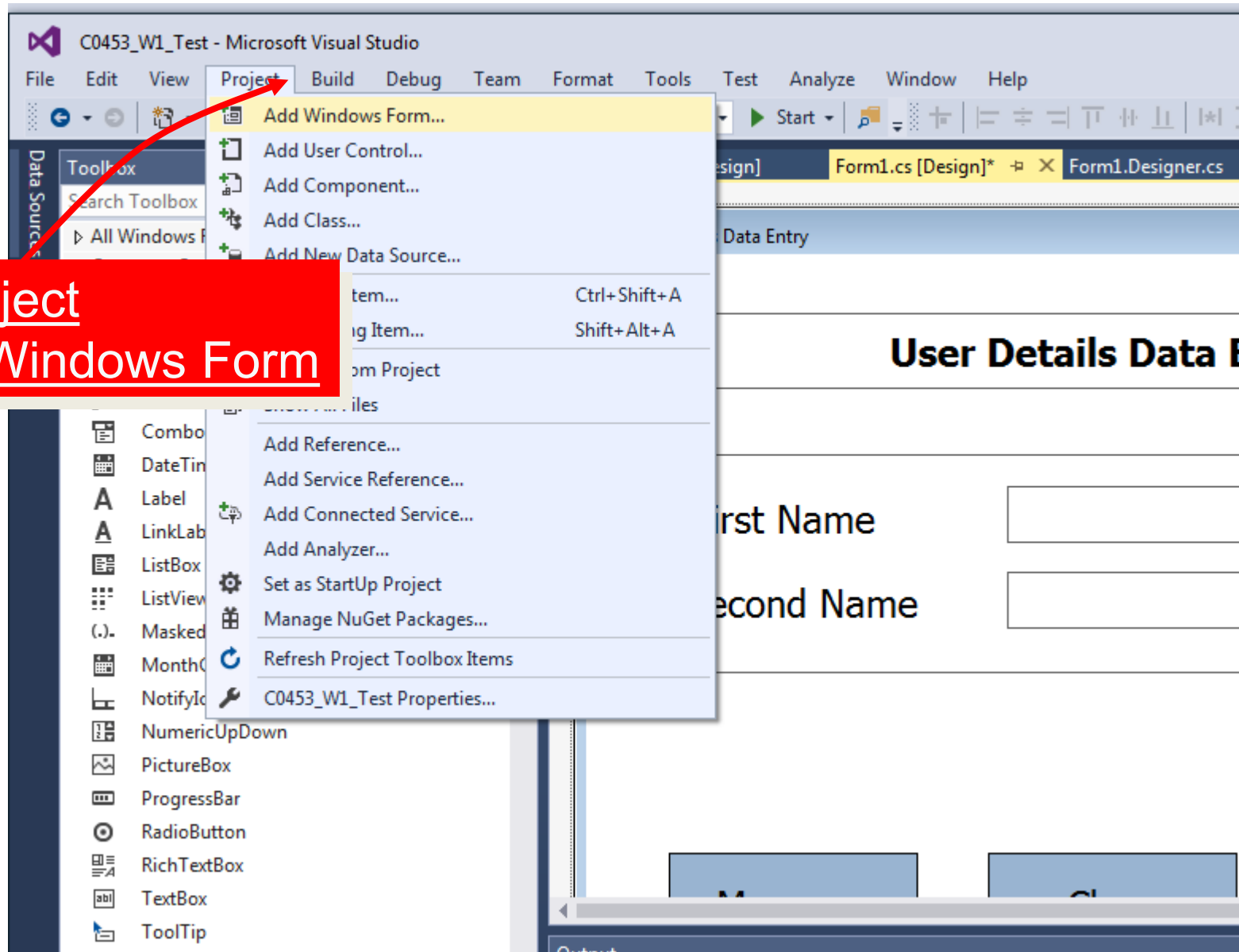
Activity

Replicate the form design and functionality below:



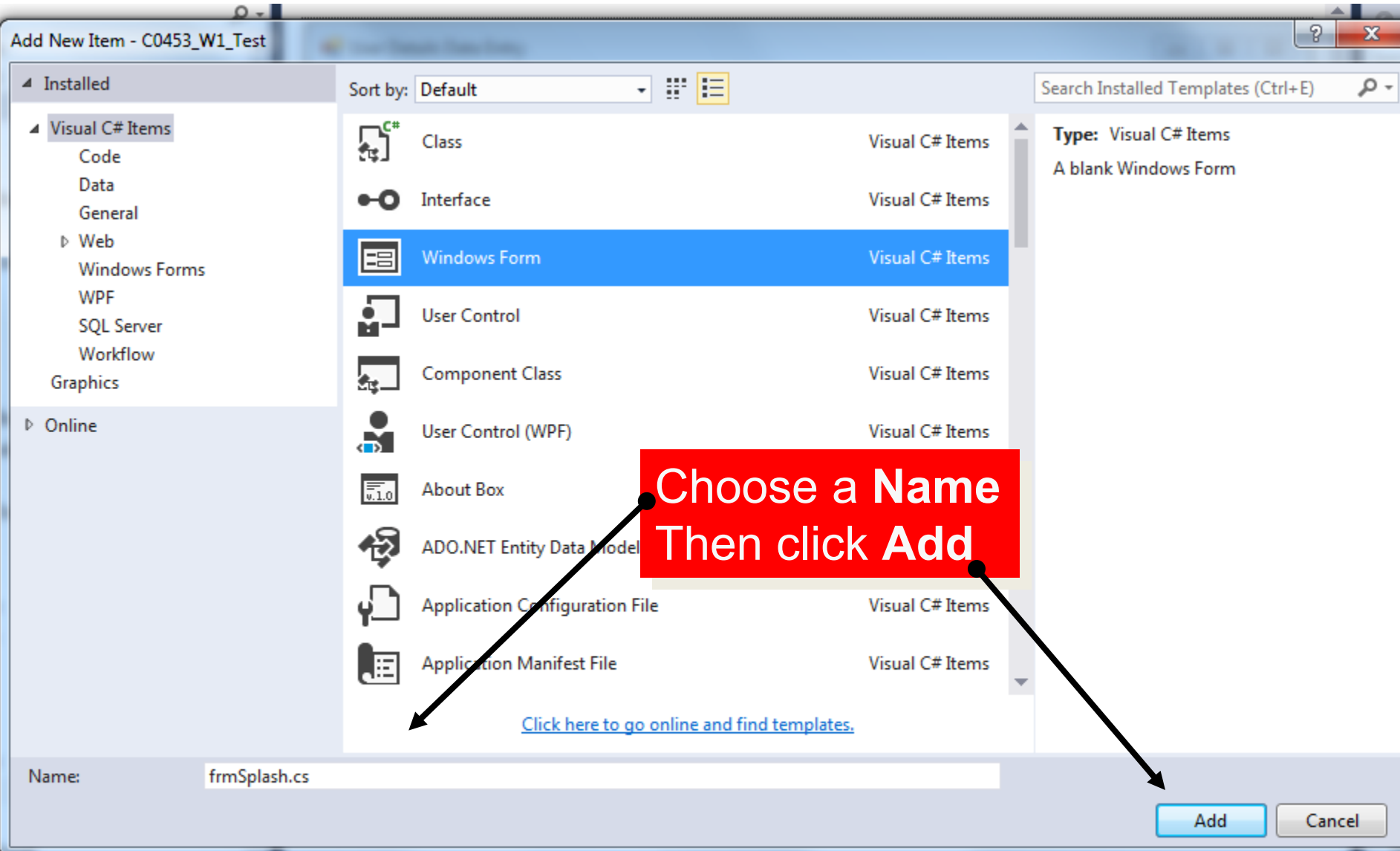
**Adding
a new form
to the project
(a Splash screen)**

1: Choose Add Windows Form



Select Project
then Add Windows Form

2: Name the new Form



3: Design new startup Screen



Add a label and set the **Text** and **Font** properties

Form Properties

FormBorderStyle = *none*

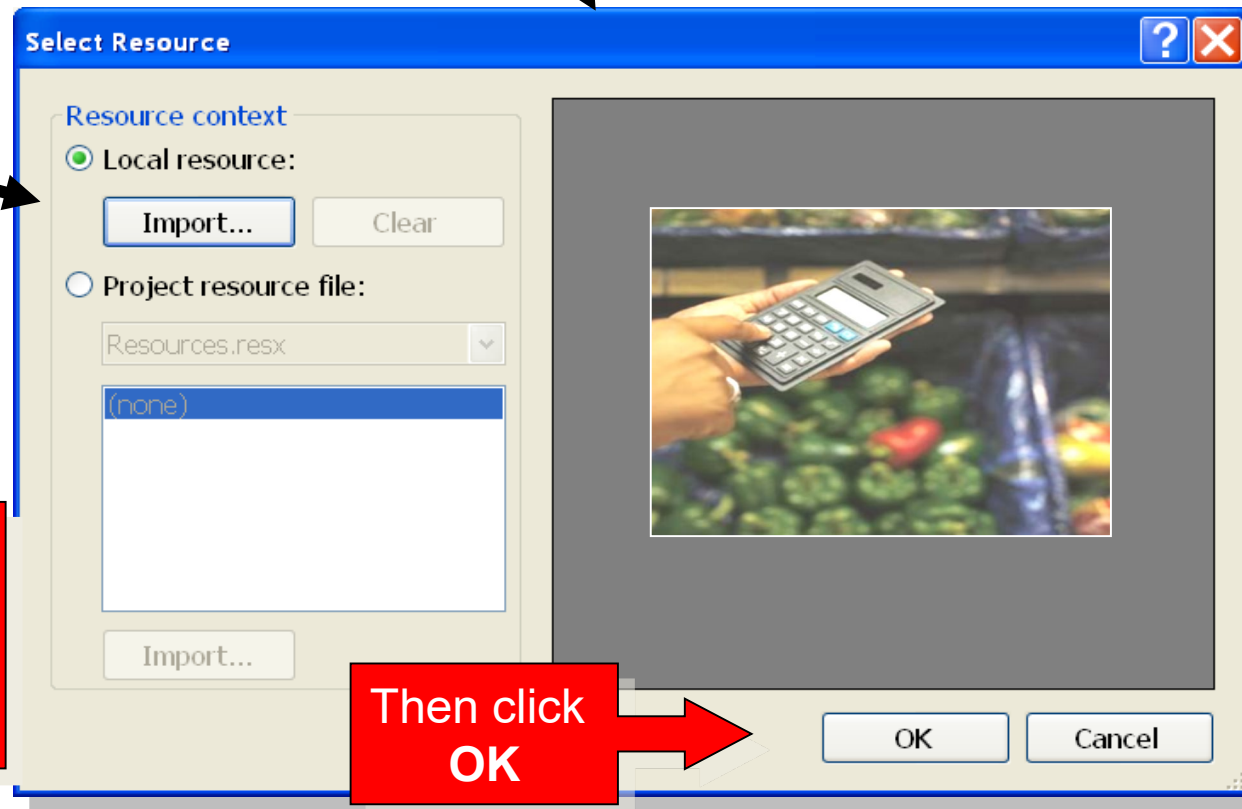
StartPosition = *CenterScreen*

BackgroundImage = *< see next slide >*

4: Adding a BackgroundImage



- Click Import ..
- find an image file

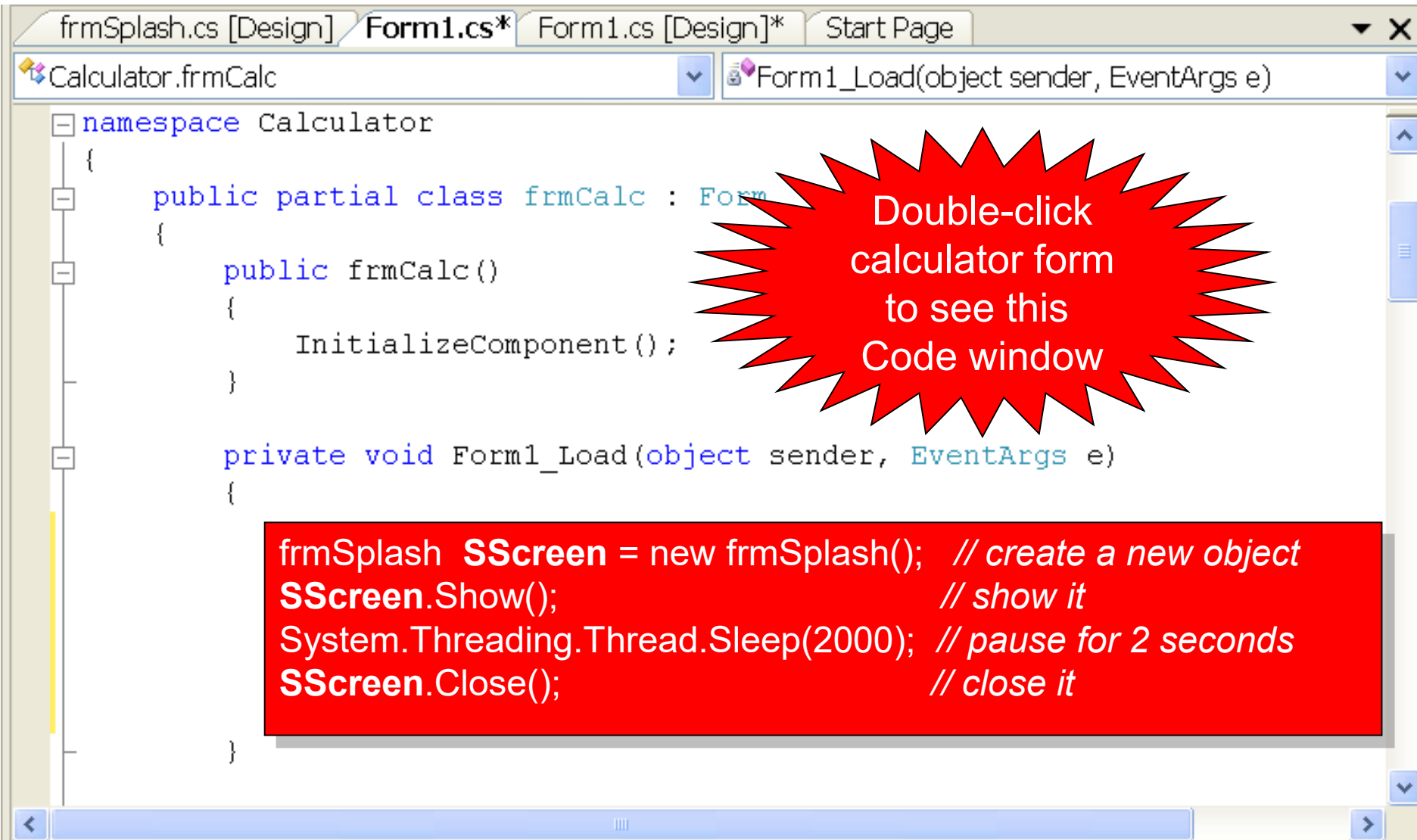


Note:
BackgroundImageLayout

- set this form property to:
Stretch
(or Tile, Center or Zoom)

5a: Coding the Calculator Form

Use Form1's Load event to show the Splash screen first



Calculator.frmCalc Form1_Load(object sender, EventArgs e)

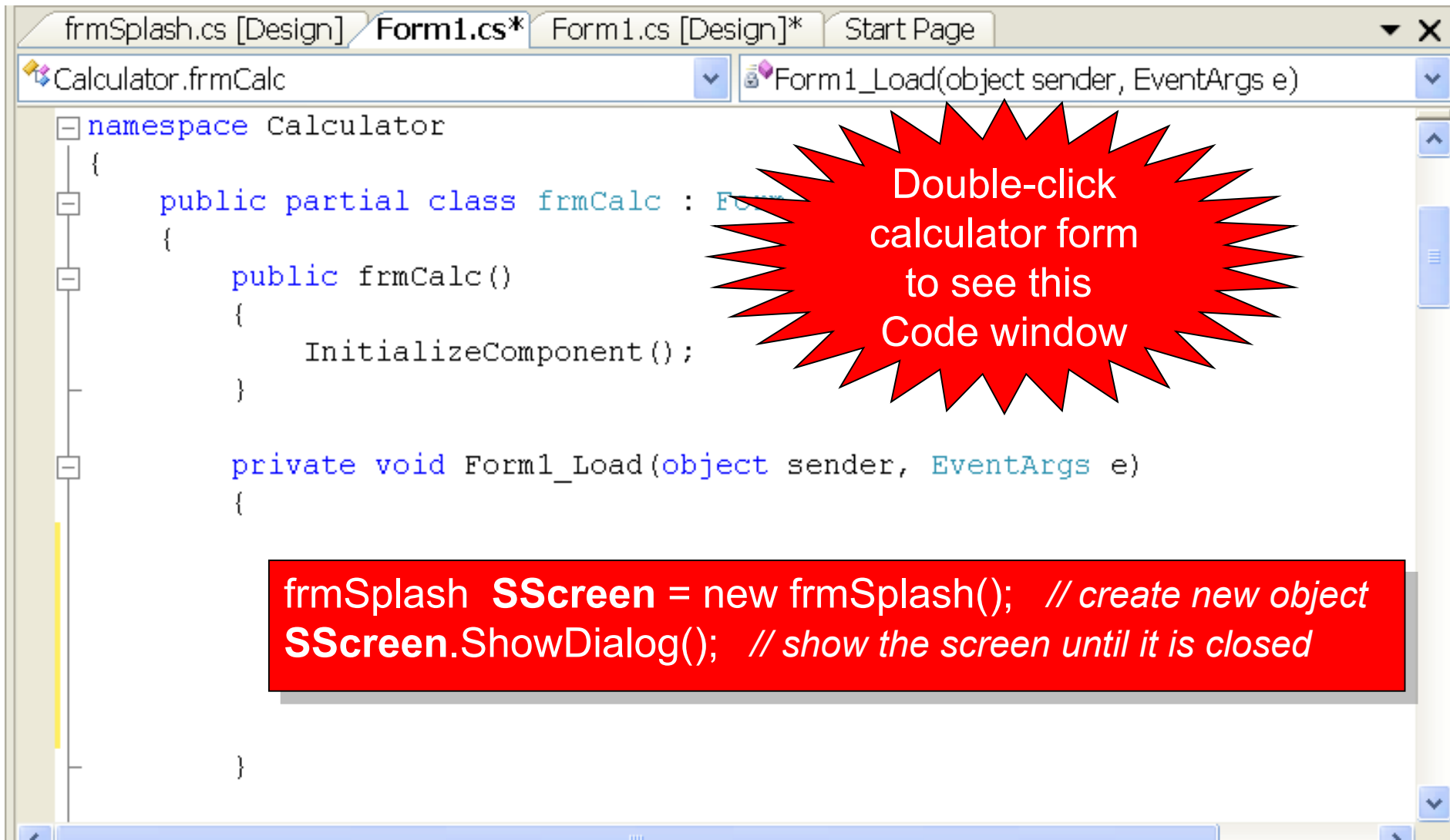
```
namespace Calculator
{
    public partial class frmCalc : Form
    {
        public frmCalc()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            frmSplash SScreen = new frmSplash(); // create a new object
            SScreen.Show(); // show it
            System.Threading.Thread.Sleep(2000); // pause for 2 seconds
            SScreen.Close(); // close it
        }
    }
}
```

Double-click calculator form to see this Code window

5b: Alternative Coding

We want the **Splash** screen to show until we click it!



Calculator.frmCalc Form1_Load(object sender, EventArgs e)

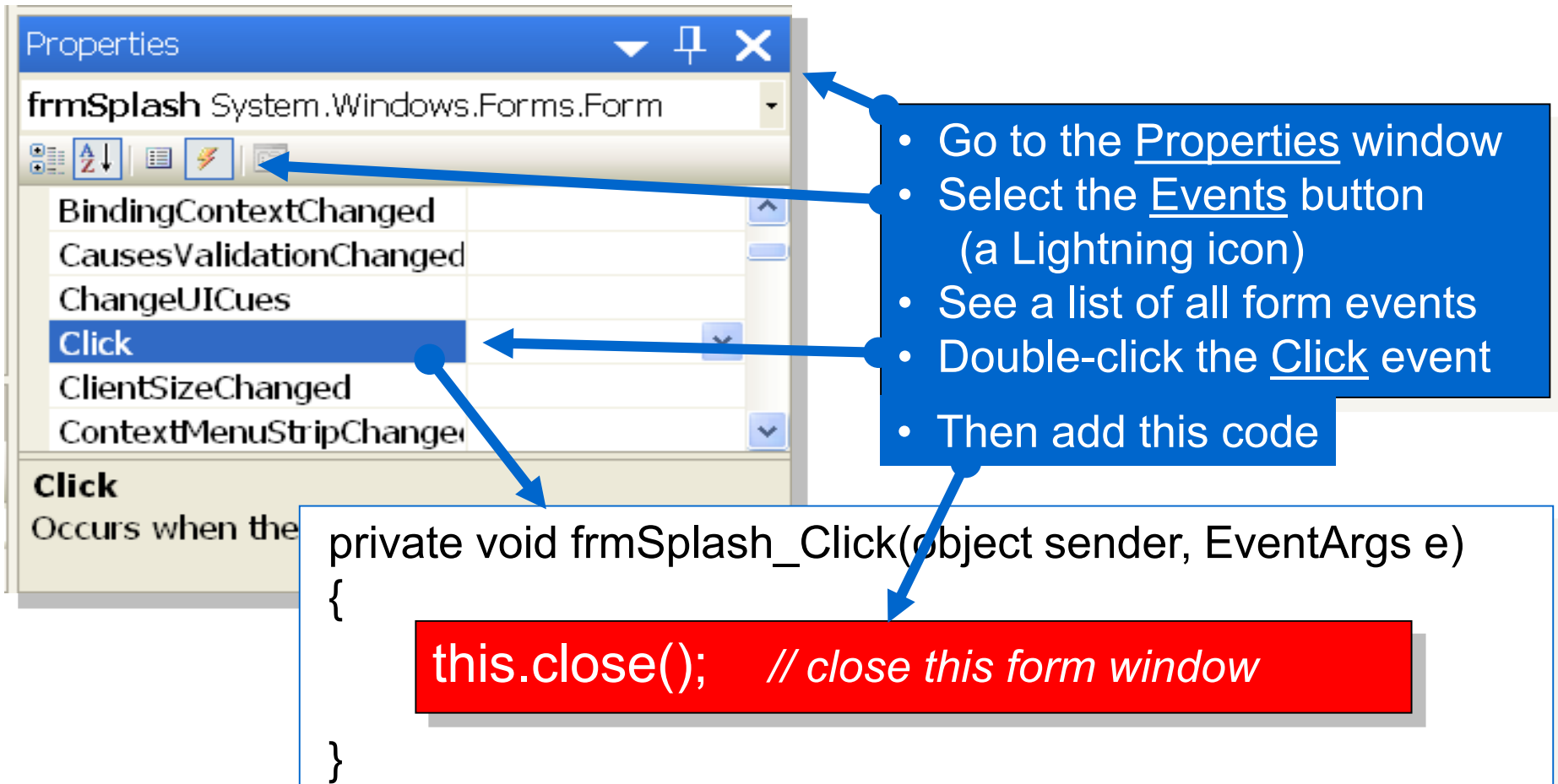
```
namespace Calculator
{
    public partial class frmCalc : Form
    {
        public frmCalc()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            frmSplash SScreen = new frmSplash(); // create new object
            SScreen.ShowDialog(); // show the screen until it is closed
        }
    }
}
```

Double-click calculator form to see this Code window

5b: Continued

- Add code to the frmSplash Click method to close the form. But :
- The Load event is a form's default method!!
 - To get to its Click method do this:



Properties window for frmSplash (System.Windows.Forms.Form). The Events view is selected, showing a list of events. The Click event is highlighted. A blue callout box provides instructions:

- Go to the Properties window
- Select the Events button (a Lightning icon)
- See a list of all form events
- Double-click the Click event
- Then add this code

The code block shows the implementation of the Click event handler:

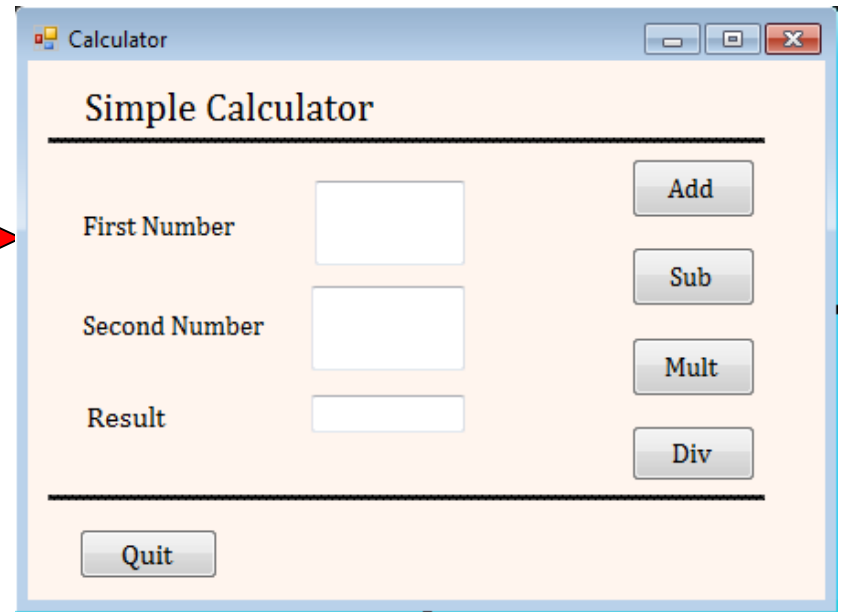
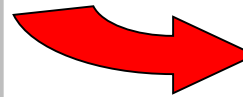
```
private void frmSplash_Click(object sender, EventArgs e)
{
    this.close(); // close this form window
}
```

Final Result

The Splash screen shows first. Click it to use the Calculator.



click

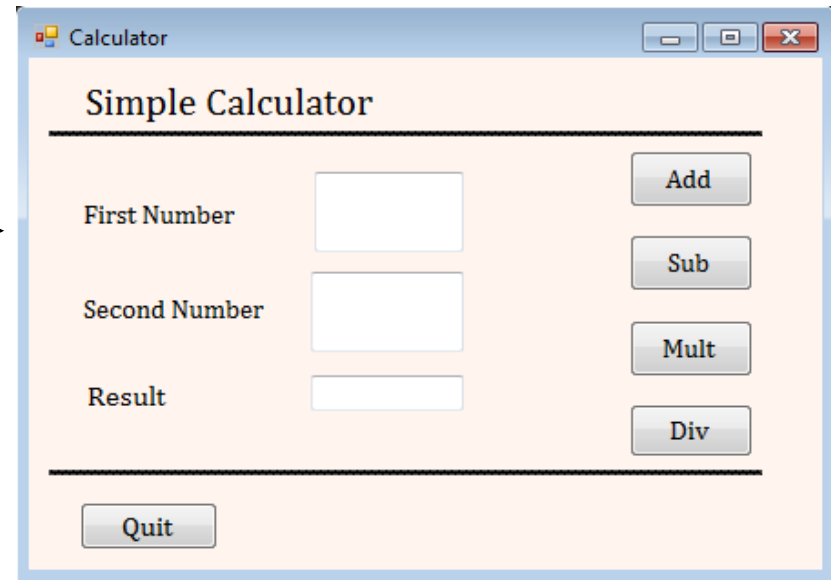
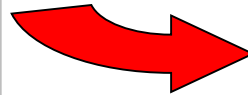


Activity

Add a splash screen to your calculator (Task 1.6)



click

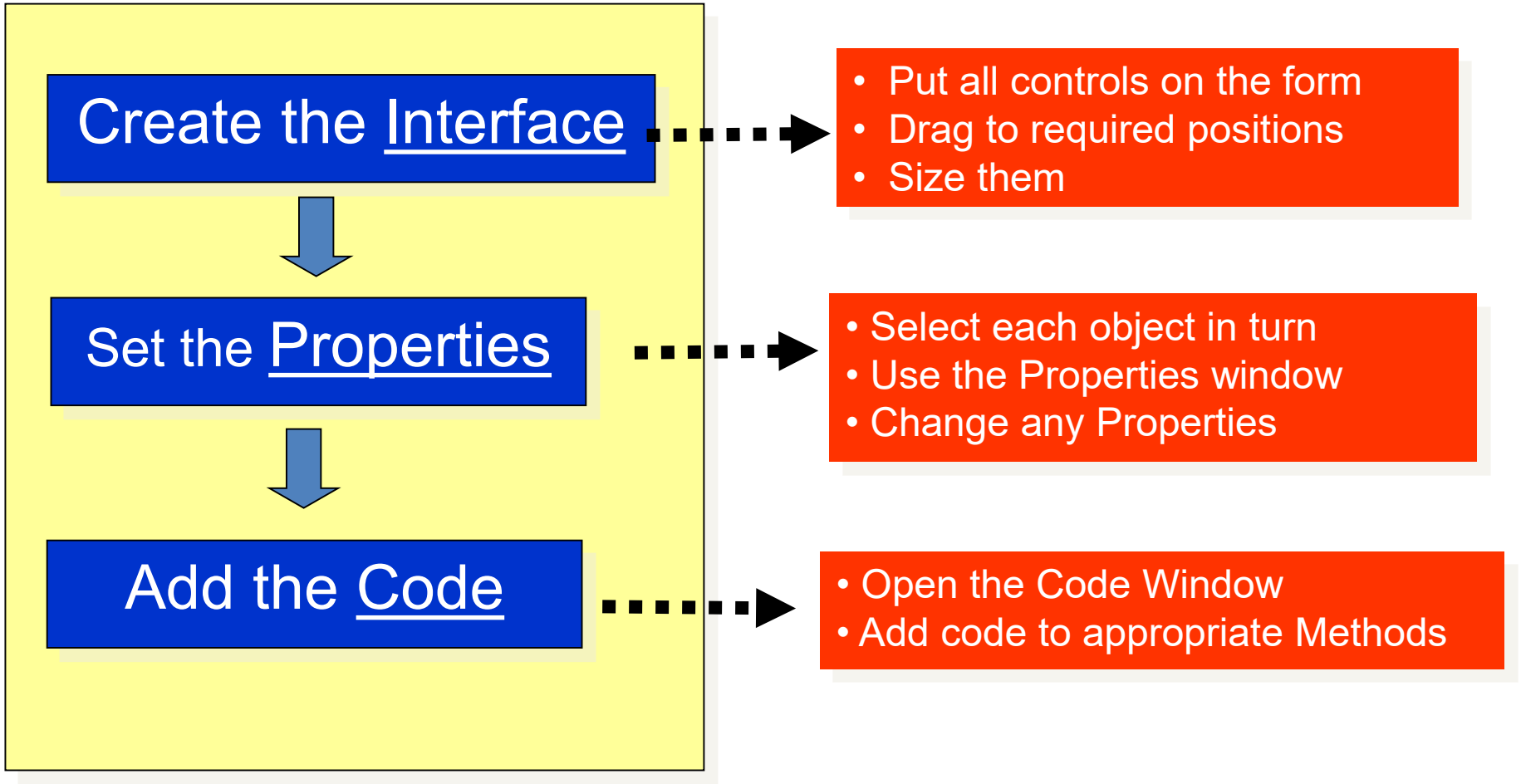


The Last Slide



Extra Reading

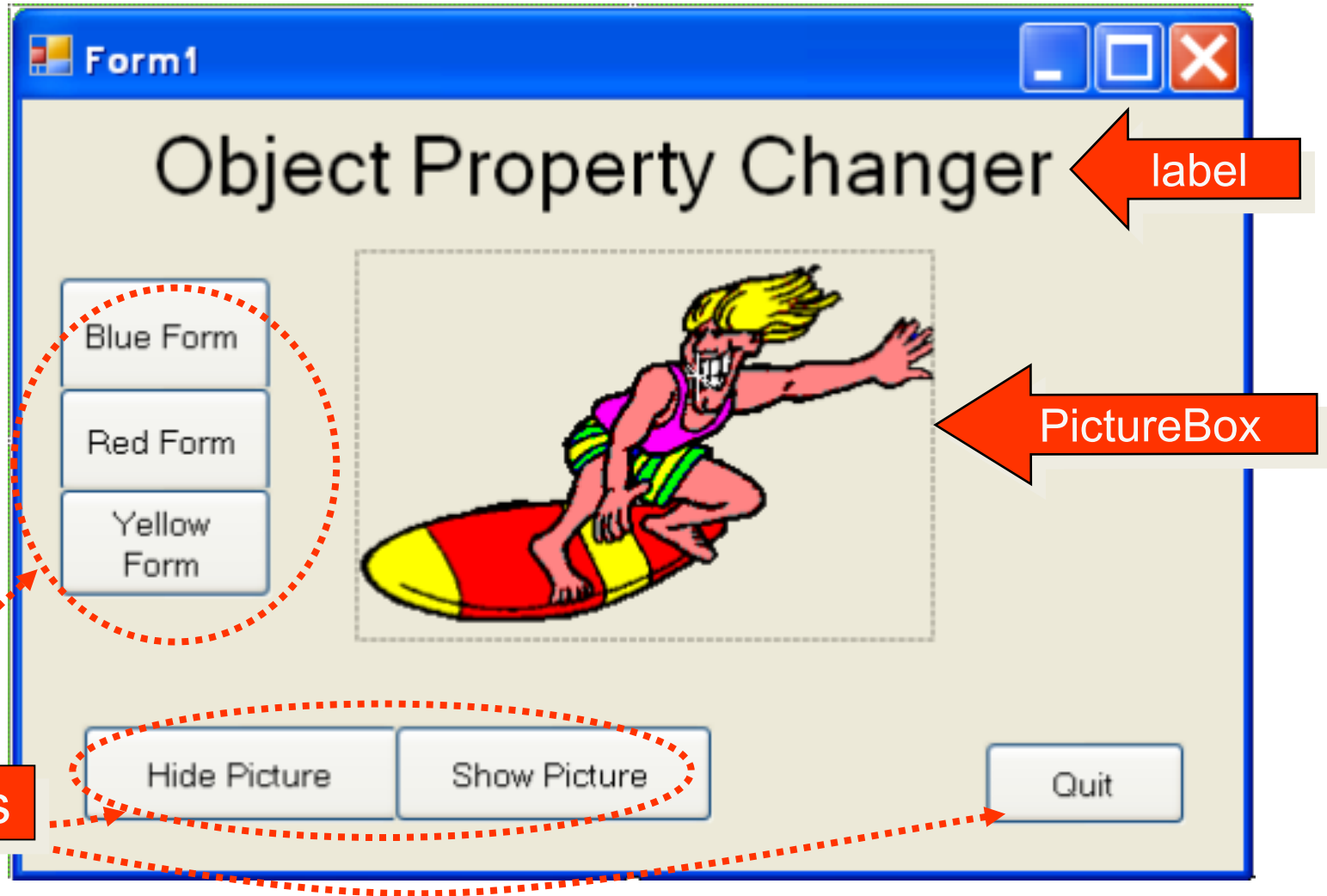
Writing a C# .NET Windows Program



Object Properties

How to change them
as the program runs
(run-time)

Create the Interface and set the properties



Add Code to the Buttons

```
Start Page Form1.cs* Form1.cs [Design]*
Surfer.frmSurfer btnQuit_Click(object sender, EventArgs e)
private void btnBlue_Click(object sender, EventArgs e)
{
    this.BackColor = Color.Blue;
}
private void btnRed_Click(object sender, EventArgs e)
{
    this.BackColor = Color.Red;
}
private void btnYellow_Click(object sender, EventArgs e)
{
    this.BackColor = Color.Yellow;
}
private void btnHide_Click(object sender, EventArgs e)
{
    pbxSurferImage.Visible = false;
}
private void btnShow_Click(object sender, EventArgs e)
{
    pbxSurferImage.Visible = true;
}
private void btnQuit_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

These change the BackColor property of this form

These change the Visible property of the PictureBox


Exit the Application

Running the Program

Form1

Object Property Changer

Blue Form
Red Form
Yellow Form




Hide Picture Show Picture Quit

Detailed description: This is the initial state of the application window. The window has a white background. On the left, there are three stacked buttons labeled 'Blue Form', 'Red Form', and 'Yellow Form'. In the center, there is a cartoon illustration of a surfer with blonde hair riding a red and yellow surfboard. At the bottom, there are three buttons: 'Hide Picture', 'Show Picture', and 'Quit'. Red dotted arrows point from the 'Blue Form', 'Red Form', and 'Show Picture' buttons to the corresponding windows below.

Form1

Object Property Changer

Blue Form
Red Form
Yellow Form




Hide Picture Show Picture Quit

Detailed description: The window background is blue. The 'Blue Form' button is highlighted with a white border. The surfer image and other UI elements are visible.

Form1

Object Property Changer

Blue Form
Red Form
Yellow Form




Hide Picture Show Picture Quit

Detailed description: The window background is red. The 'Red Form' button is highlighted with a white border. The surfer image and other UI elements are visible.

Form1

Object Property Changer

Blue Form
Red Form
Yellow Form



Hide Picture Show Picture Quit

Detailed description: The window background is yellow. The 'Yellow Form' button is highlighted with a white border. The surfer image and other UI elements are visible.

Form1

Object Property Changer

Blue Form
Red Form
Yellow Form


Hide Picture Show Picture Quit

Detailed description: The window background is yellow. The 'Yellow Form' button is highlighted. The surfer image is hidden. The 'Hide Picture' button is highlighted with a white border.

Form1

Object Property Changer

Blue Form
Red Form
Yellow Form

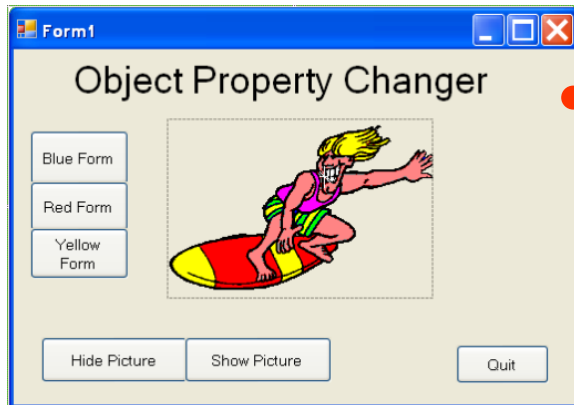


Hide Picture Show Picture Quit

Detailed description: The window background is yellow. The 'Yellow Form' button is highlighted. The surfer image is visible. The 'Show Picture' button is highlighted with a white border.

The Form's Load Event

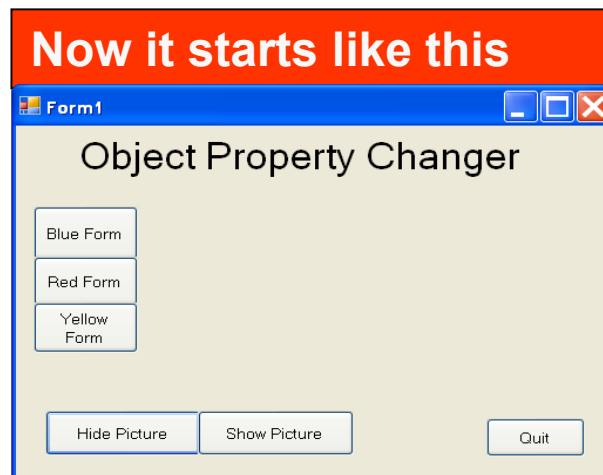
We often need to make something happen when a form first loads. Suppose we want the surfer to be invisible at the start



Double-click the form to bring up its code window

```
private void frmSurfer_Load(object sender, EventArgs e)
{
    pbxSurferImage.Visible = false ;
}
```

Then add code to the form's Load event method



Extra Reading
Formatting
Numeric
Output

A Reminder

If you want to format a double type of number to 2 decimal places:

```
lblResult.Text = answer.ToString("0.00");
```