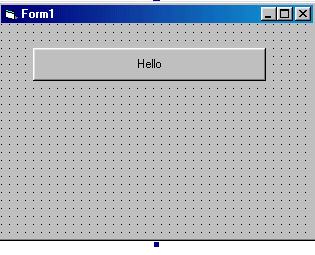
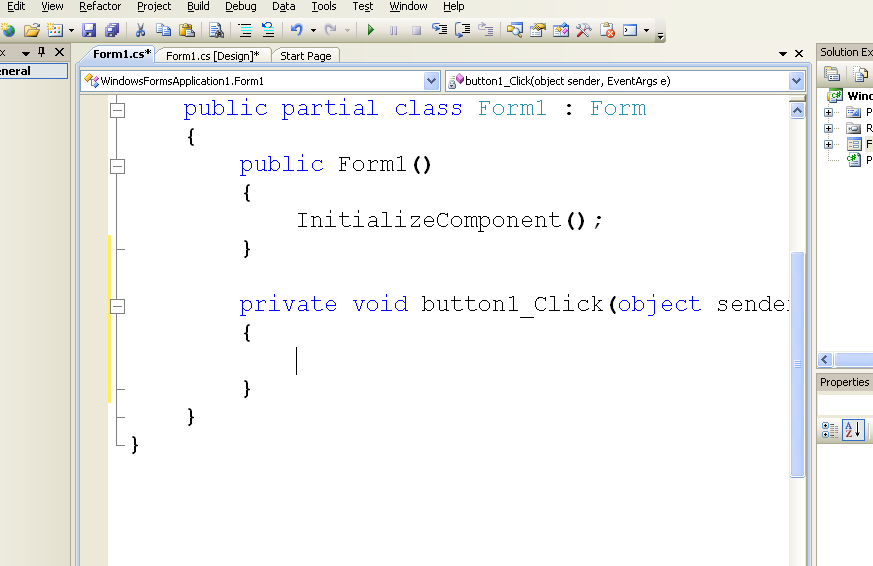
**­­­­­­Components/Controls/Objects: Their Properties and Events**

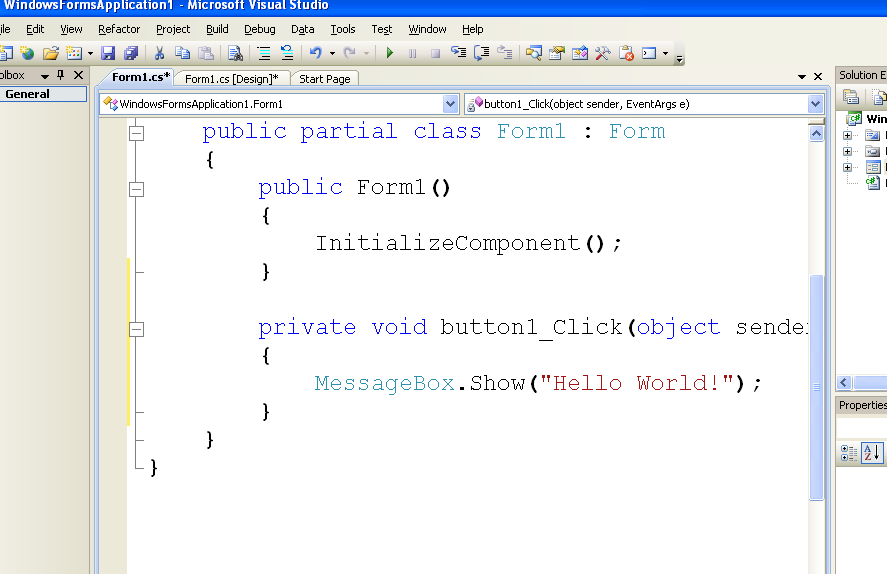
Components/Controls/Objects🡪include all the components seen in the C# toolbox

* Others can be added to the toolbox🡪Right Click on Toolbox, then select Choose Items…
* C# involves **design** and then **code**
* Therefore, it’s important not just how code is written but how a program’s interface is laid out (people like to use nice looking programs)
* Part of the design of a program involves setting the **properties** of each control - for ex. its width, height, backcolor, etc.
* Each control has a list of properties that reflect its **appearance and behaviour**
* These properties can be changed at **run** or **design time**
* To change properties during design time simply select the control on the form (remember a form is also a control/object) and then access its properties window
* In the properties window you will see a list of properties for the control
* You can change these properties here
* Properties usually have one of four main types of values **1. number ex.width, top 2. Boolean ex. enabled, visible 3. string ex. “hello” 4. object ex. font**
* Before changing a property you must know what kind of data it requires
* The second method to change a control’s properties is during run time i.e. when the program is running/being executed 🡪this is done in the code window
* However before any code can be written you must determine two things:1. when do you want the code to be executed i.e. when a person clicks a button or when a person moves their mouse cursor over a label etc. and 2. what will happen when this event occurs i.e. will a control’s property change or will a messagebox pop up etc.
* Here’s a simple example:
* 
* Here we have drawn onto a form a button control
* Its text property was set to “hello” in the **properties window**
* We set the text property to “hello” during design time because we don’t need it to change while the program is running—we want to change its value now
* Now we ask ourselves what do we want to happen when this button is pressed?—the answer is we want a small dialog box to pop up saying hello!
* Obviously this must happen while the program is running so we’ll have to write code for this to happen
* To do this we must access the code window for this form (click on View and then on Code
* the **form window/design window** disappears
* you are now ready to write code to tell your program to pop up a dialog box saying hello when the user clicks the button
* to do this you must first select a control from the form
* then select the events options from the properties window of the selected component
* then double click the event
* Visual Studio will enter the code window and write an event function for the event and the component (in the following example button1 has been set up to respond to a click event)



Code written in here will be executed when this event is occurs

Event function

* now we write the code that will respond to the event
* 

Messagebox.show(“info”) tells C# to pop up a dialog box with the info message inside of it

This code gets activated when the button called button1 gets clicked

**Questions and Exercises:**

* 1. Start C# and look at the toolbox. Each control has a name. You can find this name by holding your mouse over the control and looking for the tooltip to appear. Draw a button, a label ,a textbox and a control of your choice onto the form. Select each one and list the names of five properties for each one.

**Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| **Button** | **Label** | **TextBox** | **? (you pick a component)**  **Checkbox** |
| Text | TextAllign | RightToLeft | Checked |
| Cursor | Location | ScrollBars | BackgroundImage |
| Image | Forecolor | MultiLine | Padding |
| BackColor | Margin | Enabled | Locked |
| Autosize | Cursor | BorderStyle | ThreeState |

* 1. Now select for each control one property that requires a number, a string value and a Boolean value. Write the name of these properties in the appropriate cells in the table below.

**Property Values**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Button** | **Label** | **TextBox** | **? (you pick a component)**  **Checkbox** |
| String | Text | Forecolor | Backcolor | Font |
| Number | Location | Tabindex | Size | Margin |
| Boolean | TabStop | Locked | CausesValidation | Threestate |

* 1. Now select three different properties for each of these controls, change their values in the properties window and describe how these changes affect the appearance or behaviour of the control (if you notice nothing happens when you change the value of the property in the properties window then run the program and note any changes).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Property Name** | **Button** | **Label** | **TextBox** | Checkbox |
| Text |  |  |  |  |
| Size |  |  |  |  |
| BackColor |  |  |  |  |

* 1. Below is a list of properties for the button. Highlight those that affect the control during run time.

**Visible**

**Backcolor**

**Enabled**

**Width**

* 1. Select the **textbox**. List 5 different events for the textbox. Describe what you think must happen to the textbox for those events to occur.

a.

b.

c.

d.

e.

* 1. Select the **button**. Do the same.

a.

b.

c.

d.

e.

* 1. Are all the events for each control the same? Why do you think this is? List three different events that are common to both controls.
  2. Create a simple program that will display 3 lines of a song in three separate messageboxes when a button is clicked. Do the same thing when someone presses a key in a textbox. Add a label that says “Click Me!” in a large, bold, yellow font, colour its background blue and then set it up so that when a user hovers their mouse over it, a messagebox appears telling the user to “GET AWAY!”.

**9. Zip this handout and the project from exercise #8 into one file. Post a link to your Unit 2 “My Work” section. Put it under Lesson2.**