Lab # 2: User Interface Design [Feb 17, 2011]

Course: BESE 15 B **Instructor: Lab Engr. Nausheen**

OBJECTIVE

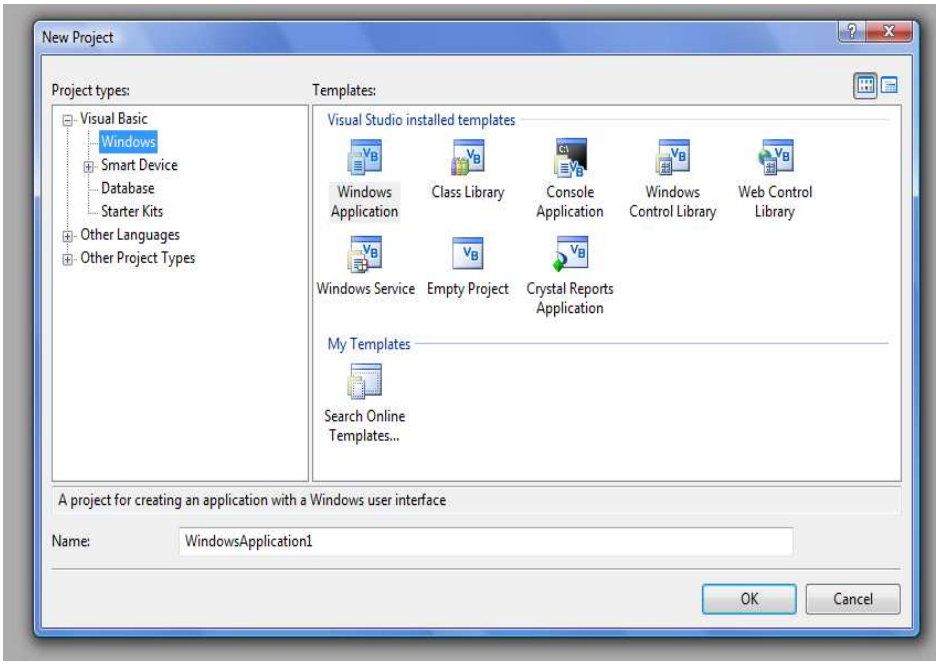
Today's lab will introduce you to the basics of designing a user interface using C#.Net. The first task will guide you in a step-wise manner and you will design a simple interface that accepts two numbers from the user and performs their addition. In the second task, you will design a very basic calculator.

By the end of this lab, you will learn how to use:

- Labels
- Text boxes
- Buttons

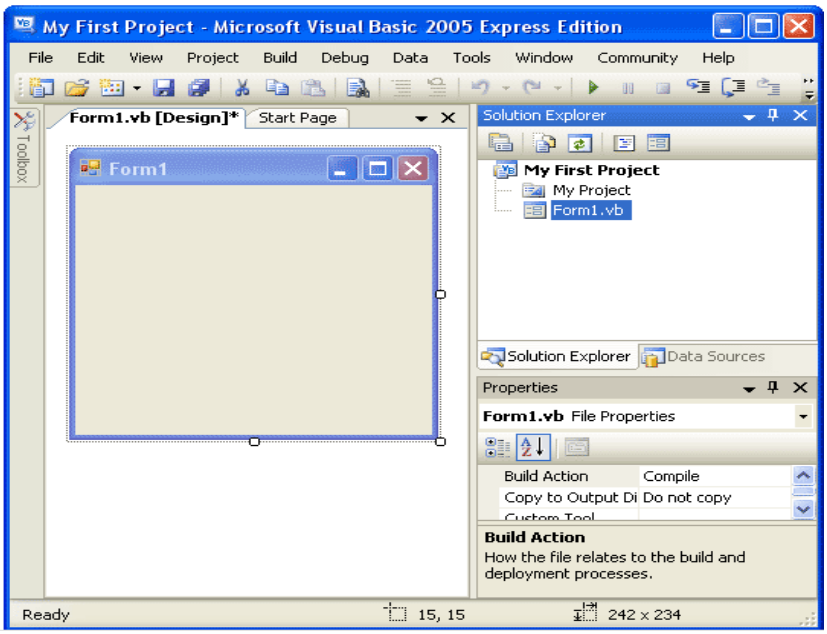
TASK 1

- Launch your Visual Basic .NET.
- To get started, click the File->New Project button. When you do, you'll see this dialogue box appear:



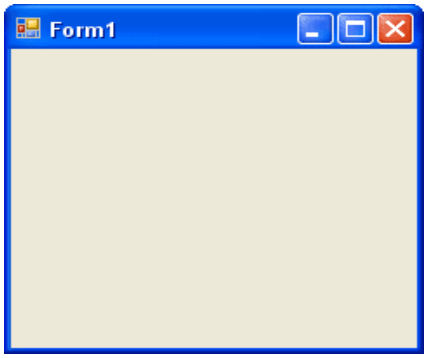
- Make sure that in Project Type C# is selected.
- As a beginner, you'll normally want the option selected: "Windows Application", in the "Visual Basic Projects" folder

- If you look in the Name textbox at the bottom, you'll see it says "WindowsApplication1". This is the default name for your projects. Click inside this textbox and change this Name to any meaningful description (e.g. "AddingNumbers").
- Click the OK button, and the Visual Basic NET design time environment will open. It will look like the following :

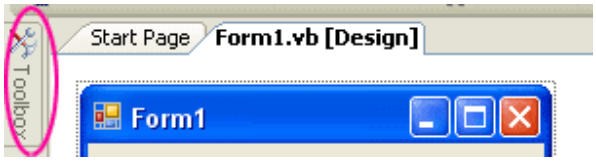


- Save your project (preferably in your account), a new folder will then be created for you, and its name will be the one you typed in the "Name" textbox.
- In the Visual Basic NET design time environment, the big square that you see is called a form. It's actually your program, the part that others will see when your program is launched. To run the form, try this:
  - From the menu bar, click **Debug**
  - From the drop down menu, click **Start**
  - Alternatively, press the **F5** key on your keyboard
  - Your program is launched

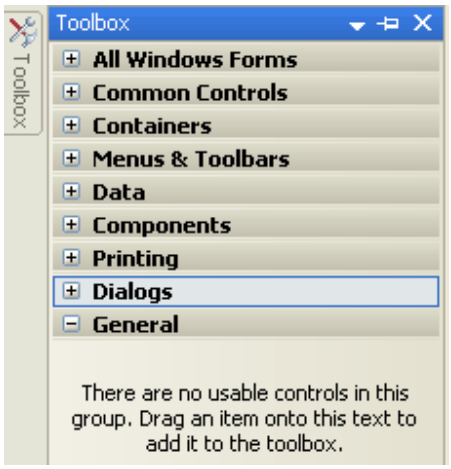
You have now created your very first program. It should look like this:



- Close the Form and you will return to the design environment.
- If you compare the first form with the one above, you'll see that they look very similar. Visual Basic has two distinct environments, a **Design** environment and a **Debug** environment. Design Time is where you get to play about with the form, add textboxes, and buttons, and labels (and code, of course); Debug is where you can test your program and see how well it performs.
- Things like buttons, textboxes, and labels are all things that you can add to your Forms. They are know as Controls, and are kept in the Toolbox for ease of use.
- The Toolbox can be found on the left of the screen. In the picture below, you can see the toolbox icon next to Form1:



- To display all the tools, move your mouse over the toolbox icon. You'll see the following automatically appear:

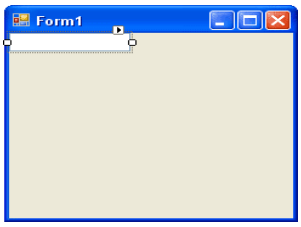


- There are seven categories of tools available. The toolbox you'll be working with first is the Common Controls toolbox. To see the tools, click on the plus symbol next to **Common Controls**. You'll see a long list of tools.
- As you can see, there are a lot of tools to choose from! For today's exercise, we'll only be using the **Button**, the **TextBox** and the **Label**.
- If you want to keep the toolbox displayed, click the Pin icon next to the X. To close the toolbox, simply move your mouse away.

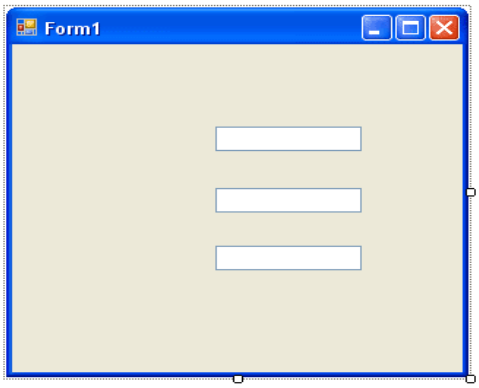
Adding Controls to the form

- We'll start by adding a textbox to the form. With the tools displayed, do the following:
  - Locate the TextBox tool
  - Double click the icon
  - A textbox is added to your form

- The textbox gets added to the top left position of your form. To move it down, hold your mouse over the textbox and drag to a new position:



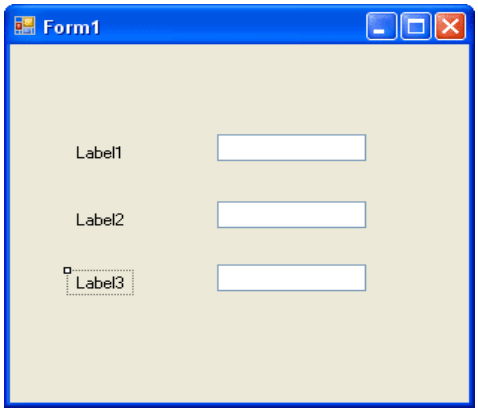
- Resize the textbox to an appropriate size.
- Create two more textboxes by double clicking on the textbox icon in the toolbar
- Resize them to the same size as your first one
- Line them up one below the other with space in between
- Try to create something that looks like the one below:



Adding a Label to your Form

Let's add some labels near the textboxes so that your users will know what they are for.

- Locate the label control in the toolbox
- Double click the label icon
- A new label is added to your form
- Add three labels to have a form like this:

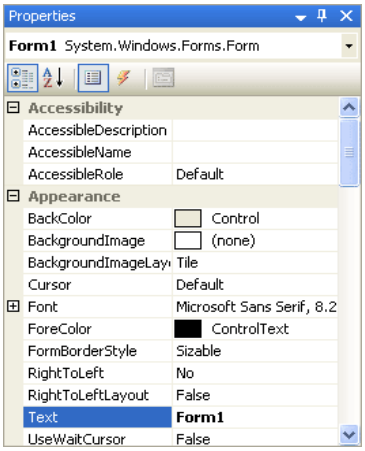


Control Properties

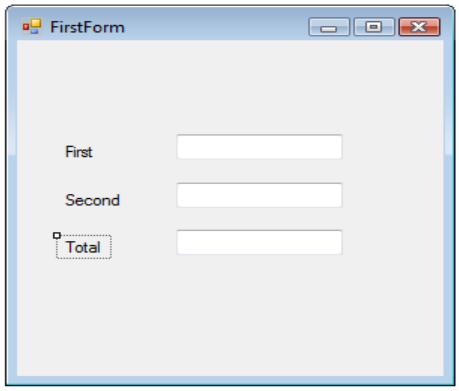
The controls you added to the form (textboxes and labels), and the form itself, are called control objects. You can think of controls as things, something solid that you can pick up and move about. Controls have properties (like size, color etc).

You can change a property of a control from the Properties Box. (You can also change a property using code, which you'll do quite a lot.) If we go back to our Form object, and the properties and values it has, we can see how to change them using the Properties Box. We'll change only one of these values for now - the value of the Text property. So, do this:

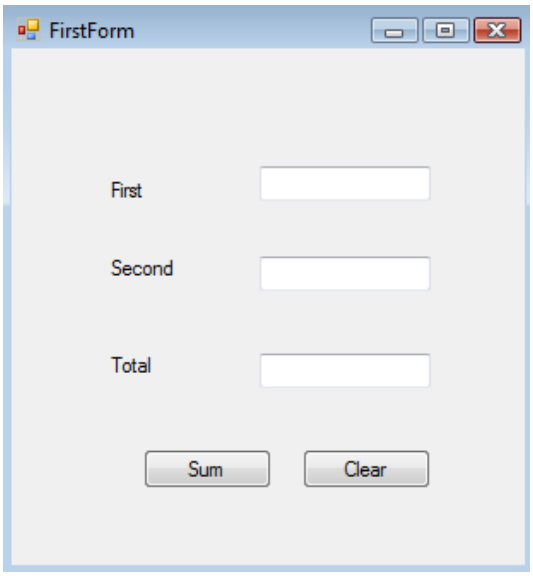
- Click anywhere on the form that is not a label or a textbox, somewhere on the form's grey areas.
- On the right of the design environment there should be the following Properties box:



- What you are looking at is a list of the properties that a form has: Name , BackColor, Font, Image, Text, etc. Just to the right of these properties are the values for them. These values are the default values, and can be changed. We're going to change the value of the Text property.
- Find the following properties and change them to appropriate values of your choice:
  - Text – The title of form that appears on the title bar
  - Background Color
  - Size
  - Etc.
- Similarly, change the text property of the three labels to have a form like this:

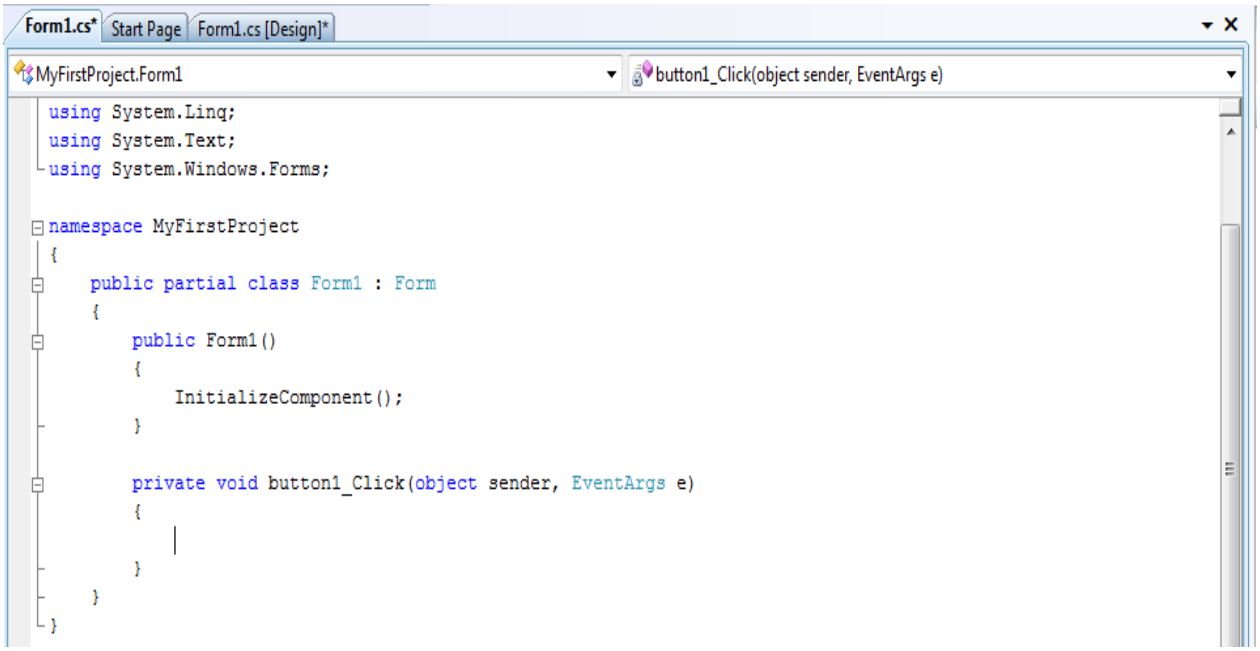


- Now add two buttons to the form:



Adding functionality to Buttons

To get our first look at the code window, double click your Button control. The code window will appear, and will look like this:



Here's the part we're concentrating on:

button1_Click(object sender, EventArgs e): This is something called an Event. In other words, when the button is clicked, the Click Event will fire, and the code we're going to write will be executed Now, type the following code inside this method:

```
int first, second, result;
first =  Int32.Parse(textBox1.Text);
second = Int32.Parse(textBox2.Text);
result = first + second;
textBox3.Text = result.ToString();
```

- In a similar fashion, add the following functionality to the “clear” button.

```
TextBox1.Clear();
TextBox2.Clear();
TextBox3.Clear();
```

- Run and test your program. It will add the two numbers provided by the user.

TASK 2

Design a simple CALCULATOR. The interface must have digits 0 to 9, +, -,x, /,= and ‘Clear’ buttons.

You only need to put very basic functionality in the calculator (two operands only). E.g. operations like:

5 + 10 = 15

15 / 3 = 5

You do not need to cater for operations like 5+10 X 2; 4+3+5 etc.