Lab 2 – User Interface Design Using VB.NET

Human Computer Interfacing

(BESE 14)

Objectives

Today's lab will introduce you to the basics of designing a user interface using VB.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

In your interfaces.

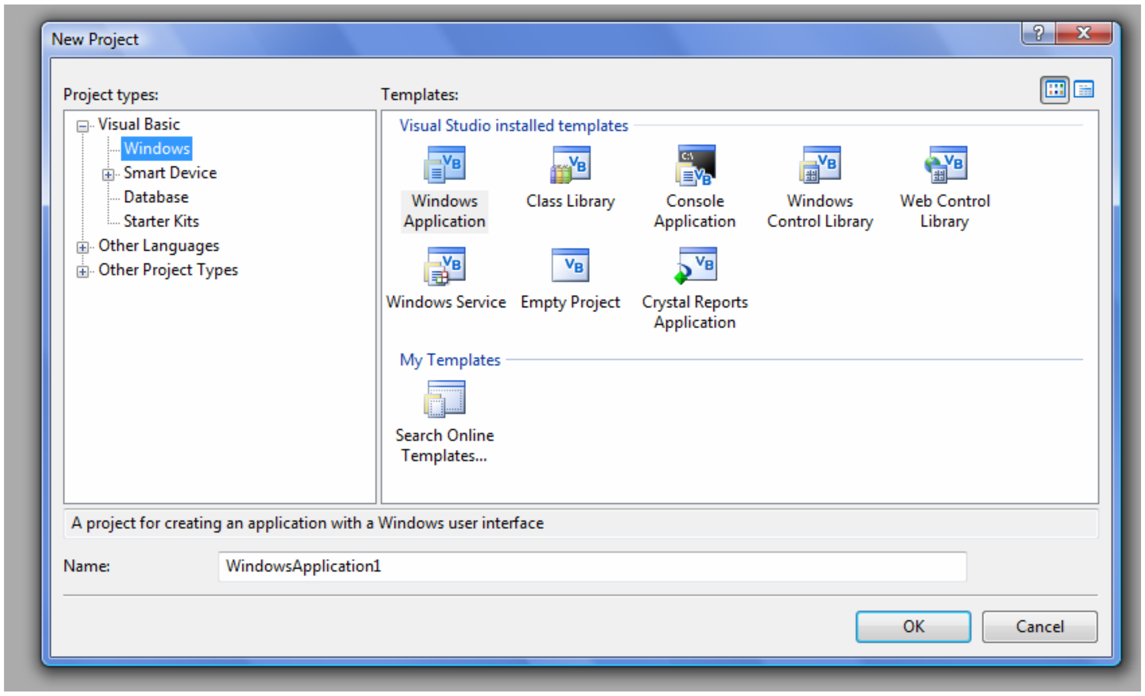
In addition to the step by step guide below, you will also be explained different VB controls and adding functionality to them, during the lab session.

You are also free to use any form of online/offline help as long as you cite it in your code/report.

Task 1

Note: This task is a practice exercise and you do NOT need to submit any lab report/program code.

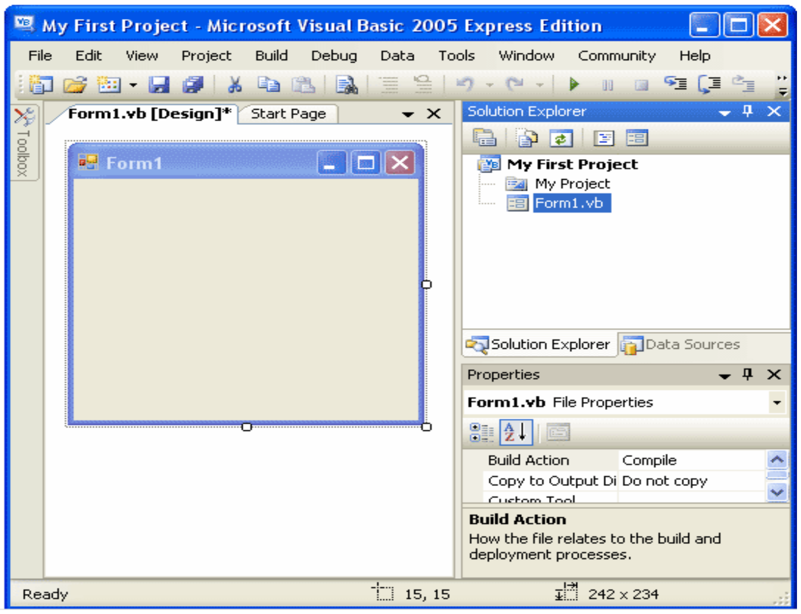
- Launch your Visual Basic .NET or Visual Studio software.
- To get started, click the File->New Project button. When you do, you'll see this dialogue box appear:



A project for creating an application with a Windows user interface

Name: WindowsApplication1

- Make sure that in Project Type Visual Basic 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 :

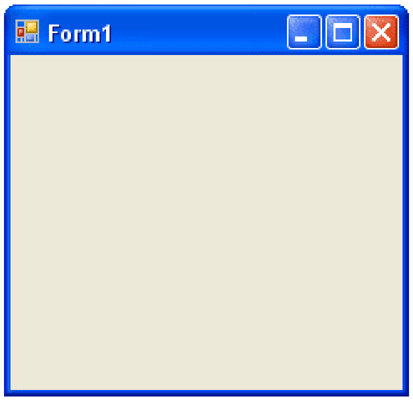


Build Action	Compile
Copy to Output Di	Do not copy
Custom Tool	

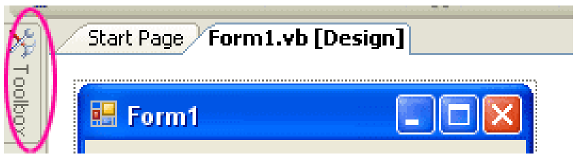
Build Action
How the file relates to the build and deployment processes.

- 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. All of your files for your first project are then saved in this folder.
- 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

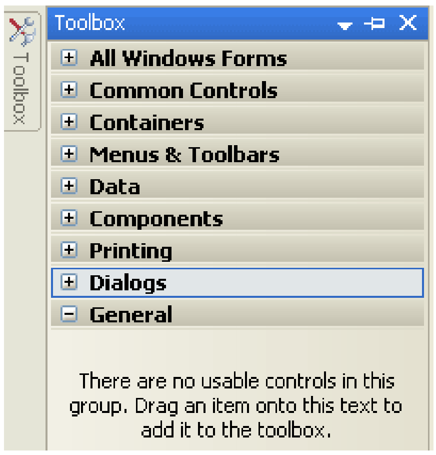
You have now created your very first program. It should look like this:



- 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 no usable controls in this group. Drag an item onto this text to add it to the toolbox.

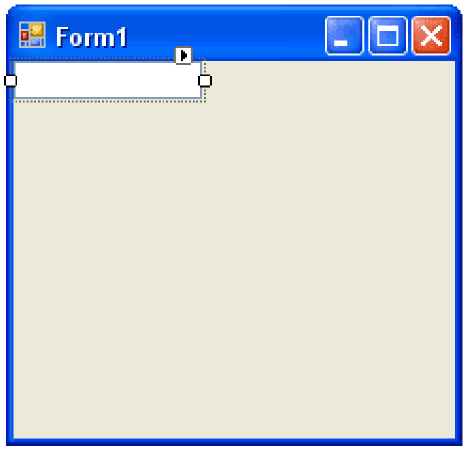
- There are seven categories of tools available. The toolbox you'll be working with first is the Common Controls toolbox. To see the tools, click on the plus symbol next to **Common Controls**. You'll see a long list of tools.

- 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**.

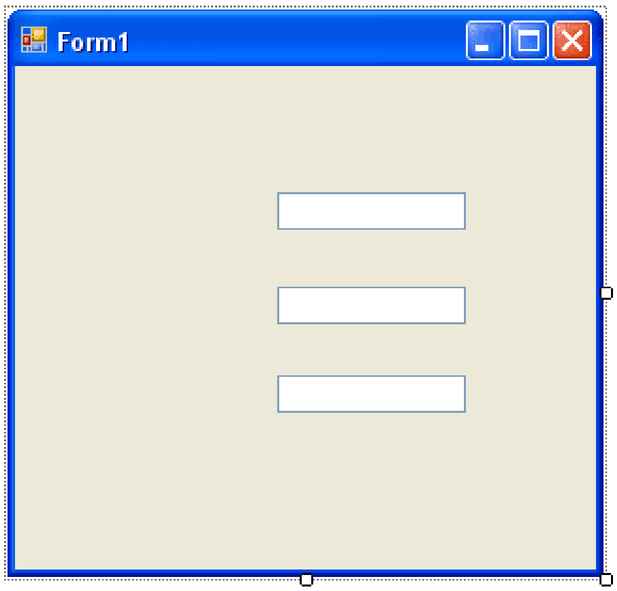
- If you want to keep the toolbox displayed, click the Pin icon next to the X. To close the toolbox, simply move your mouse away.

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:

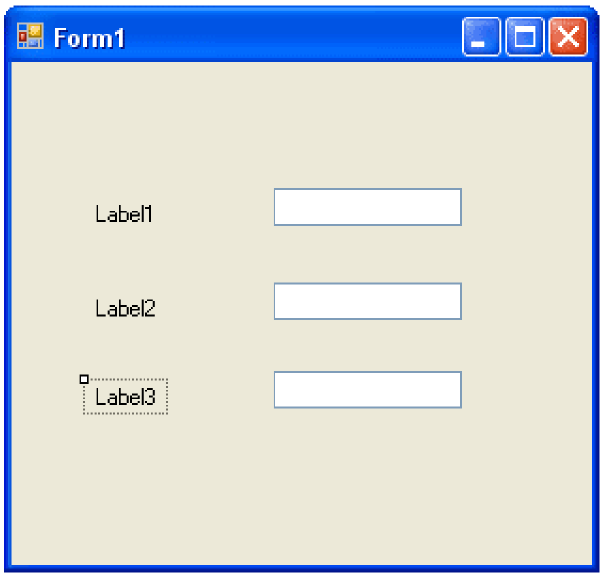


•

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:

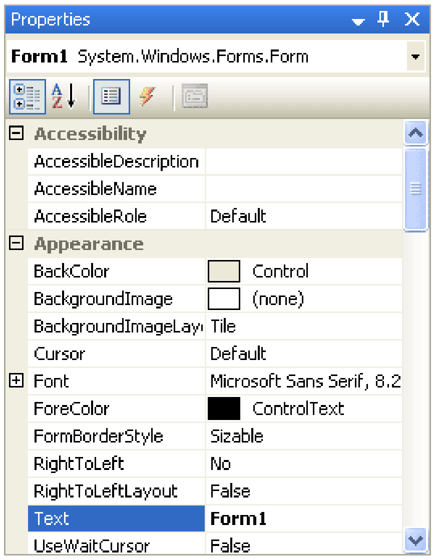


Control Properties

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).

In VB.NET, 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:



Properties

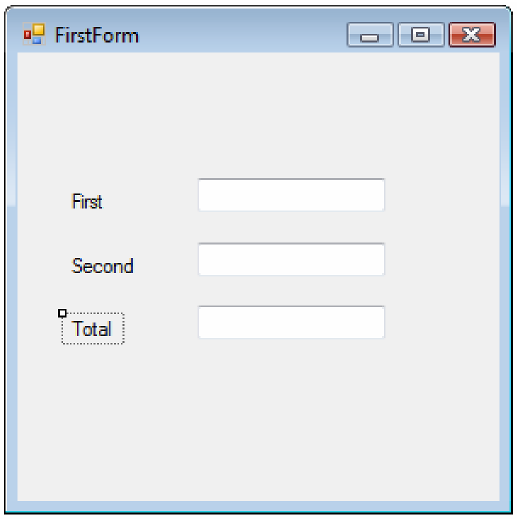
Form1 System.Windows.Forms.Form

⊟ **Accessibility**	
AccessibleDescription	
AccessibleName	
AccessibleRole	Default
⊟ **Appearance**	
BackColor	☐ Control
BackgroundImage	☐ (none)
BackgroundImageLay	Tile
Cursor	Default
⊞ Font	Microsoft Sans Serif, 8.2
ForeColor	■ ControlText
FormBorderStyle	Sizable
RightToLeft	No
RightToLeftLayout	False
Text	**Form1**
UseWaitCursor	False

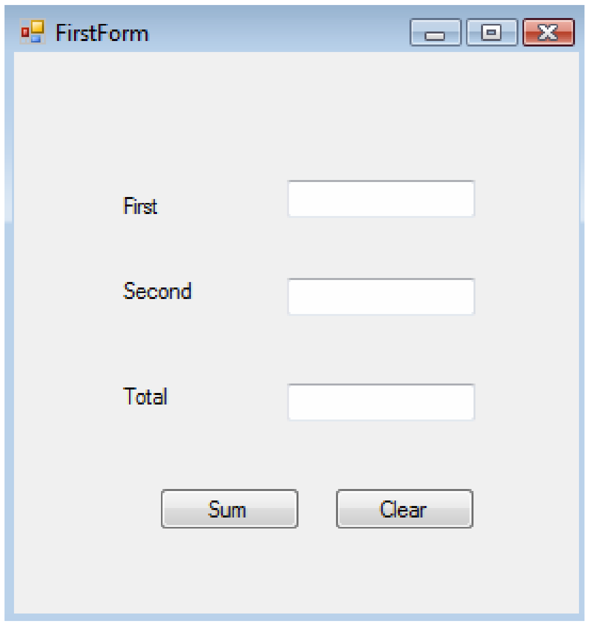
- What you are looking at is a list of the properties that a form has: Name , BackColor, Font, Image, Text, etc. Just to the right of these properties are the values for them. These values are the default values, and can be changed. We're going to change the value of the Text property.
- Find the following 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:



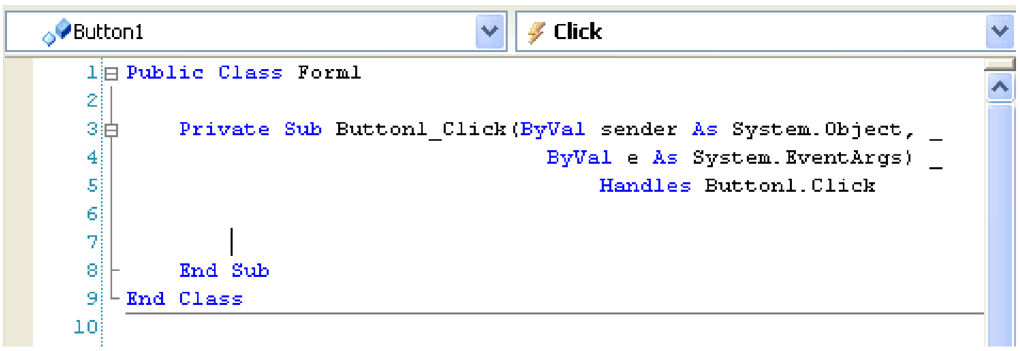
- Now add two buttons to the form:



STOP HERE

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:



```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, _
                              ByVal e As System.EventArgs) _
                                 Handles Button1.Click


    End Sub
End Class
```

Here's the part we're concentrating on:

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) _
Handles Button1.Click

End Sub
```

The part of the code we're interested in is highlighted in red in the code above. Notice, too, that the underscore character (_) has been used to spread the code over more than one line. You can do this in your own code, too, if it becomes to long

Private

Private means that no other part of the programme can see this code except for our button

Sub

Short for Subroutine. The "Sub" word tells VB that some code follows, and that it needs to be executed

Button1

This is the name of our button. You might think that we've just erased the word "Button1" when we changed the Text property, so why does VB insist that it's still called Button1? Well, this is because it is Name property of the control that is used in the code.

_Click ( )

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

End Sub

The subroutine ends right here. This signifies the end of our code

Click your mouse on the blank line after **Private Sub Button1_Click**, etc, but before End Sub. Type the following code (It will be explained during the lab session):

```
Dim first As Integer
Dim second As Integer
Dim result As Integer

first = TextBox1.Text
second = TextBox2.Text

result = first + second


TextBox3.Text = result
```

- 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.

<u>**Task 2**</u>

Note: You need to submit your program code for this task. If you do not manage to finish it within the lab session, you may submit it later on but <u>**before the start of the next lab**</u> (next week).

You need to design a simple CALCULATOR.

Requirements: 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.

Guidelines (Will be explained in the lab session)

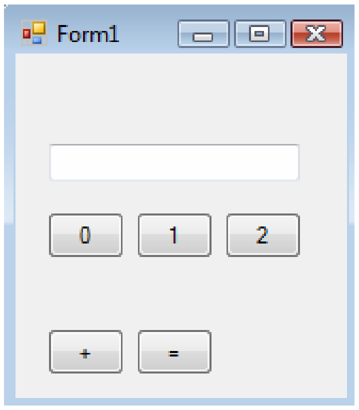


Figure 1 Basic idea (not complete)

- When user presses a key, the value is displayed in the text box.
- To get the value of the key pressed use:
  - Number = button0.Text and so on …
- To get values like 55, 14, 23 etc use:
  - TextBox1.Text = TextBox1.Text & button0.Text

- When the user presses + key, the contents of text area are stored in a variable and the text area is cleared.
- The user then inputs the second number which is also displayed in the text box.
- When the user clicks ‘=’, get the value of second operand from the text box, add it to the first operand (already stored) and display the result.

You may use Help from the cited reference or any other resource (but do mention it in your code).

Reference:

These exercises are based on an online VB.NET tutorial that can be found at:

http://www.homeandlearn.co.uk/NET/vbNet.html