MAKERERE UNIVERSITY BUSINESS SCHOOL

Bachelor of Business Computing

BUC 2227: Business Application Programming
Lecture 2

Introduction to VB Language and IDE

- **Visual Basic** is a tool that allows you to develop Windows Graphic User Interface (**GUI**) applications.
- It is **event-driven**, meaning the code remains idle until called upon to respond to some event (button pressing, menu selection etc)

Some Features of Visual Basic

- Full set of objects - you 'draw' the application
- There are Lots of icons and pictures for your use
- It responds to mouse and keyboard actions
- Clipboard and printer access
- It has a full array of mathematical, string handling, and graphics functions

A Visual Basic Application is made up of;

- **Forms** - Windows that you create for user interface
- **Controls** - Graphical features drawn on forms to allow user interaction
- **Properties** - Every characteristic of a form or control is specified by a property
- **Methods** - Built-in procedure that can be invoked to impart some action to a particular object.

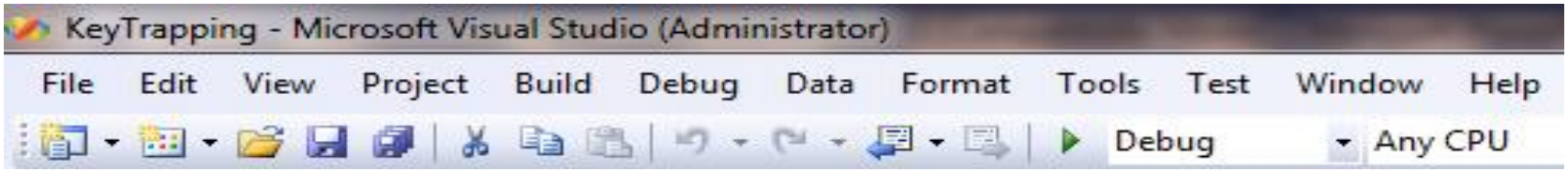
Steps in developing a VB application

1. Draw the user interface
2. Assign properties to controls
3. Attach code to controls

The VB Interface windows

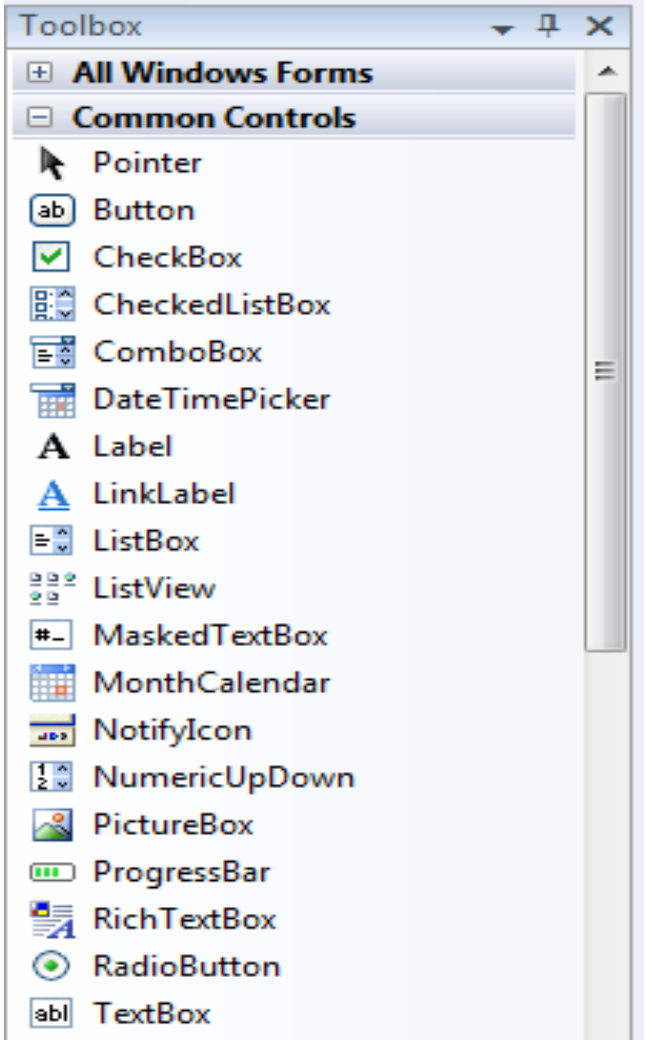
- Visual basic has a number of windows that the developer interacts with. These include;
  - The Main Window
  - The Tool box Window
  - The Design Window
  - The Solution Explorer Window
  - The Properties Window
  - The Code Window

The Main Window



- It consists of the title bar, menu bar, and toolbar.
  - The *title bar* has project name, the current VB operating mode, and the current form.
  - The *menu bar* has drop-down menus from which you control the operation of the Visual Basic environment.
  - The *toolbar* has buttons that provide shortcuts to some of the menu options.

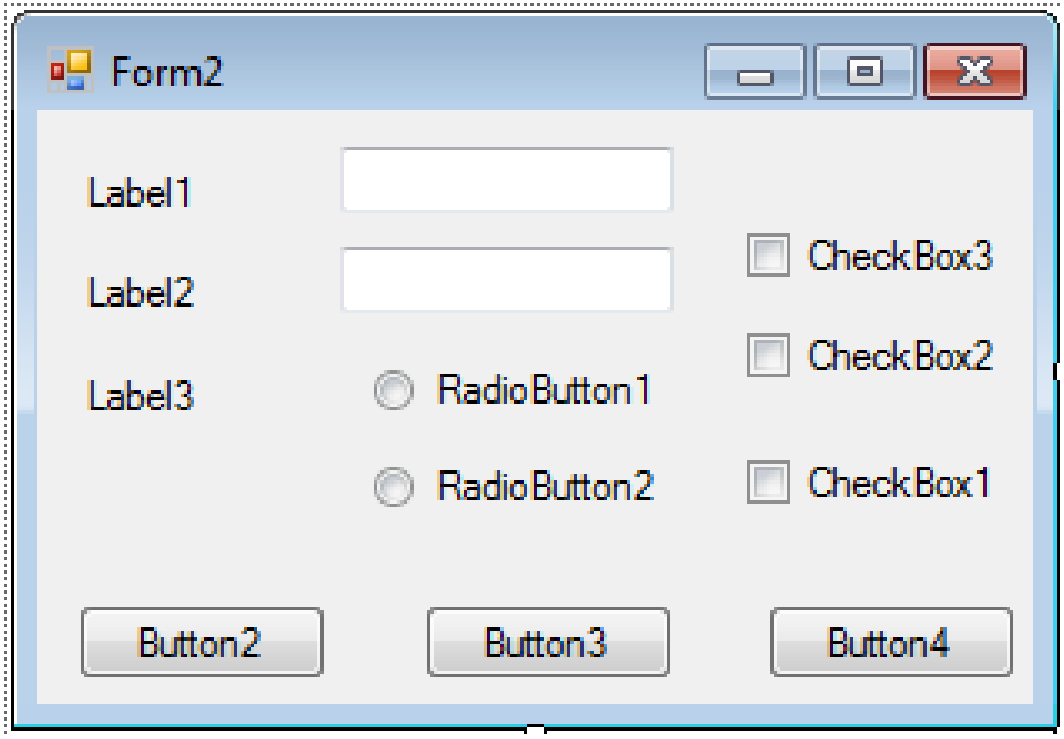
The Tool box Window



- This windows contains the tools that we use to draw/design our user interfaces eg a Textbox, button, radio button, label, check box etc

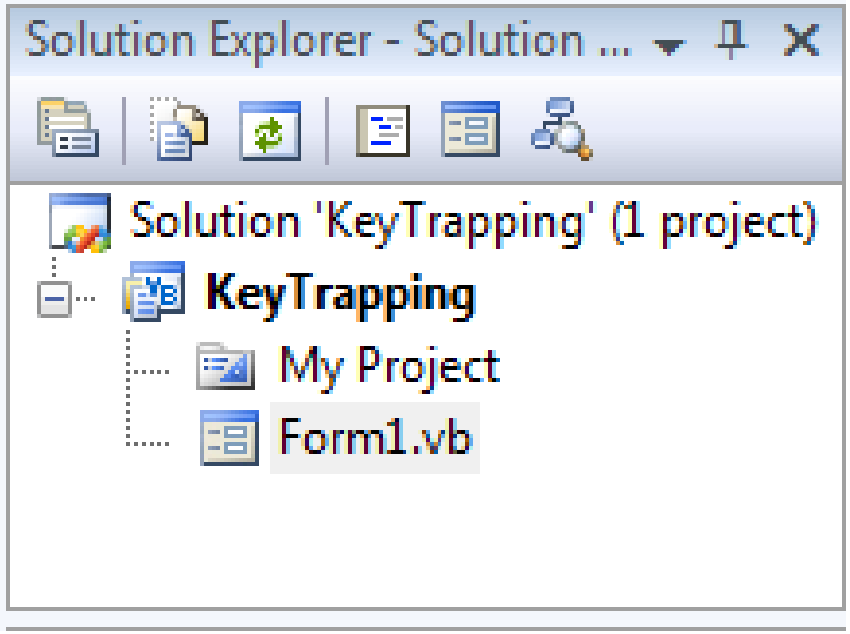
The Design Window

- The design window gives us access to the form and it is where we draw/create our graphical user interfaces



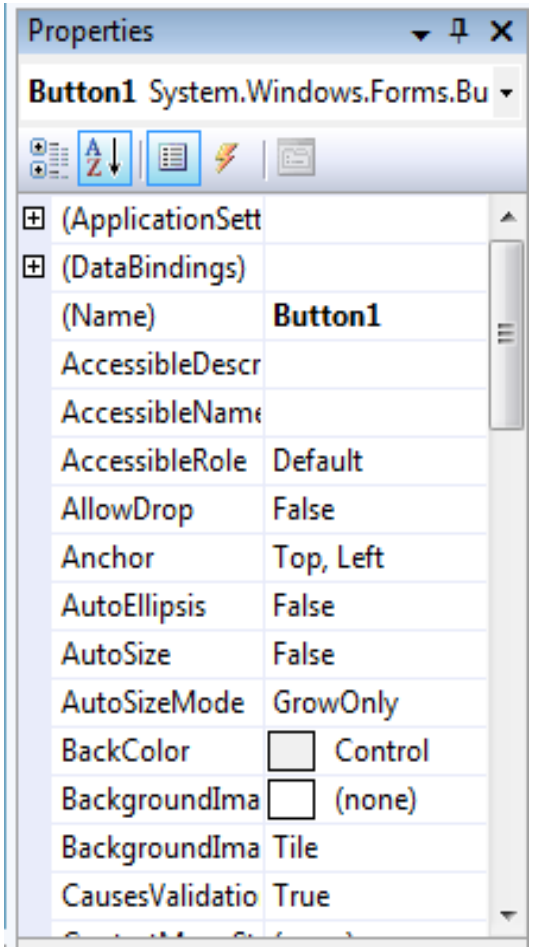
The Solution Explore Window

- This window displays all the elements that make up your project for example forms, modules and Classes



The Properties Window

- It is used to establish initial property values for objects. The drop-down box at the top of the window lists all objects in the current form. Two views are available: Alphabetic and Categorized.
- Under this box are the available properties for the currently selected object



The Code Window

Form1 | Button2_Click

```
Public Class Form1
    Private Sub txtnumbers_KeyPress(ByVal sender As Object, ByVal e As
        'To capture only numbers,decimals and backspace in the textbox
        If (e.KeyChar < Chr(48) Or e.KeyChar > Chr(57)) And e.KeyChar
            e.Handled = True
        End If
    End Sub

    Private Sub txtAlphas_KeyPress(ByVal sender As Object, ByVal e As
        If (Microsoft.VisualBasic.Asc(e.KeyChar) < 65) Or (Microsoft.V
            'space allowed
            If (Microsoft.VisualBasic.Asc(e.KeyChar) <> 32) Then
                e.Handled = True
            End If
        End If
        If (Microsoft.VisualBasic.Asc(e.KeyChar) = 8) Then
            e.Handled = False
        End If
    End Sub
```

- It provides us a view of the code or the logic that runs our applications
- It contains the actual Basic coding.

The Visual Basic Environment

Visual Basic operates in **three modes**.

- **Design mode** - used to build application
- **Run mode** - used to run the application
- **Break mode** - application halted and debug the application in case of errors.

Naming controls

- A convention has been established for naming Objects
- We use a three character prefix (depending on the object) followed by the name you assign.
- Control name does **not allow some characters** to be part of the name, for instance a Space, / or \, =, “, ”.
- The control name can be anything for as long as you can refer to it during the code development stage, therefore it is advisable and good practice to use a meaningful name in reference to the purpose of the control.
- The control name should **not be too long,** at most a maximum of 15 characters

Some common examples of prefixes and controls

Object	Prefix	Example
Form	frm	frmWatch
Command Button	cmd, btn	cmdExit, btnStart
Label	lbl	lblStart, lblEnd
Text Box	txt	txtTime, txtName
Menu	mnu	mnuExit, mnuSave
Check box	chk	chkChoice

PRACTICAL APPLICATION 1

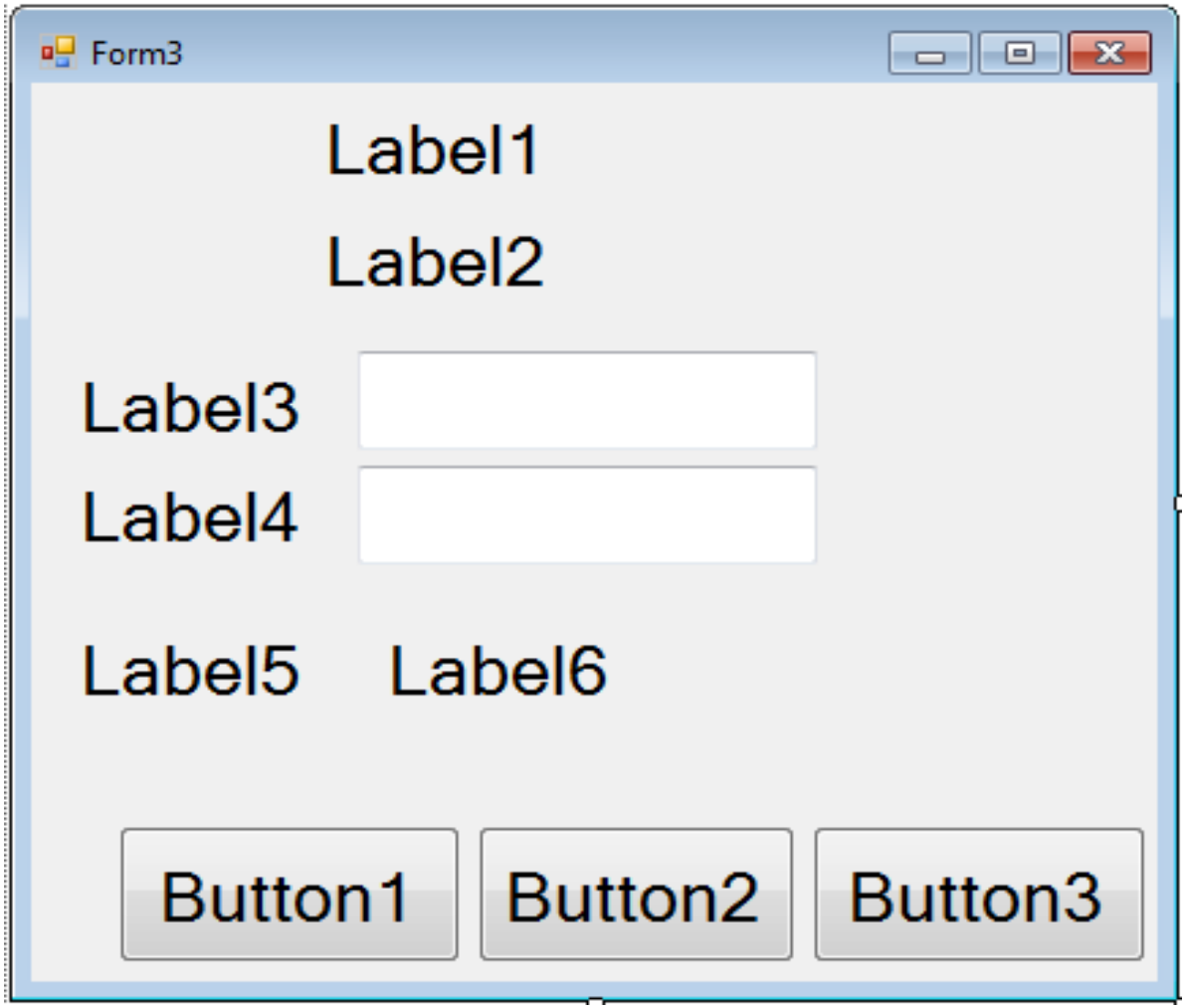
Application description;

- Request a user to enter two values (Integers)
- Add them together
- Display the result on the screen

Do you recall the steps?

1. Draw the user interface
2. Assign properties to controls
3. Attach code to controls

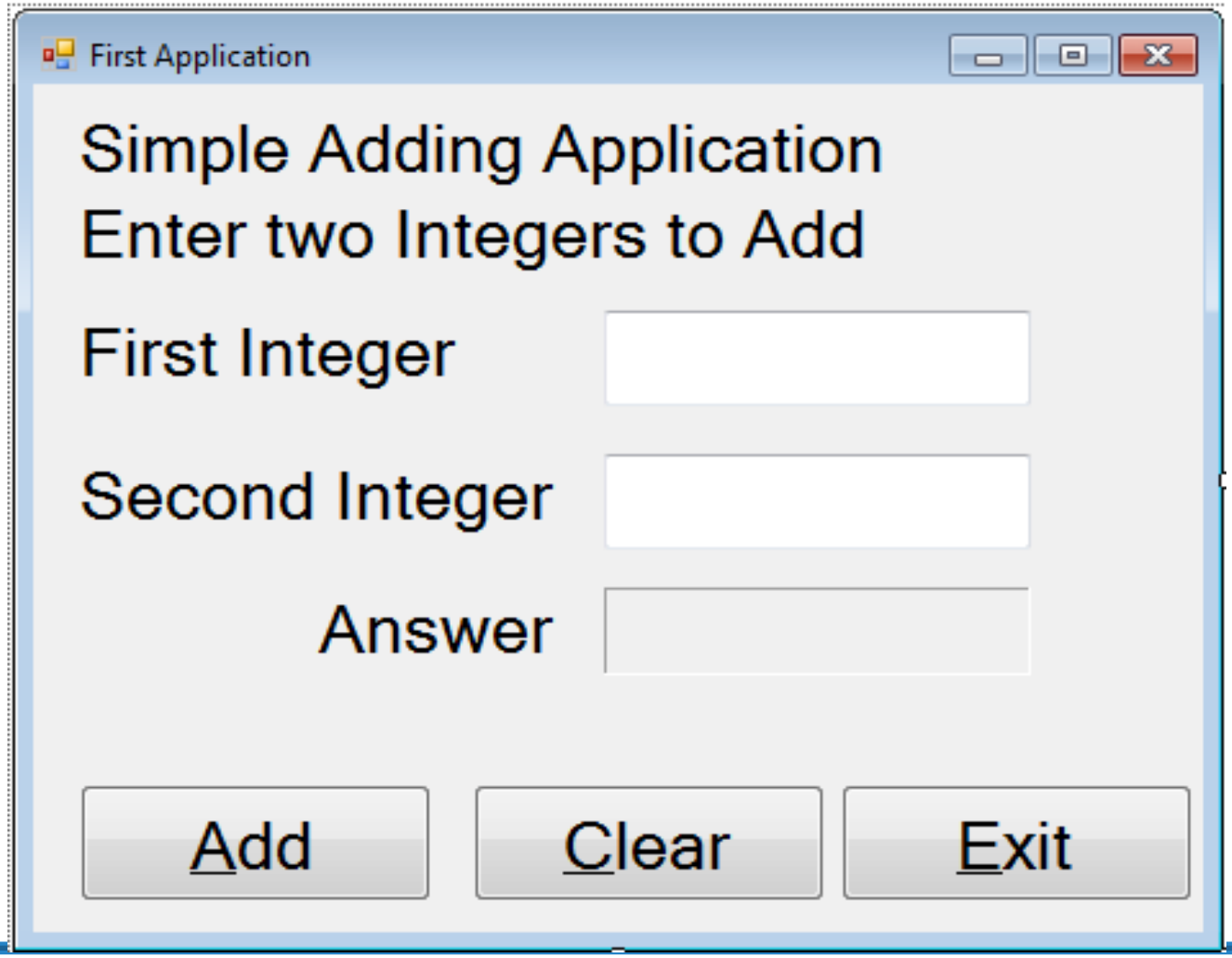
Draw the user Interface as below



Attach the following properties to the respective controls

Control	Name	Text	Font & Size
Form	frmApplication	First Application	
Label 1	Label 1	Simple Adding application	Garamond, 18, Bold
Label 2	Label 2	Enter two Integer to Add	Garamond, 18, Bold
Label 3	Label 3	First Integer	Garamond, 12, Bold
Textbox 1	Txtnumber1	[blank]	Garamond, 12, Bold
Label 4	Label 4	Second Integer	Garamond, 12, Bold
Textbox 2	Txtnumber2	[blank]	Garamond, 12, Bold
Label 5	Label 5	Answer	Garamond, 12, Bold
Label 6	LblAnswer	[blank]	Garamond, 12, Bold
Command1	cmdAdd	&Add	Garamond, 12, Bold
Command2	Cmdclear	&Clear	Garamond, 12, Bold
Command3	cmdexit	&Exit	Garamond, 12, Bold

When you are done attaching properties, your interface should now look like this



Now, its time to give life to your Application by attaching code to controls/writing code

- After a good interface design, the next step is to write the program statements known as the code that will activate the commands and make the application work.
- Code development stage involves the following steps:

Declaration of variables

- A variable is a temporally memory location where application values/data is stored
- A variable's data is temporarily stored in a data storage area

Variable declaration:

The structure:

- **Dim variable name As Data type**
- **Eg Dim Num1 As Integer**
- We use the word "Dim" to declare variables. Dim stands for "Dimension"
- **Note:**
- You must separate "Dim" and the name of the variable with a space.
- "As" is an instruction to the computer to define the data type of new variable.

The next step in code…..

- Assigning values to variables
- The Structure is;
- VariableName = Value

Eg;

- Answer=firstNum+secondNum
- Answer here is the variableName
- While “firstNum+secondNum” is the value

Attach the following code to cmdAdd


```
Private Sub cmdAdd_Click(ByVal sender As_ System.Object,
ByVal e As System.EventArgs) Handles cmdAdd.Click
        Dim firstNum As Integer
        Dim secondNum As Integer
        Dim answer As Integer
       firstNum = Val(Txtnumber1.Text)
       secondNum = Val(Txtnumber2.Text)
        answer = firstNum + secondNum
        LblAnswer.Text = answer
    End Sub
```

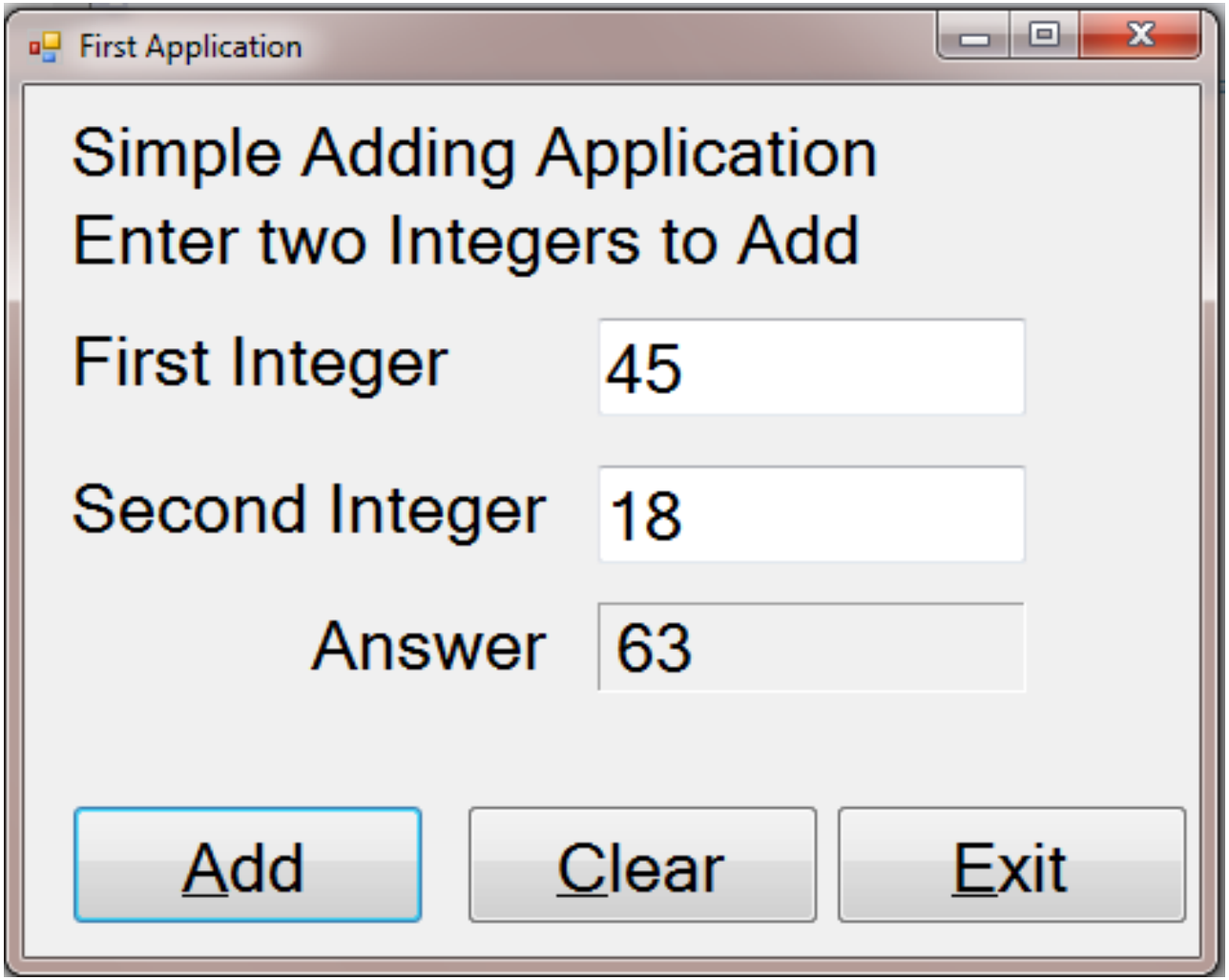
Add the following code to cmdclear


```
Private Sub cmdclear_Click(ByVal sender As_
System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
    Txtnumber1.Text = ""
    Txtnumber2.Text = ""
    LblAnswer.Text = ""
 End Sub
```

Add the following code to cmdexit


```
Private Sub cmdexit_Click(ByVal sender As_
System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        End
End Sub
```

If you have succeeded on the above steps, run your Application. It should appear like this;



Learning Exercise

1) Add more three buttons for Subtraction, Multiplication and Division
2) Attach the relevant properties to the above commands
3) Write the necessary code under each button to activate their functionalities

Next Lecture