

Topic 2A

Introduction to C# and Windows Form

Topics

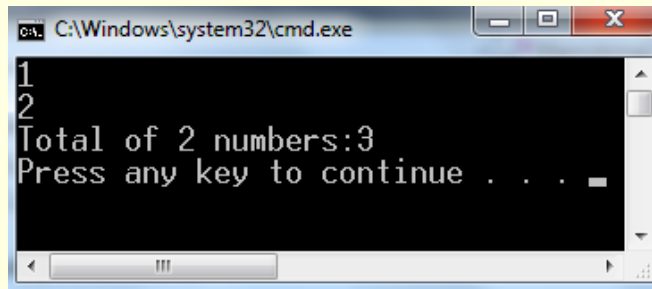
- ❑ Introduction to the C# Language
- ❑ A Word about Windows Form
- ❑ Creating a new project, compile and execute a C# program

Objectives:

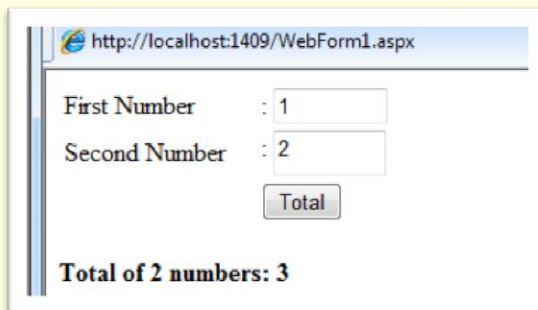
- ❑ Be able to understand what C# can achieve
- ❑ Be able to understand fundamentals of Windows Form
- ❑ Be able to create a C# project, compile and execute the program.

Introduction to the C# Language

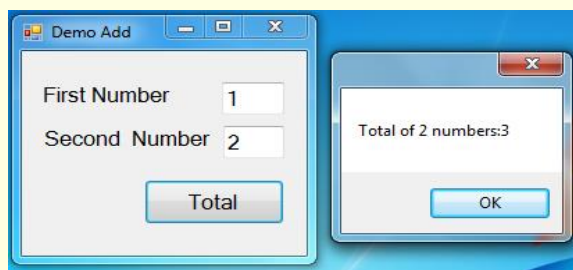
- ❑ The C# language that you will learn in this module can be used in many environments:
- ❑ Console Applications



- ❑ Web Applications

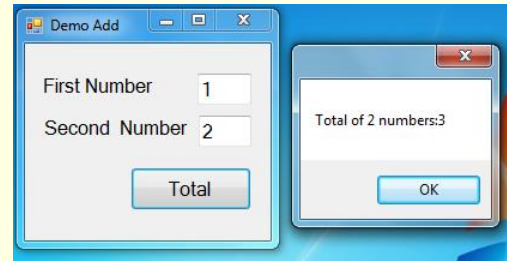


- ❑ Windows Form Applications



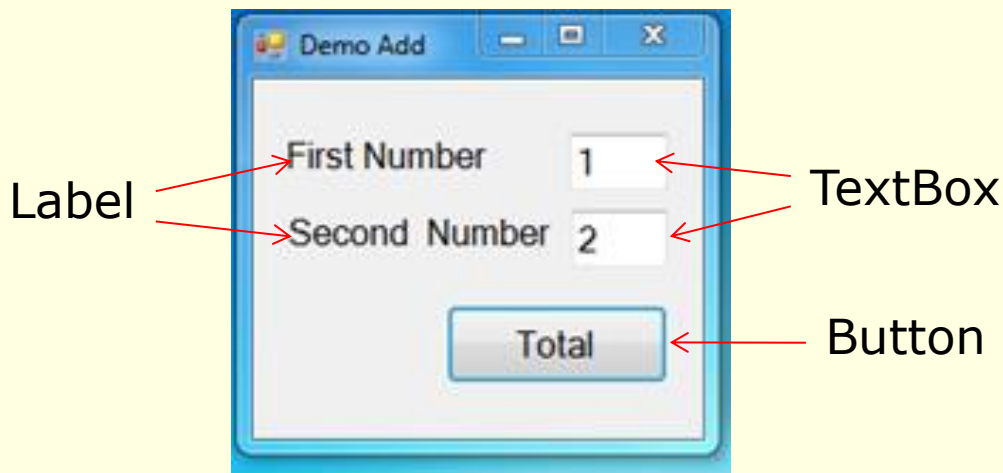
Introduction to the C# Language

- ❑ And many others such as mobile applications. So once you master C#, you will also do well in other programming modules!
- ❑ In this IT1753 Module, you will be using **Windows Form** to learn how to write programs using C#.
- ❑ **Why Windows Form?** It helps you to develop programs **quickly** that are **friendly, interactive** and **fun** (like games), without worrying about the details of how its **controls** (example, button, text box) are created.
- ❑ In this way, you can focus on mastering the C# Language!



A Word about Windows Form

- ❑ Let's examine Windows Forms in more detail



- ❑ The controls on the form above – Label, TextBox, Button and even the Form itself - are *objects*.
- ❑ But what are objects?

A Word about Windows Form

Illustration of the relationship between class and object



- ❑ In this iPhone example, an **iPhone Class** is a **template or blueprint** needed to make one or more iPhone **objects**. Classes have **properties** and **methods**.
 - ❑ Property refers to what it **has**, say, its color and dimensions (description)
 - ❑ Methods refer to what it **can do** say, dial a number, surf internet, answer a call etc. (actions)

A Word about Windows Form

Illustration of the relationship between class and object



- ❑ In the above example, from the IPhone **class**, we create 2 **Objects** – IPhone1 and IPhone2
- ❑ Once an object, we can change the value of properties and methods derived from the class.
- ❑ For example,



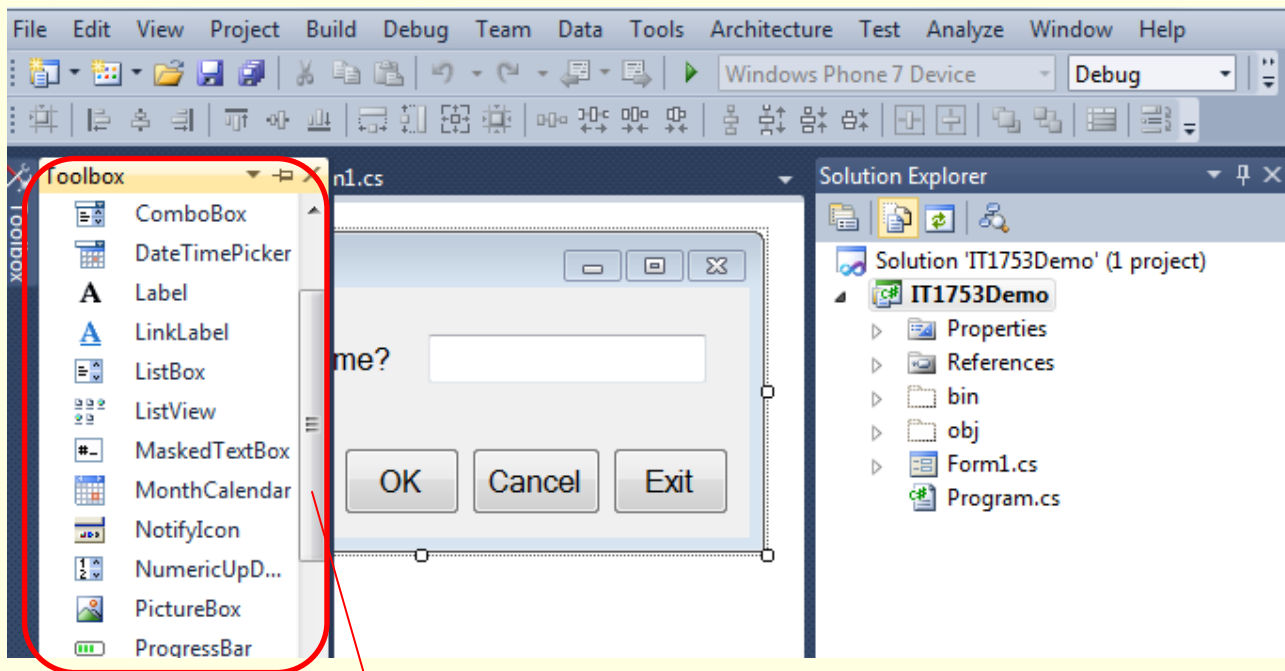
Color Property of Object IPhone1 is **Black**



Color Property of Object IPhone2 is **White**

A Word about Windows Form

- ❑ Similar to the **iPhone Class** example, each **control** in the **Toolbox** is a **class** you can use. The Toolbox is part of the **IDE** or Integrated Development Environment.

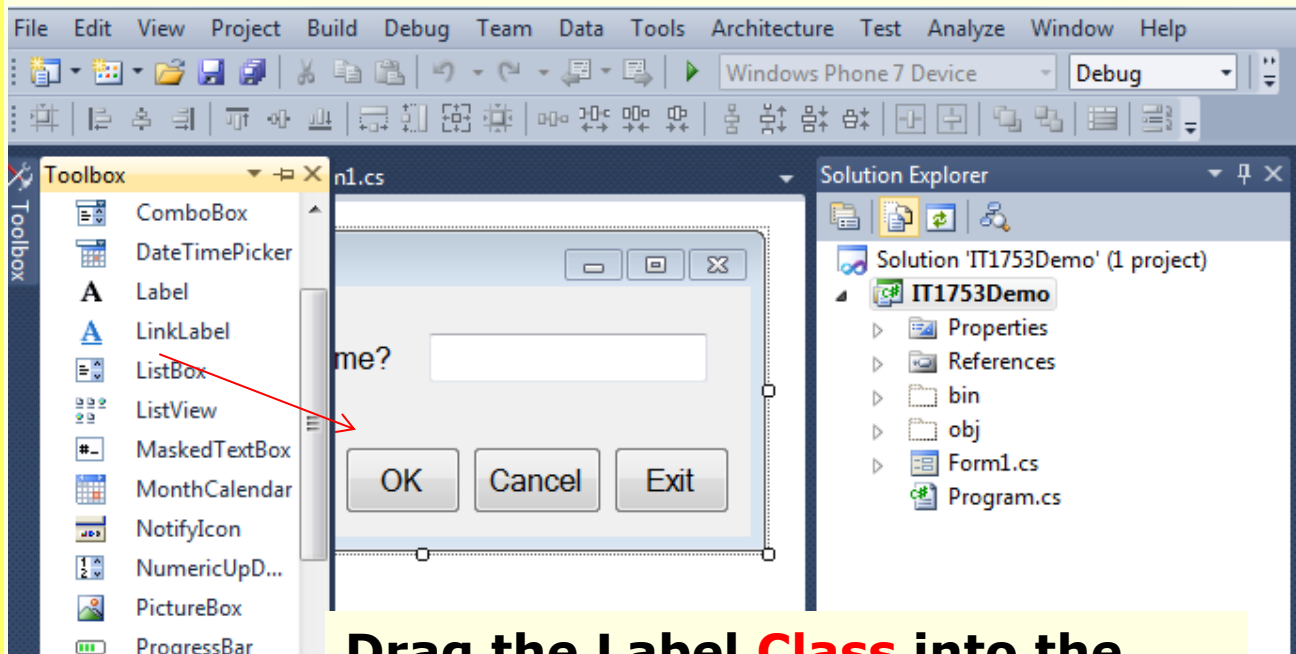


These are Control classes available for you to use

- ❑ As you can see from the Toolbox, you have a ComboBox class, DateTimePicker class, a Label class etc.

A Word about Windows Form

- ❑ How do you create an object from the control classes in the Toolbox?
- ❑ You just need to 'drag and drop' the control you want from the Toolbox to the Form!

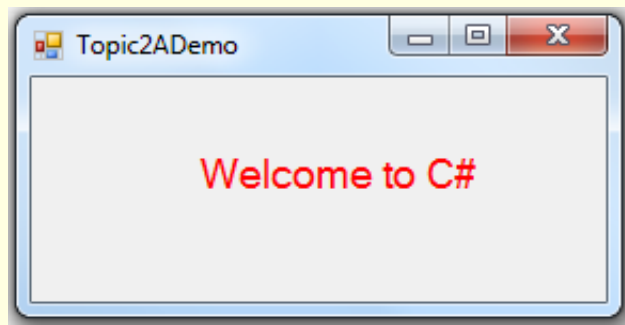


Drag the Label Class into the Form to create a Label Object.

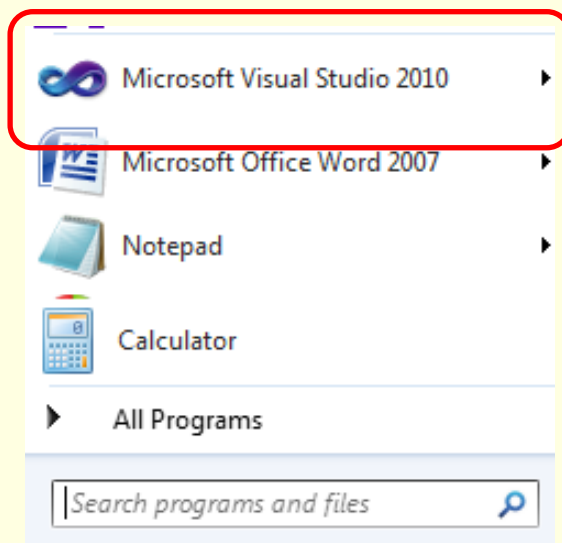
- ❑ Once the control is placed on the Form, the IDE will create automatically the necessary code for you to create the object; The basic C# coding are provided by VS.
- ❑ However, you may need to add in C# codes for your application. You will learn it soon!

An Example: Creating a Simple C# Windows Form

- Having understood what a control class is and how to create objects from them, let's go through a step by step example to create this form:

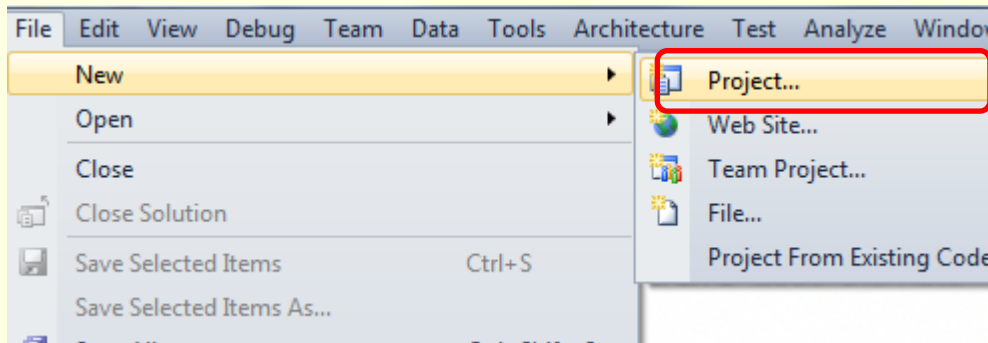


1. Click on Visual Studio to run the IDE environment.



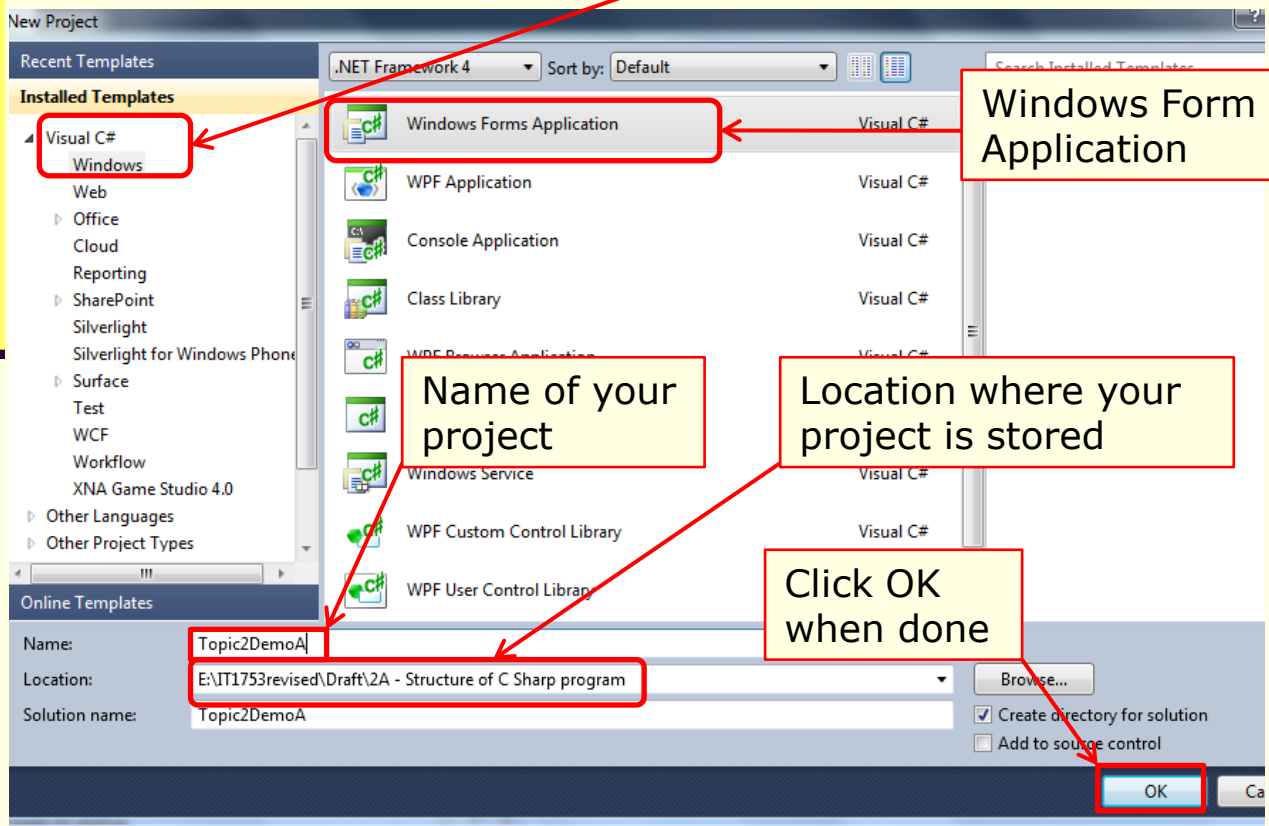
An Example: Creating a Simple C# Windows Form

2. Select File->New->Project.



3. Set the following:

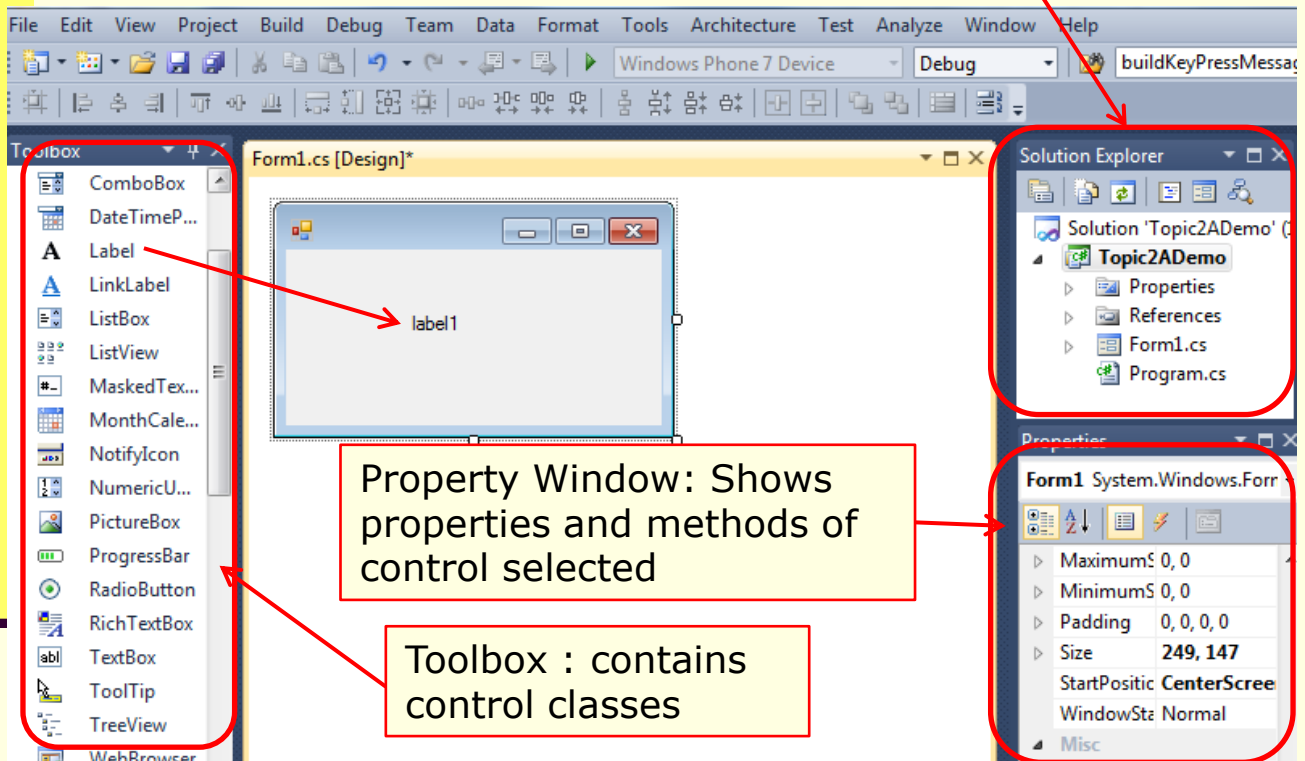
Choose C# and Windows



An Example: Creating a Simple C# Windows Form

4. The IDE that opens up comprises a Toolbox, Document Window, Solution Explorer, Property Window

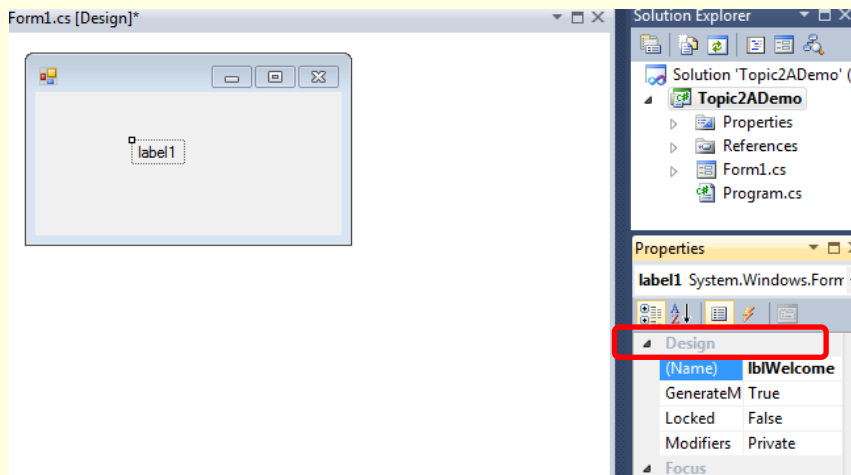
Solution Explorer: Shows all files in your project



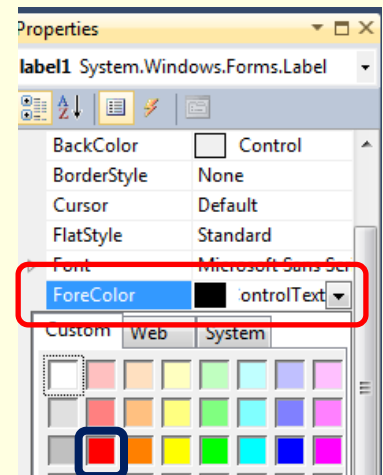
5. Drag and drop a Label control **class** from the Toolbox onto the Form1 to create a Label **object**.

An Example: Creating a Simple C# Windows Form

6. Change the Name property of the Label object to **lblWelcome**. It is a good practice to give your object meaningful names. In this, **lbl** refers to the prefix for Label and Welcome is because this Label is used to display a Welcome message.



7. Just for fun, change the ForeColour of the Label object to Red.



An Example:

Creating a Simple C# Windows Form


8. Double click on the Form using the mouse. The C# code will be shown. Don't worry about the rest of the code yet. Just look at the portion **Form1_Load**.

This is the method which executes when the Form first starts up (Remember a **method** is what an object can do?). In this case, the Form can perform a Load method called **Form1_Load**.

Fill up the line of text in *italics and red* within the **Form1_Load** method { } curly braces. Notice the line ends with a semicolon ;

```
namespace Topic2ADemo
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

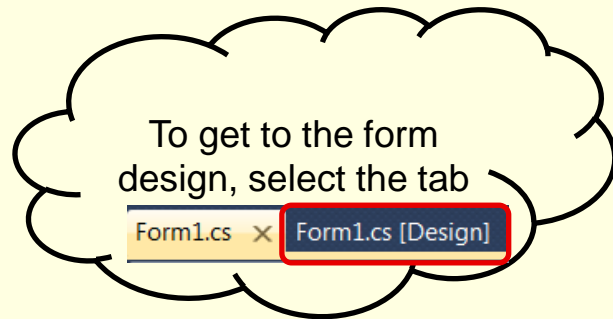
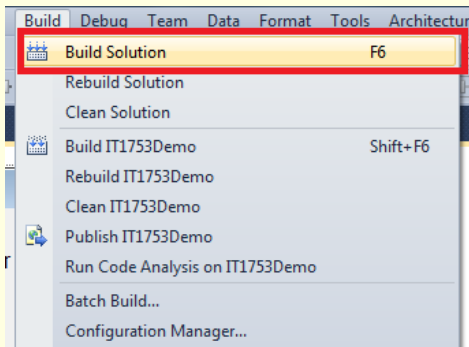
        private void Form1_Load(object sender,
            EventArgs e)
        {
            lblWelcome.Text = "Welcome to C#";
        }
    }
}
```



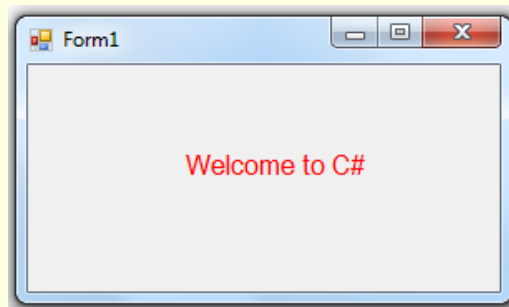
This means you are giving the Text property of Label lblWelcome the value "Welcome to C#"

An Example: Creating a Simple C# Windows Form

9. Compile the file by pressing **F6** or clicking on the highlighted menu item shown highlighted in red below. Notice if compilation is successful, a message *“Build successful”* is displayed on the bottom left.



10. Execute the file by pressing **F5**. The following Form will appear.



- ☐ Congratulations! You have successfully written a C# program with only 1 line of code!

An Example: Creating a Simple C# Windows Form

- Now that you have developed your first C# Form, let's look at the code in more detail:

```

namespace Topic2ADemo
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            // welcome statement
            lblWelcome.Text = "Welcome to C#";
        }
    }
}

```

Diagram annotations: A large red bracket on the left groups the opening and closing braces of the namespace, class, and Form1_Load method. A red circle with the number '1' is next to the first brace of the namespace. A red circle with the number '2' is next to the closing brace of the namespace. A red circle with the number '3' is next to the closing brace of the Form1_Load method.

- First, notice that for each section, the { } braces comes in pair – an opening and closing brace
- Namespace is just a way to organise information. You do not have to be too concern about this for now.
- Form1_Load is the method called when your Form first starts up. You write your C# codes inside methods like Form1_Load

An Example:

Creating a Simple C# Windows Form

- Now that you have developed your first C# Form, let's look at the code in more detail:

```
namespace Topic2ADemo
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent(); 6
        }

        private void Form1_Load(object sender,
            EventArgs e)
        {
            // welcome statement 5
            lblWelcome.Text = "Welcome to C#"; 4
        }
    }
}
```

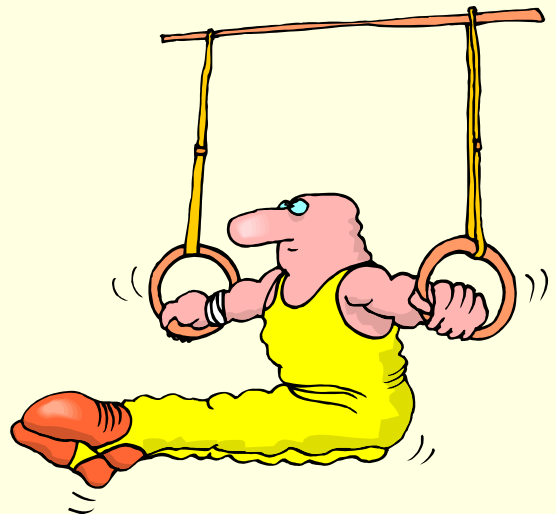
4 Note C# codes end with a semi colon;

5 This is a comment. Comments are ignored by the compiler

6 This line is automatically created by the IDE to create all necessary codes to get your program working

Summary

- ❑ Introduction to C# Language
- ❑ Understand the fundamentals of Windows Form
- ❑ How to create a C# project, compile and execute C# Windows Form application
- ❑ Do **PRACTICAL 2A**

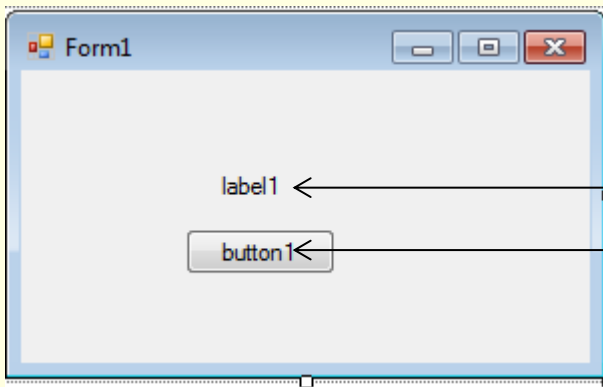


Practical 2A

Exercise 1:

Creating a Simple C# Windows Form

1. Using the steps given from Page 11 onwards, create a new Windows Form Project called Prac2AEx1.
2. Create a button and label object as shown.



Label:lblWelcome
Button:btnOK

3. Assign them the following properties:

Control	Property	Value
Button	Name	btnOK
Button	Text	OK
Label	Name	lblWelcome
Label	Text	<blank>

To get the **Property window**,
Right mouse click on the
object, choose



Practical 2A

Creating a Simple C# Windows Form

4. (a) Double click on the button.
(b) You will see a **btnOK_Click** method in the C# code. The button has a click method which will respond to the mouse click.
5. (a) When the button is click, we want to display a message “Welcome to C# Programming” in the Label we created earlier.
(b) To achieve this, in the btnOK_Click method, between the curly braces { }, enter the following sentence.

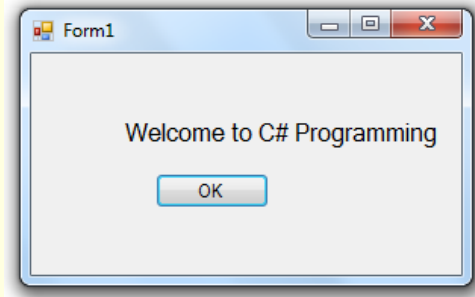
```
lblWelcome.Text = "Welcome to C# Programming";
```

- (c) Note there is a semicolon at the end of each C# sentence.
6. Compile your program by pressing F6. Make sure a message “Build Successful” is displayed at the bottom right corner of the IDE.

Practical 2A

Creating a Simple C# Windows Form

7. Run the program by pressing F5.
8. Test your program by pressing the OK button. The following should appear.



9. Congratulations! You have written a C# Windows Form program.

Practical 2A

Creating a Simple C# Windows Form

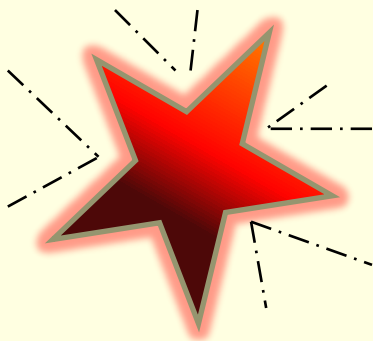
❑ Bonus exercise 1!!

❑ Modify the program to do the following:

❑ When user clicks OK button

❑ Program displays “Ah Meng,
Welcome to C#”

(Where Ah Meng is your name)



Practical 2A – C# Applications

Exercise 2 – GUI creation

We will be creating the C# GUI for all the practical questions:

- Refer: week 1 - Topic 1B, Practical 1B
- Based on the GUI design on paper
- Create the user controls in VS



For example:

Q1. Create a new application window project named as Pract2AEx2Q1. Based on the GUI you have designed, name your controls and create the GUI (using button, label and text box)

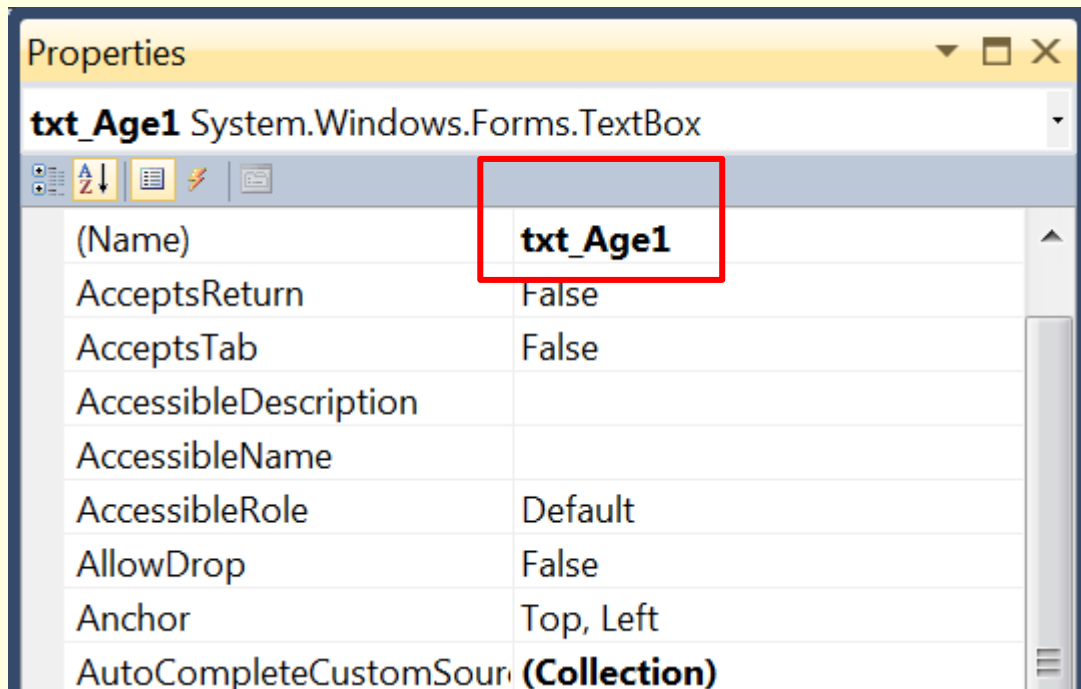
Design a program that prompts and reads the age of 2 different people, calculates the average age, and displays it.

Control	Name
label: lblAge1	TextBox: txt_Age1
label: lblAge2	TextBox: txtAge2
	Button: btnResult
label: lblAverage	label: lblResult

Practical 2A – C# Applications

Exercise 2 – GUI creation

Here is a screen shot on how to setting up the properties for **txt_Age1** text box control.



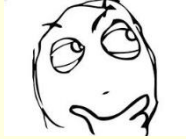
What are the Naming conventions for label, textfield and button? What is the difference in textfield and label??



Practical 2A – C# Applications

Exercise 2 – GUI creation

- ☐ You may test your application. Does it work?
Yes/No – Why??????????



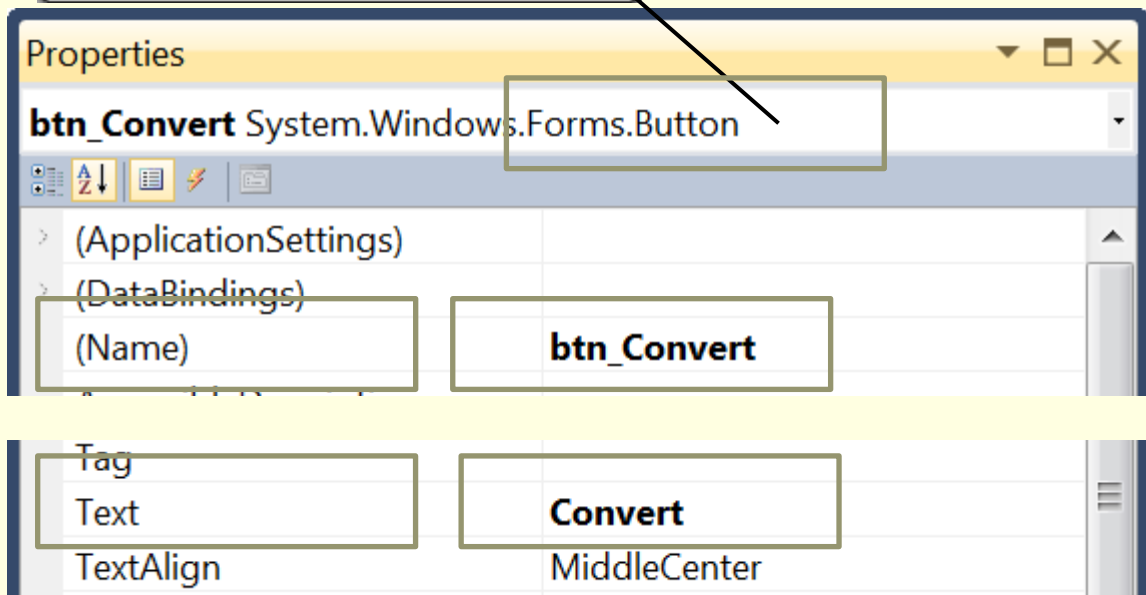
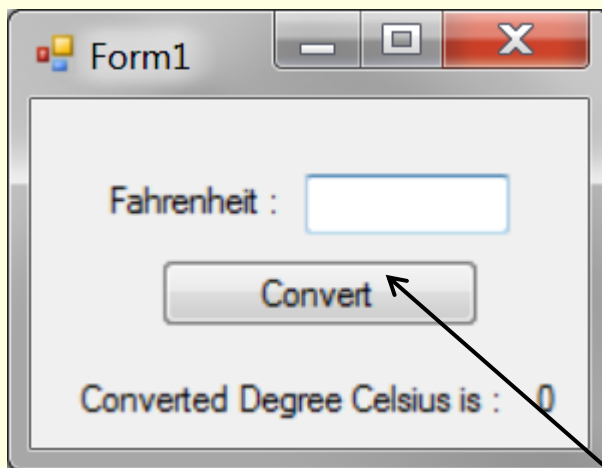
Complete ALL the GUI design in VS for all questions, Q2 to Q5 in Practical 1B

- ☐ Name the projects as:
 - ☐ Pract2AEx2Q2
 - ☐ Pract2AEx2Q3
 - ☐ Pract2AEx2Q4
 - ☐ Pract2AEx2Q5
- ☐ You are free to use your own names for the controls but must follow the naming convention:
 - ☐ Label – lbl****
 - ☐ Button – btn****
 - ☐ TextField – txt****
- ☐ **Very important → save all your files as you will need them in the next lessons.**
- ☐ The C# code will make your application
“ALIVE”

Practical 2A – C# Applications

Exercise 2 – GUI creation

- ❑ Screen design of Q2 GUI and how to setup control for **Convert button**
 - ❑ It is a BUTTON
 - ❑ Name of Control is btn_Convert, Text displayed on Control is Convert

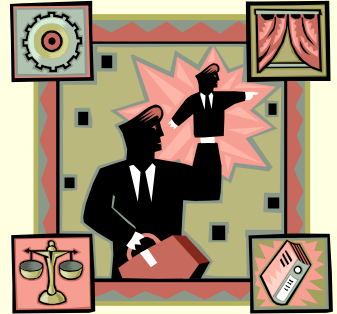


File Management

- ❑ You must be able to do the following:
 - ✓ Create a VS project
 - ✓ Save it.
 - ✓ Close it and open it again.
 - ✓ Compress the project and send it somewhere:
 - Send a copy to yourself or friend and see if you can open it.
 - Submit thru BB (BlackBoard) – refer to your tutor on this.

Note: if you cannot master this, you may spend a lot of time re-doing lost work or your future submissions may fail.

End of Topic 2A



Introduction to C# and Windows Form