

**Course:** EGDF20  
**Module:** EGE202 Application Programming

**SDL1:** Setting Up and Familiarization with Visual Studio IDE

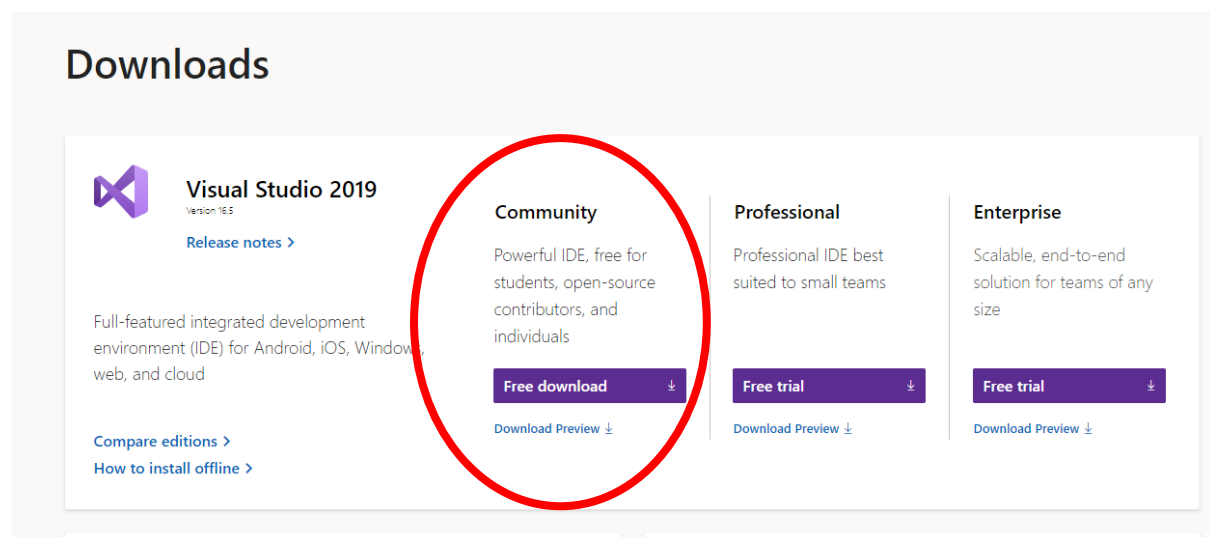
**Objectives:** At the end of this lab, the student should be familiar with procedure in installing Visual Studio and developing WinForm application.

## Part 1: Preparing Visual Studio IDE

### Exercise 1 –Setting Up Visual Studio

#### Step 1: Download Visual Studio Community

Website: <https://visualstudio.microsoft.com/downloads/>



#### Step 2: Install Visual Studio Community

- Run the installer file that you have downloaded
- Update the installer if required (you will receive a prompt)











Please update Visual Studio Installer before proceeding.

Update

- On **Workloads** tab, choose the required packages
  - .NET Desktop Development (For WinForm Application)

Installing — Visual Studio Community 2019 — 16.5.3

Workloads	Individual components	Language packs	Installation locations
<b>Web &amp; Cloud (4)</b>			
 <b>ASP.NET and web development</b> Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker support. <input checked="" type="checkbox"/>		 <b>Azure development</b> Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET Core and .NET... <input type="checkbox"/>	
 <b>Python development</b> Editing, debugging, interactive development and source control for Python. <input type="checkbox"/>		 <b>Node.js development</b> Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime. <input type="checkbox"/>	
<b>Desktop &amp; Mobile (5)</b>			
 <b>.NET desktop development</b> Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET Core and .NET... <input checked="" type="checkbox"/>		 <b>Desktop development with C++</b> Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild. <input type="checkbox"/>	
 <b>Universal Windows Platform development</b> Create applications for the Universal Windows Platform with C#, VB, or optionally C++. <input checked="" type="checkbox"/>		 <b>Mobile development with .NET</b> Build cross-platform applications for iOS, Android or Windows using Xamarin. <input type="checkbox"/>	

Location  
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community [Change...](#)

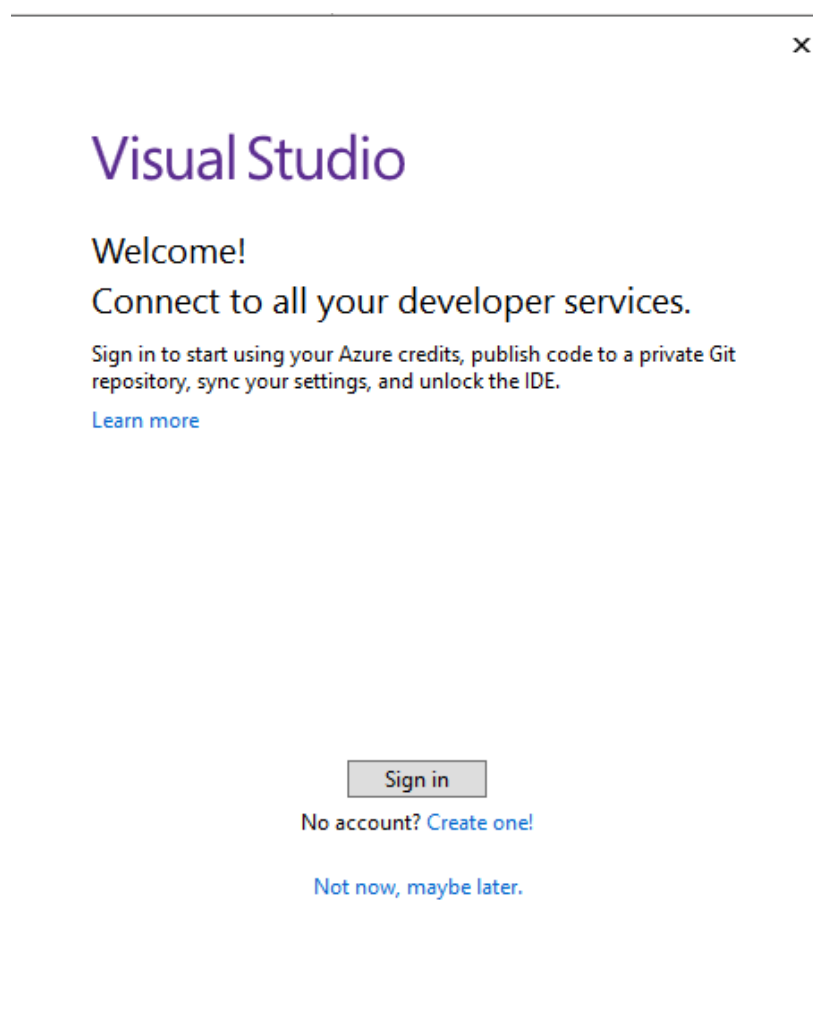
- Then click install to complete the installation

## Exercise 2 – Familiarizing with Visual Studio IDE

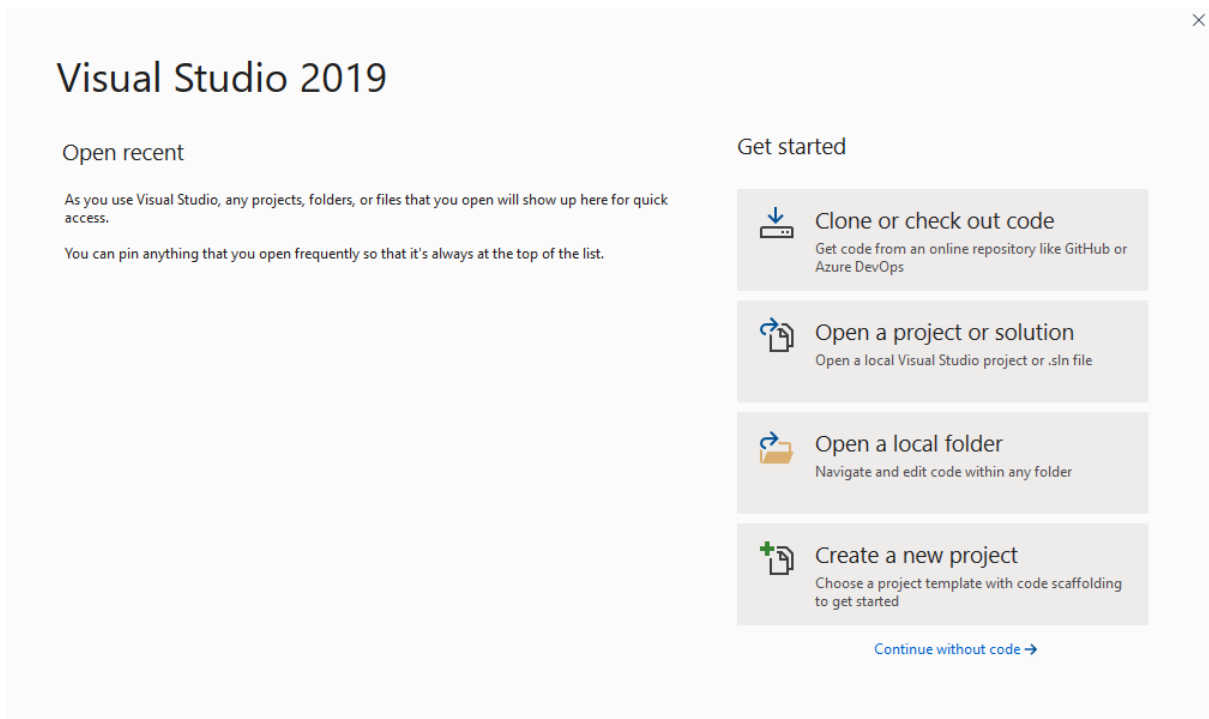
### Part 1: Running the Visual Studio IDE

1. Click on the *Window Start* menu in your Windows environment and search for **Visual Studio 2019**.
2. Click on the **Visual Studio 2019** icon to run the program.
3. If you are running Visual Studio for the first time, you are asked to sign in. **You can use any Outlook, Hotmail or NYP student email to sign in.**

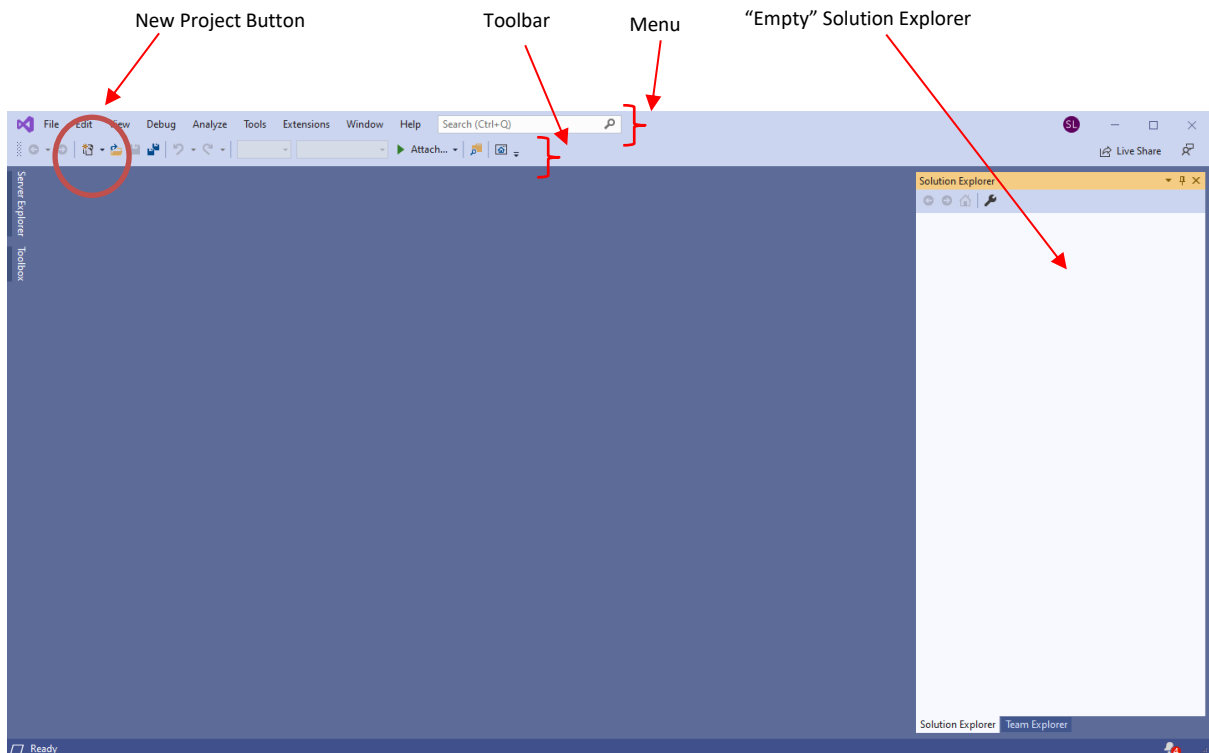
Optionally you can temporary skip this process by choosing “Not now, maybe later” option



4. After that, you should see a screen like the following screen:



5. For now, we will select **“Continue without code”**. Then you will be presented the following screen.



## Part 2: Creating a WinForm Application

1. From the menu select **File**, then click **New -> Project** to create a new project. Alternatively, you can use the **New Project** button.

**Create a new project**

Search for templates (Alt+S) Clear all

Recent project templates

A list of your recently accessed templates will be displayed here.

**C#** **Windows** **Desktop**

**Windows Forms App (.NET Framework)**  
A project for creating an application with a Windows Forms (WinForms) user interface  
**C#** **Windows** **Desktop**

**Next**

2. From the pop-up dialog, select **"C#"** for the Language filter, **"Windows"** for the Platform filter and **"Desktop"** for the Project type filter.
3. Then choose **Windows Form App (.NETFramework)** and click the **Next** button.

**Configure your new project**

**Windows Forms App (.NET Framework)** **C#** **Windows** **Desktop**

**Project name**  
Helloworld

**Location**  
C:\Users\liaws\Source\Repos

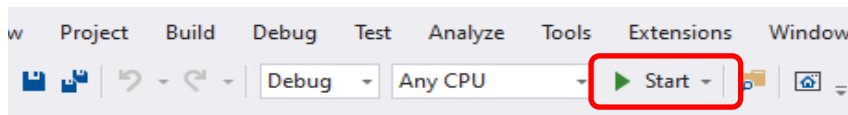
**Solution name** ⓘ  
Helloworld

☐ Place solution and project in the same directory

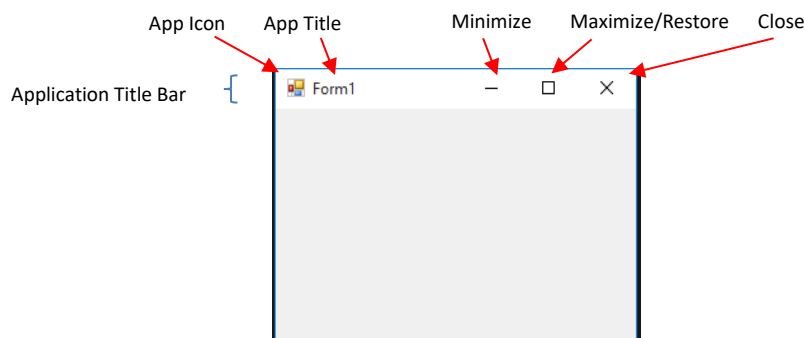
**Framework**  
.NET Framework 4.7.2

**Back** **Create**

4. Type the name of your new project as **Helloworld** and keep the Solution name the same as Project name.
5. Take note of the **Location** of where your project is created. You can either use the default or set the **Location** to point the project to your own created folder
6. **Do not** tick on the check-box of [ ☐ **Place solution and project in the same directory** ].
7. Click the **Create** button to and Visual Studio will start creating the required project files.
8. Once project is created click on the **Start** button on the Visual Studio menu to execute or run the application that we just created. (Alternatively press F5 or Ctrl+F5)



9. Make sure that the program execute correctly. You should see a Windows application running (See figure below). The application comes with a default application icon, title and the standard (minimize, maximize and close) buttons.



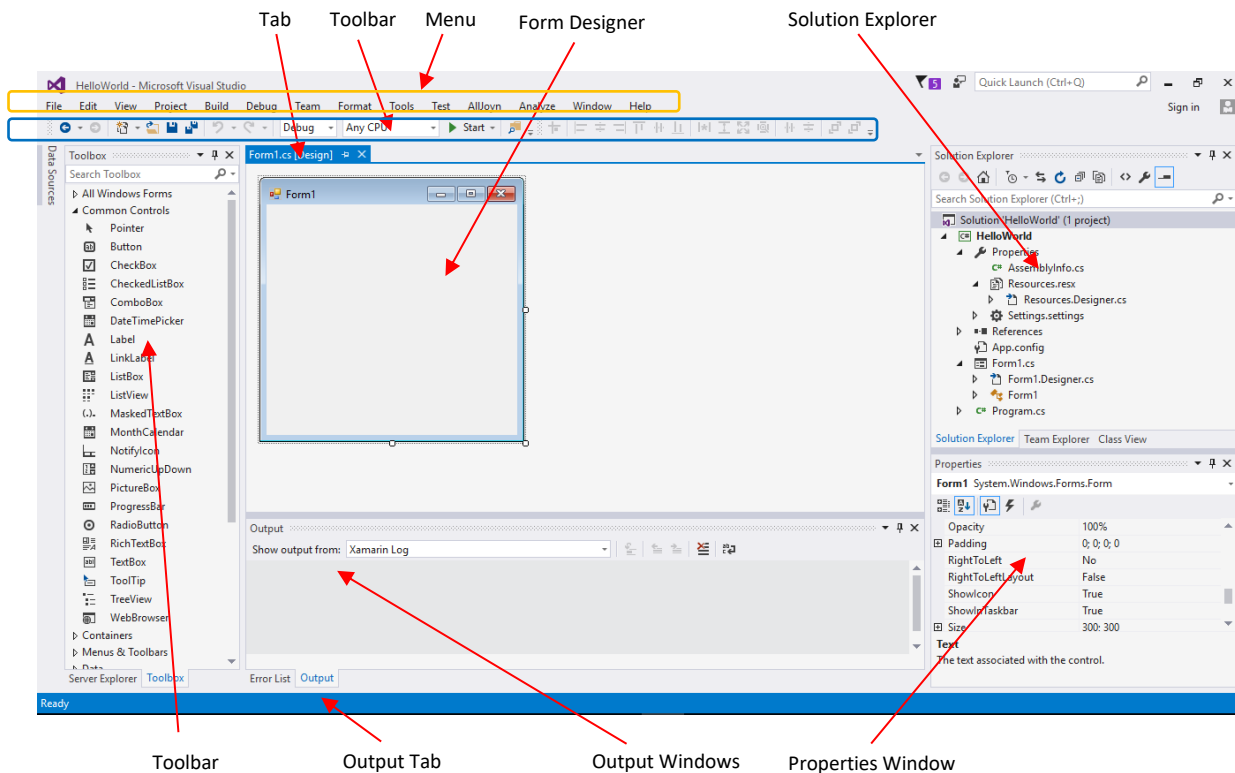
10. Try out the following actions and record down your observation.

No	Actions	Observation
1	Double Click on the Title Bar	
2	Right Click on the Title Bar	
3	Click and drag the Title Bar until it reaches the left or right side of the Windows *Snap feature	
4	Click on the Maximize button, observe and then click on the same button again	
5	Move the mouse to any 4 corners of the application. When the cursor starts to change, click and drag	
6	Using the <F5> or <Ctrl + F5>, run the application multiple times and observe the location of the application on screen	

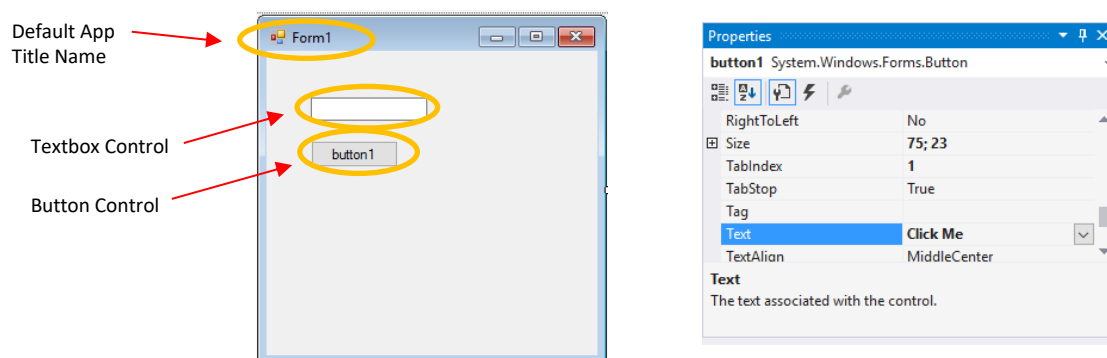
11. Under **File** menu, click **Close Solution** to close your project.

### Part 3: Understanding Visual Studio IDE (Integrated Development Environment)

1. Reopen the previous solution. Under the **File** menu, click **Open->Project/Solution**. Then navigate to the folder where the previous HelloWorld project is located. Choose the file "HelloWorld.sln" (Microsoft Visual Studio Solution).



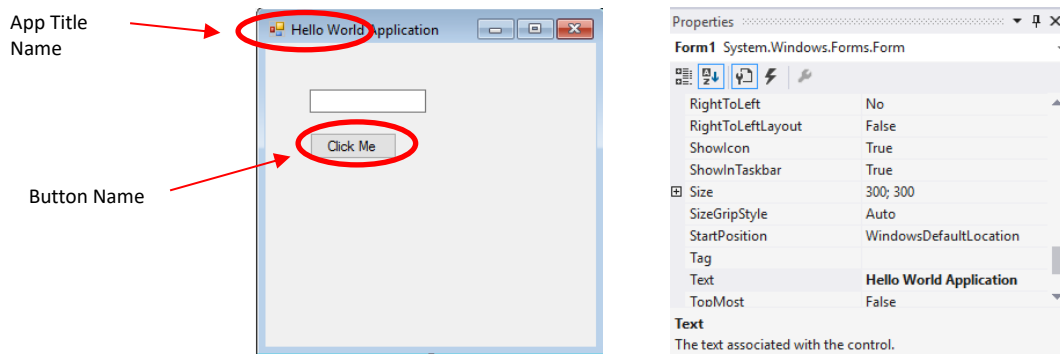
2. **Solution Explorer** window shows all the files used in the project solutions. Double click on "Form1.cs" will automatically bring you to the **Form Designer** tab. **Form Designer** is the tool to design the graphical user interface (GUI) of the application.
3. **Toolbar** window is a collection of controls which you can use to design your GUI. Drag a **TextBox** control and **Button** control onto the **Form1** window in the **Form Designer**. The name **Form1** is a default name that can be change. The **Form1** will look like the figure below:



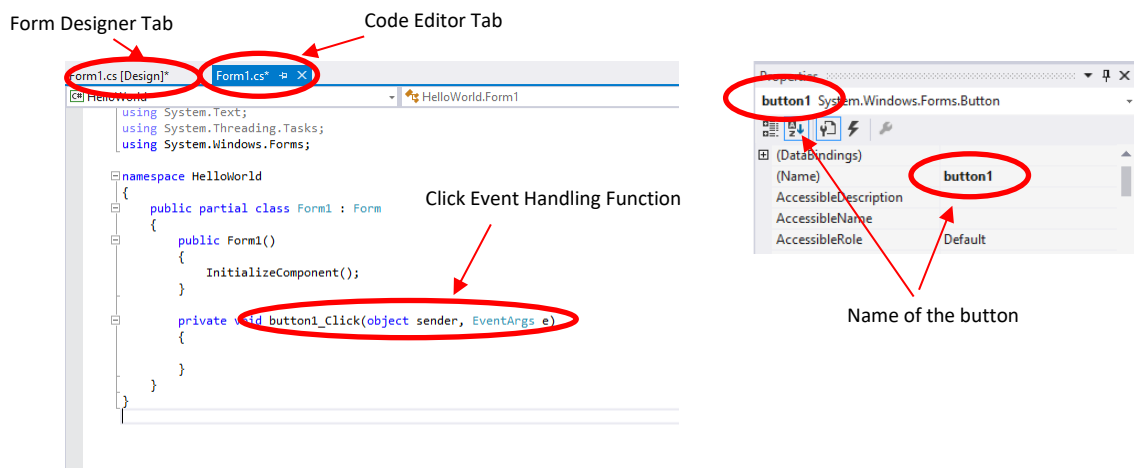
4. Select the **Button** control in the **Form Designer**. On the **Properties** window, the properties of the selected **Button** control will automatically be shown. Modify the context of the

Text property from “button1” to “Click Me”. Observe that the **Button** control will now show the word “Click Me”.

5. Select the **Form** control in the **Form Designer**. In the **Properties** window of the **Form** control, change the **TopMost** property of the **Form1** to ‘True’.
6. On the **Properties** window, the properties of the selected **Form** control will automatically be shown. Modify the context of the **Text** property from “Form1” to “Hello World Application”. Observe that the application title will now show the word “Hello World Application”.

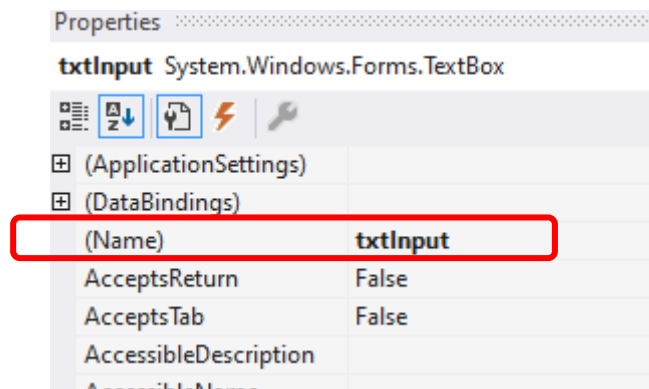


7. Double click on the “Click Me” **Button** in the **Form Designer**. That will automatically open the “Form1.cs” file. The event handling function `button1_Click (...)` is automatically appended to the file. The default naming convention for the function is `{control name}_{event type} (...)`





8. Select the **Textbox** control in the **Form Designer**. On the **Properties** window, the properties of the selected **Textbox** control will automatically be shown. Modify the content of the **(Name)** property from “textBox1” to “txtInput”.



9. At the code editor for “Form1.cs” modify **button1\_Click (...)** to include the following codes:

```
private void button1_Click(object sender, EventArgs e)
{
    string str;
    str = txtInput.Text;
    str = "Hello World " + str;
    Console.WriteLine(str);
}
```

10. Build and debug your application by hitting <F5> key or from the menu **Debug->Start Debugging**. Type in some words in the Textbox and click the “Click Me” Button. **Hint:** Observe the Output window. You must ensure the output tab is selected. (If output tab not is not visible, from the menu select **Debug -> Windows -> Output**).

