**EXPERIMENT NO: 08**

**Title**: Design a Windows Form based application for different controls.

**Aim:** Study of different types of windows form controls.

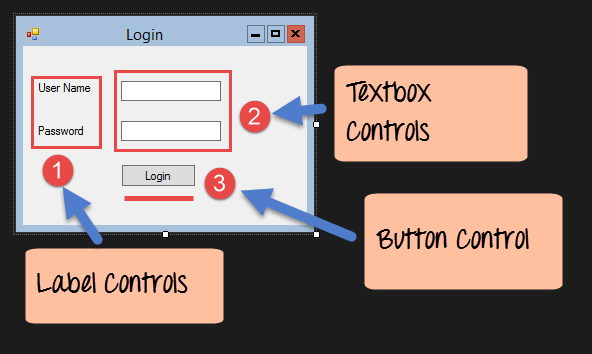
**Theory:**

**Windows Forms Basics:**

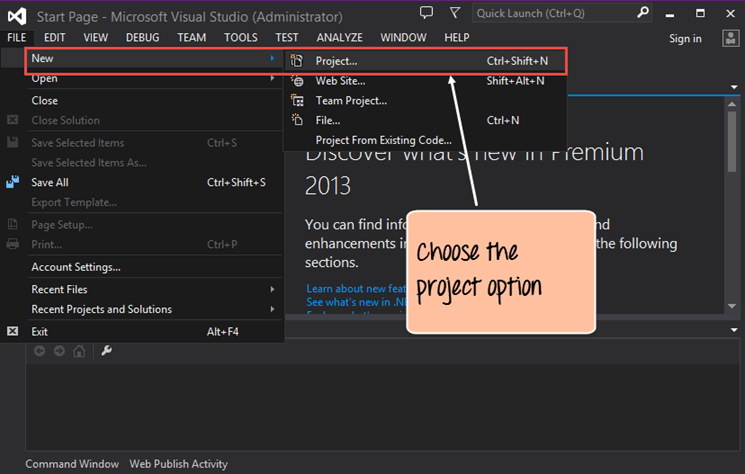
A windows form application is an application, which is designed to run on a computer. It will not run on web browser because then it becomes a web application.

A Windows forms application is one that runs on the desktop computer. A Windows forms application will normally have a collection of controls such as labels, textboxes, list boxes, etc.

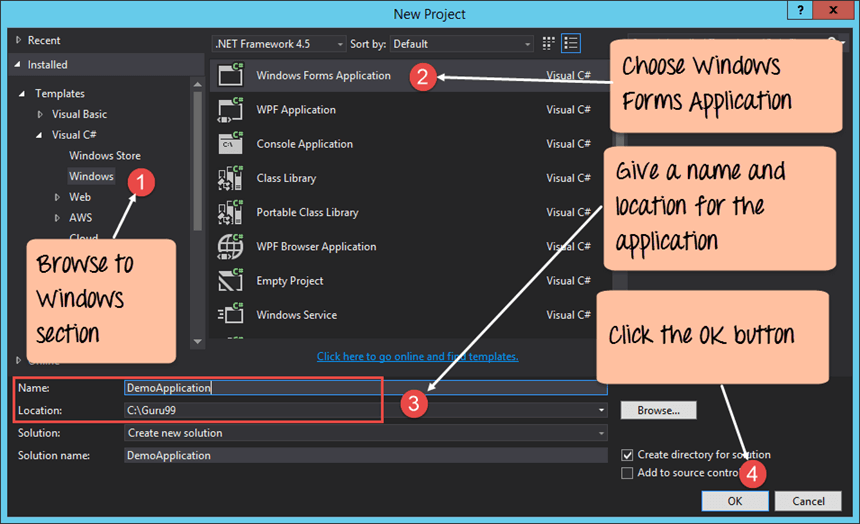
Below is an example of a simple Windows form application C#. It shows a simple Login screen, which is accessible by the user. The user will enter the required credentials and then will click the Login button to proceed.



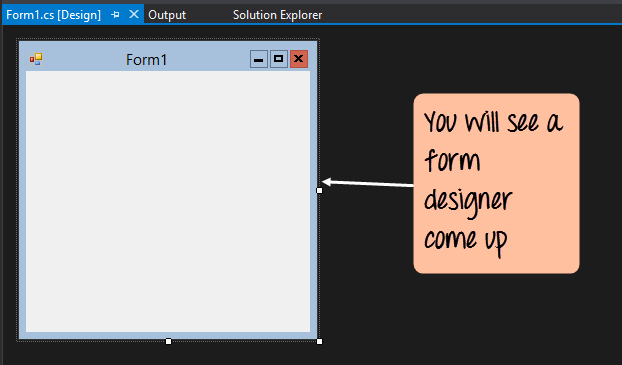
Step 1) The first step involves the creation of a new project in Visual Studio. After launching Visual Studio, you need to choose the menu option New->Project.



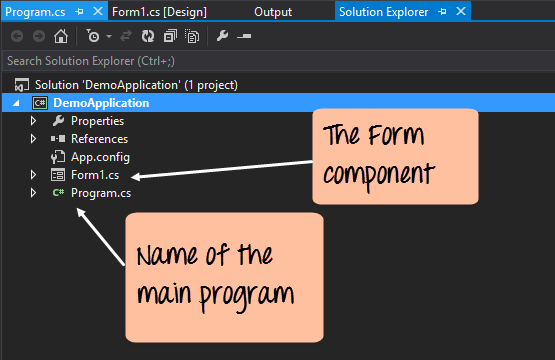
Step 2) The next step is to choose the project type as a Windows Forms application. Here we also need to mention the name and location of our project.



**Output:-**



It’s in this Form Designer that you will start building your Windows Forms application.

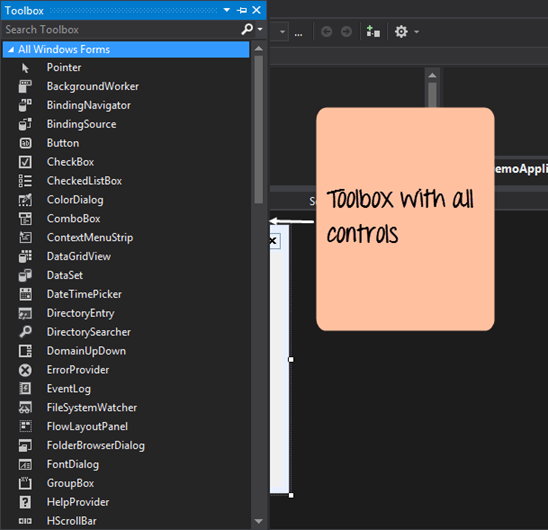


In the Solution Explorer, you will also be able to see the DemoApplication Solution. This solution will contain the below 2 project files

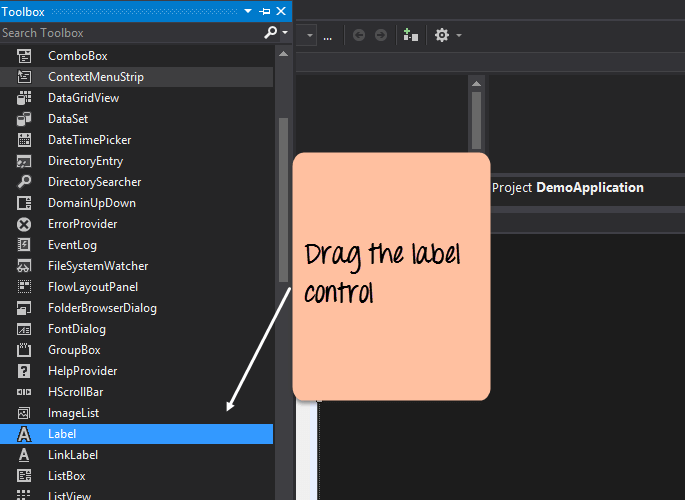
1. A Form application called Forms1.cs. This file will contain all of the code for the Windows Form application.
2. The Main program called Program.cs is default code file which is created when a new application is created in Visual Studio. This code will contain the startup code for the application as a whole.

On the left-hand side of Visual Studio, you will also see a ToolBox. The toolbox contains all the controls which can be added to a Windows Forms. Controls like a text box or a label are just some of the controls which can be added to a Windows Forms.

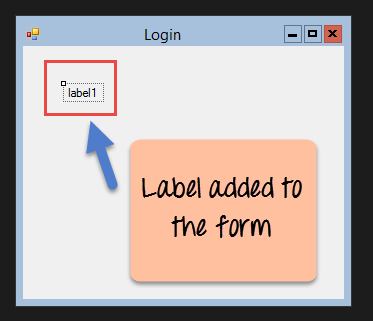
Below is a screenshot of how the Toolbox looks like.



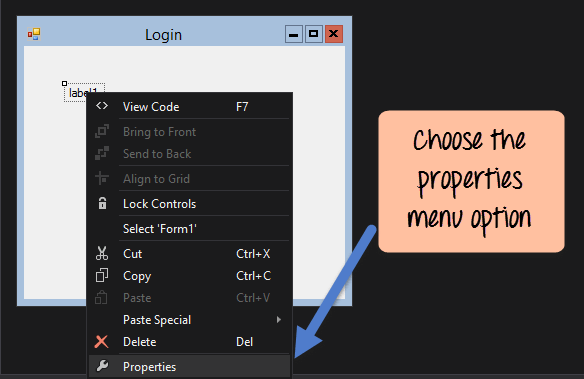
Step 3) In this step, we will now add a label to the Form which will display “Hello World.” From the toolbox, you will need to choose the Label control and simply drag it onto the Form.

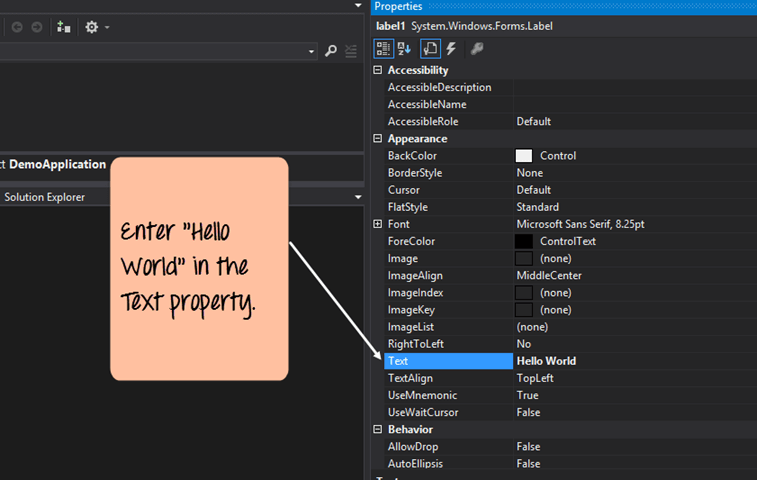


Once you drag the label to the form, you can see the label embedded on the form as shown below.

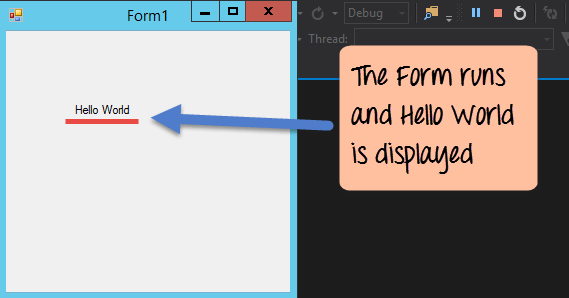


Step 4) The next step is to go to the properties of the control and Change the text to ‘Hello World’.





**Output:-**

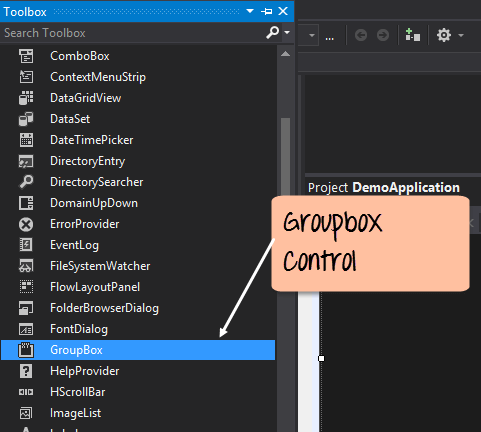


**Adding Controls to a form:**

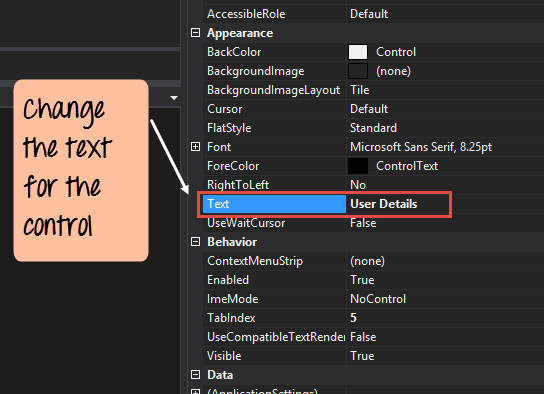
**Group Box**

A group box is used for logical grouping controls into a section.

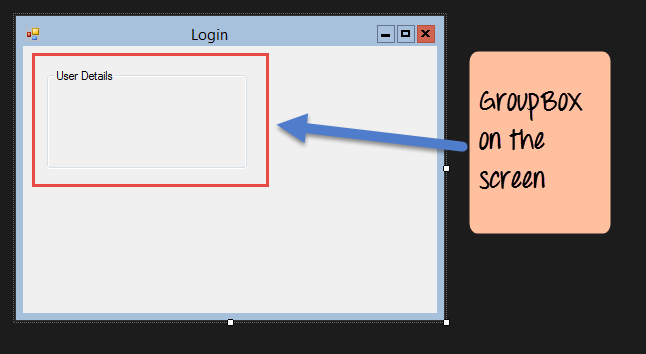
Step 1) The first step is to drag the Groupbox control onto the Windows Form from the toolbox as shown below



Step 2) Once the groupbox has been added, go to the properties window by clicking on the groupbox control. In the properties window, go to the Text property and change it to “User Details”.

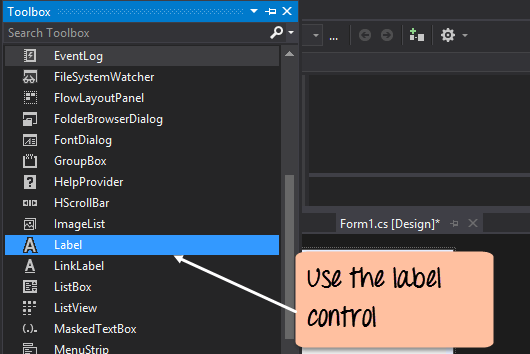


**Output:-**

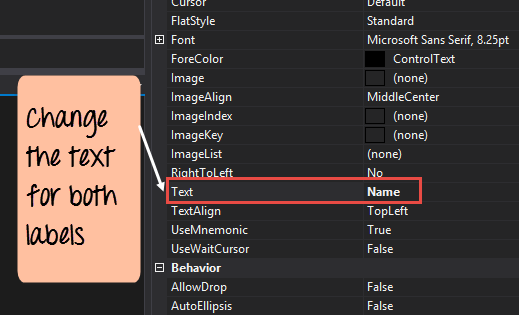


**Label Control**

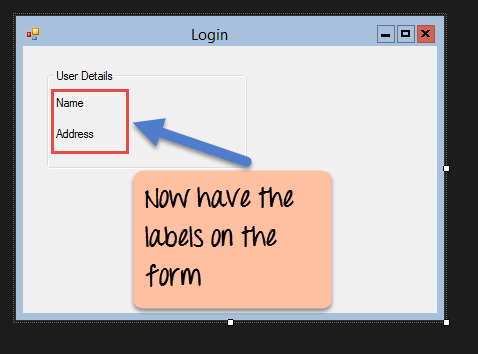
Step 1) The first step is to drag the label control on to the Windows Form from the toolbox as shown below. Make sure you drag the label control 2 times so that you can have one for the ‘name’ and the other for the ‘address’.



Step 2) Once the label has been added, go to the properties window by clicking on the label control. In the properties window, go to the Text property of each label control.



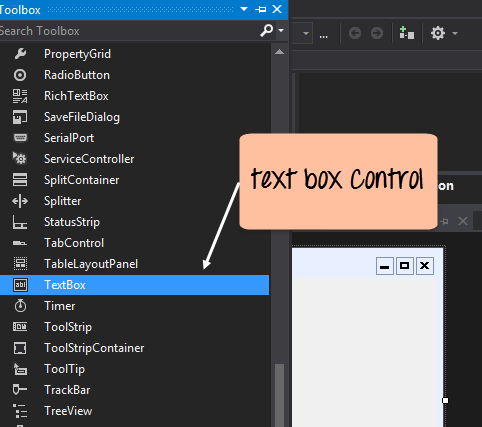
**Output:-**



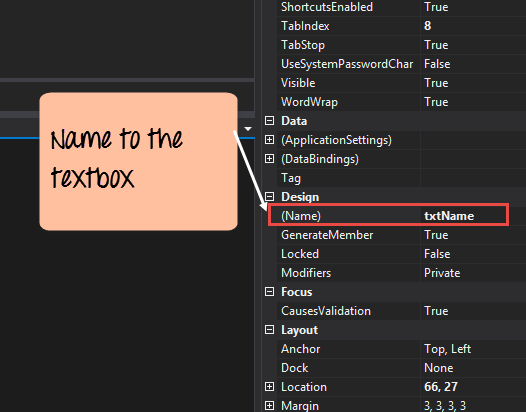
**Textbox:**

A textbox is used for allowing a user to enter some text on the Windows application in C#.

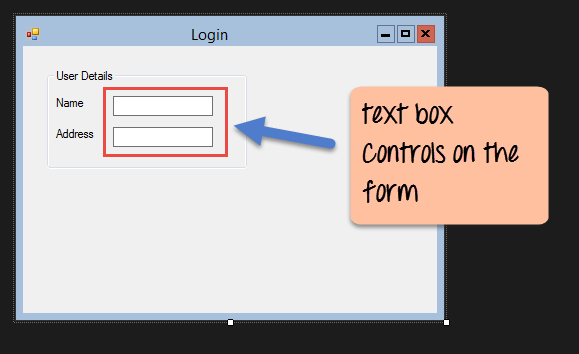
Step 1) The first step is to drag the textbox control onto the Windows Form from the toolbox as shown below



Step 2) Once the text boxes have been added, go to the properties window by clicking on the textbox control. In the properties window, go to the Name property and add a meaningful name to each textbox. For example, name the textbox for the user as txtName and that for the address as txtAddress. A naming convention and standard should be made for controls because it becomes easier to add extra functionality to these controls, which we will see later on.



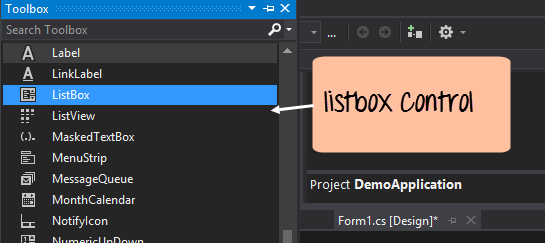
**Output:-**



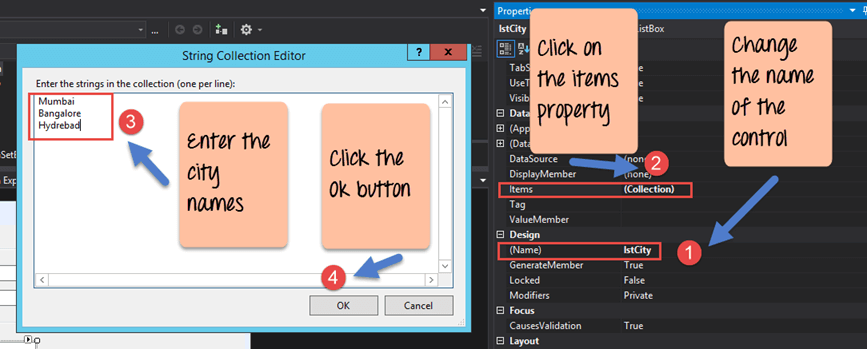
List box

A Listbox is used to showcase a list of items on the Windows form.

Step 1) The first step is to drag the list box control onto the Windows Form from the toolbox as shown below



Step 2) Once the list box has been added, go to the properties window by clicking on the list box control.



**C# Windows Forms Application:**

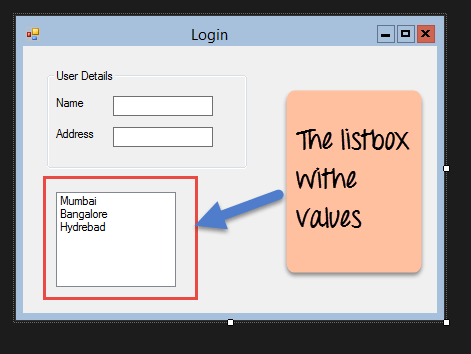
First, change the property of the Listbox box control, in our case, we have changed this to lstCity

Click on the Items property. This will allow you to add different items which can show up in the list box. In our case, we have selected items “collection”.

In the String Collection Editor, which pops up, enter the city names. In our case, we have entered “Mumbai”, “Bangalore” and “Hyderabad”.

Finally, click on the ‘OK’ button.

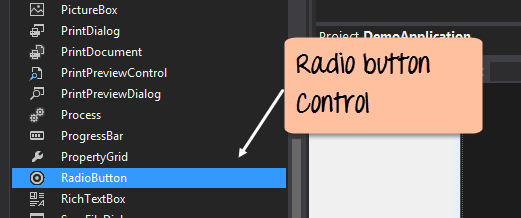
**Output:-**



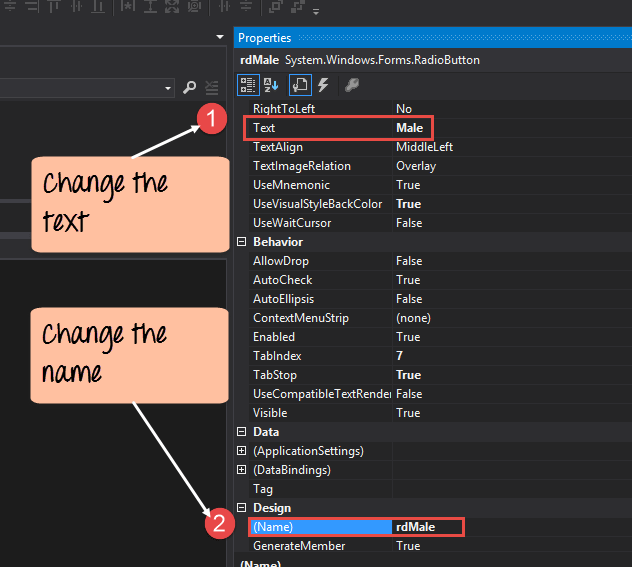
**RadioButton:**

A Radiobutton is used to showcase a list of items out of which the user can choose one.

Step 1) The first step is to drag the ‘radiobutton’ control onto the Windows Form from the toolbox as shown below.



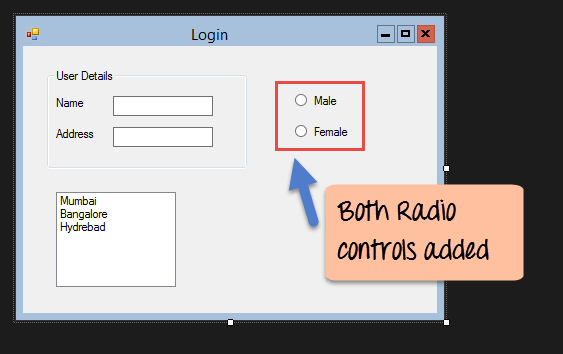
Step 2) Once the Radiobutton has been added, go to the properties window by clicking on the Radiobutton control.



First, you need to change the text property of both Radio controls. Go the properties windows and change the text to a male of one radiobutton and the text of the other to female.

Similarly, change the name property of both Radio controls. Go the properties windows and change the name to ‘rdMale’ of one radiobutton and to ‘rdfemale’ for the other one.

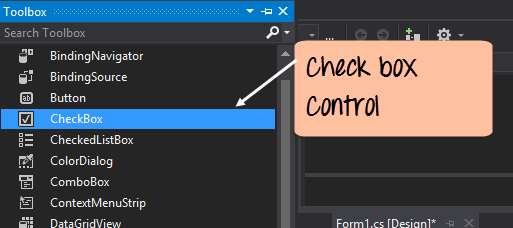
**Output:-**



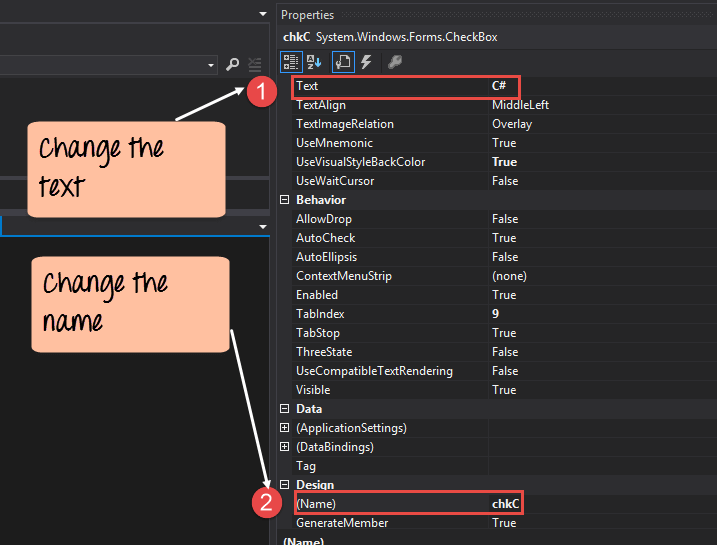
**Checkbox:**

A checkbox is used to provide a list of options in which the user can choose multiple choices.

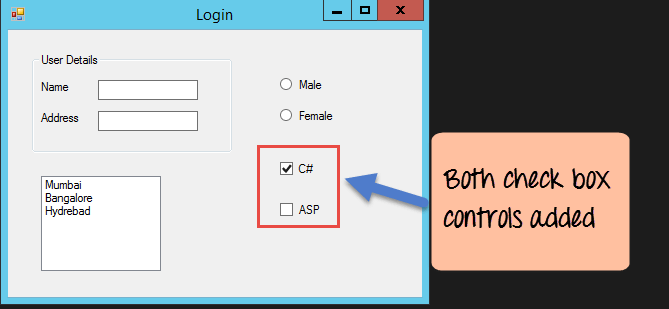
Step 1) The first step is to drag the checkbox control onto the Windows Form from the toolbox as shown below



Step 2) Once the checkbox has been added, go to the properties window by clicking on the Checkbox control.



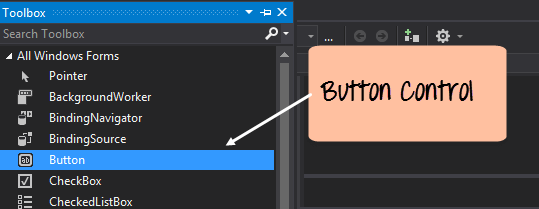
**Output: -**



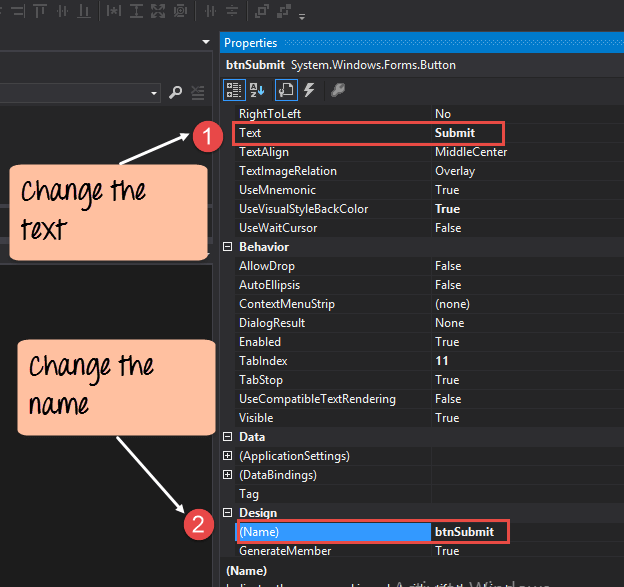
**Button:**

A button is used to allow the user to click on a button which would then start the processing of the form.

Step 1) The first step is to drag the button control onto the Windows Form from the toolbox as shown below



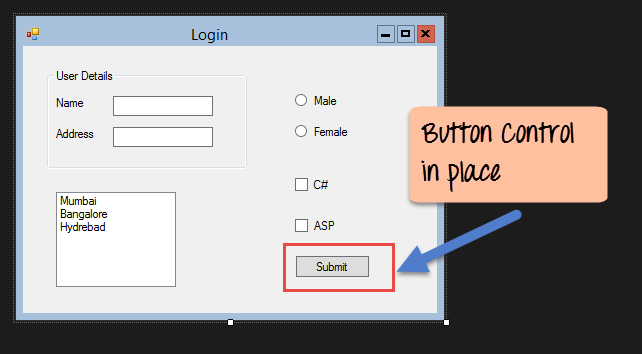
Step 2) Once the Button has been added, go to the properties window by clicking on the Button control.



First, you need to change the text property of the button control. Go the properties windows and change the text to ‘submit’.

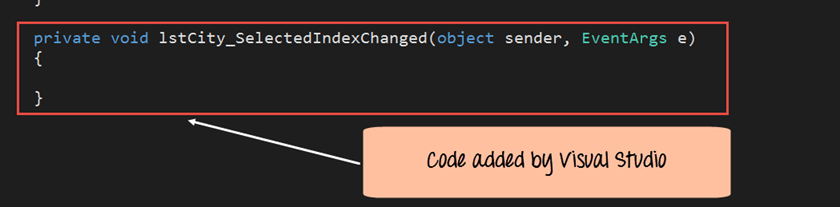
Similarly, change the name property of the control. Go the properties windows and change the name to ‘btnSubmit’.

**Output:-**

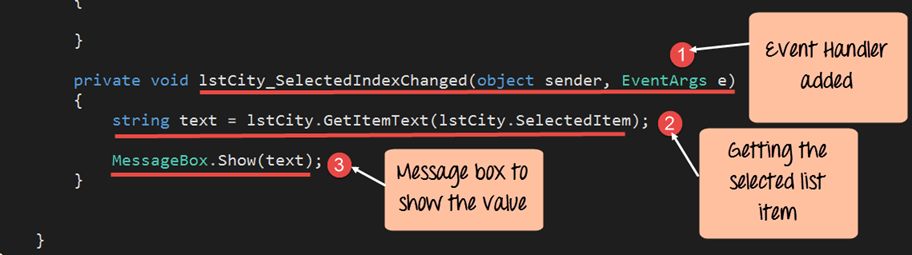


**C# Event Handling for Controls:**

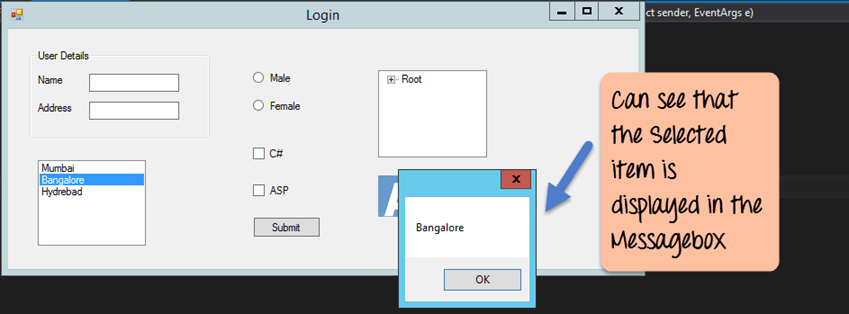
Step 1) Double click on the Listbox in the form designer. By doing this, Visual Studio will automatically open up the code file for the form. And it will automatically add an event method to the code. This event method will be triggered, whenever any item in the listbox is selected.



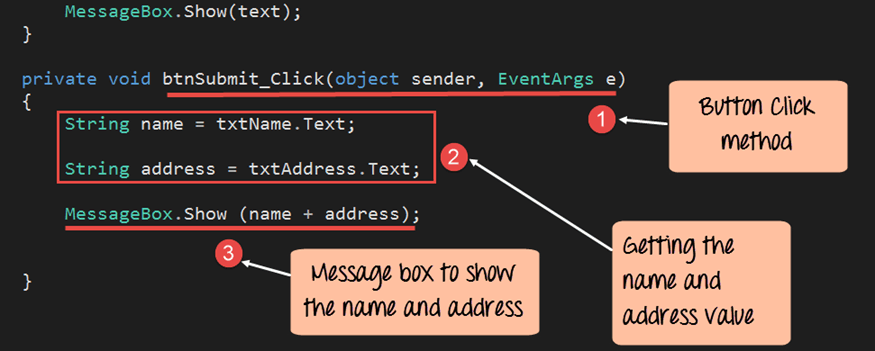
Now let’s add the below section of code to this snippet of code, to add the required functionality to the listbox event.



Output:-



Step 2) Double click on the Button in the form designer.

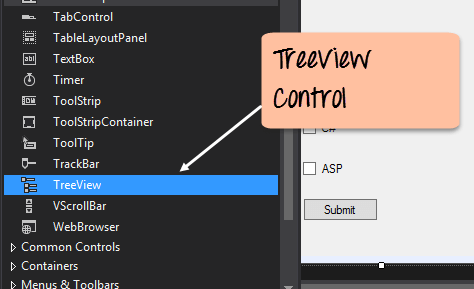


**Tree and PictureBox Control:**

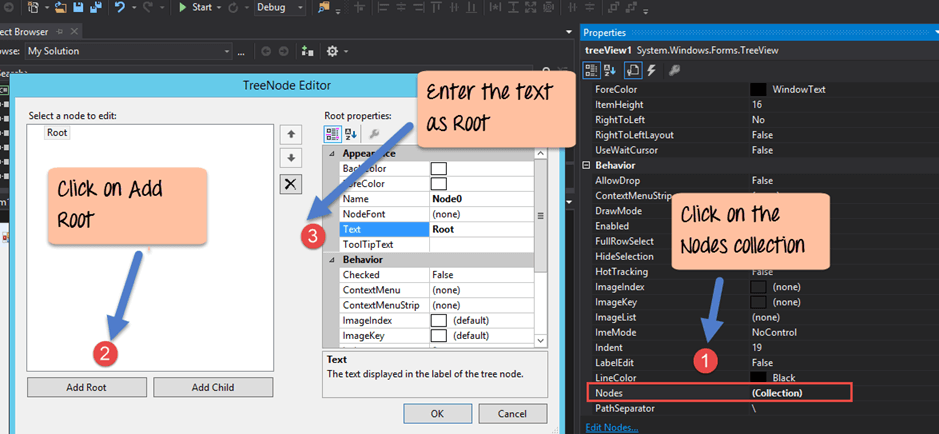
**Tree Control:**

– The tree control is used to list down items in a tree like fashion.

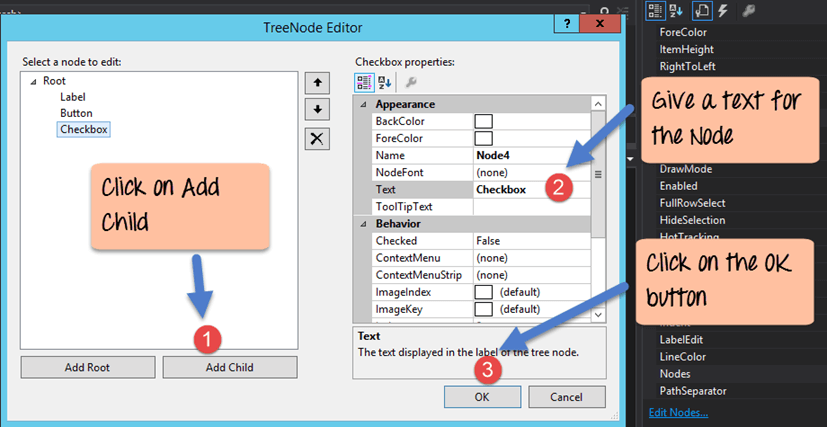
Step 1) The first step is to drag the Tree control onto the Windows Form from the toolbox as shown below



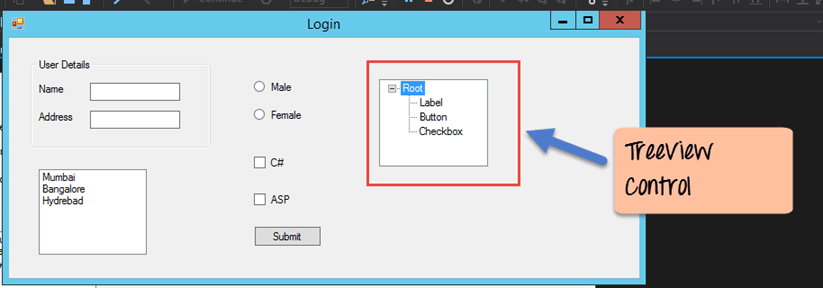
Step 2) The next step is to start adding nodes to the tree collection so that it can come up in the tree accordingly. First, let’s follow the below sub-steps to add a root node to the tree collection.



Step 3) The next step is to start adding the child nodes to the tree collection. Let’s follow the below sub-steps to add child root node to the tree collection.



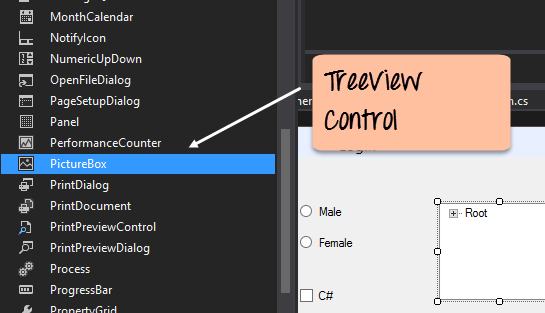
**Output:-**



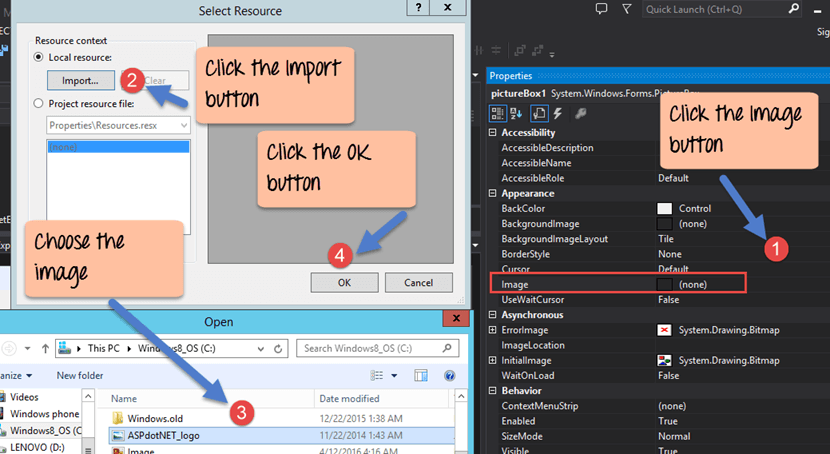
**PictureBox Control**

This control is used to add images to the Winforms C#.

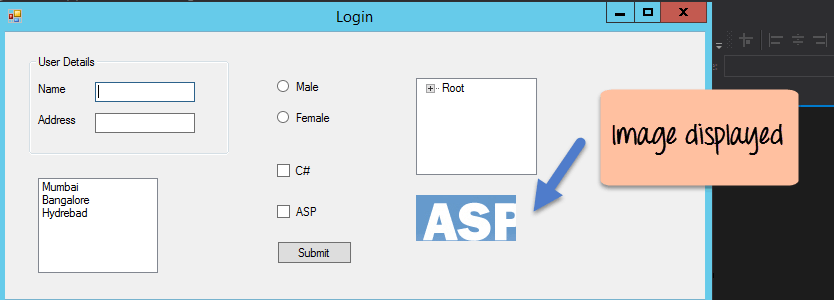
Step 1) The first step is to drag the PictureBox control onto the C# Windows Form from the toolbox as shown below



Step 2) The next step is to actually attach an image to the picture box control. This can be done by following the below steps.



Output:-



**Problem Statement:**

Design Windows Form Application for Student Registration.

**Conclusion:**

Here we have studied different types of windows form controls.

**Sample Questions:**

1) Explain various Windows Form Control?