Lab 1

Objective: To get familiar Event driven Programming and GUI components

Event Driven Programming:

Event-driven programming is a programming paradigm in which the flow of program execution is determined by *events* - for example a user action such as a mouse click, key press, or a message from the operating system or another program. An event-driven application is designed to detect events as they occur, and then deal with them using an appropriate *event-handling procedure*. The idea is an extension of *interrupt-driven programming* of the kind found in early command-line environments such as DOS, and in embedded systems (where the application is implemented as firmware).

Event-driven programs can be written in any programming language, although some languages(Visual Basic for example) are specifically designed to facilitate event-driven programming, and provide an *integrated development environment* (IDE) that partially automates the production of code, and provides a comprehensive selection of built-in objects and controls, each of which can respond to a range of events. Virtually all object-oriented and visual languages support event-driven programming. Visual Basic, Visual C++, Java and C# are examples of such languages.

A visual programming IDE such as C# provides much of the code for detecting events automatically when a new application is created. The programmer can therefore concentrate on issues such as interface design, which involves adding controls such as command buttons, text boxes, and labels to standard *forms* (a form represents an application's workspace or *window*). Once the user interface is substantially complete, the programmer can add event-handling code to each control as required.

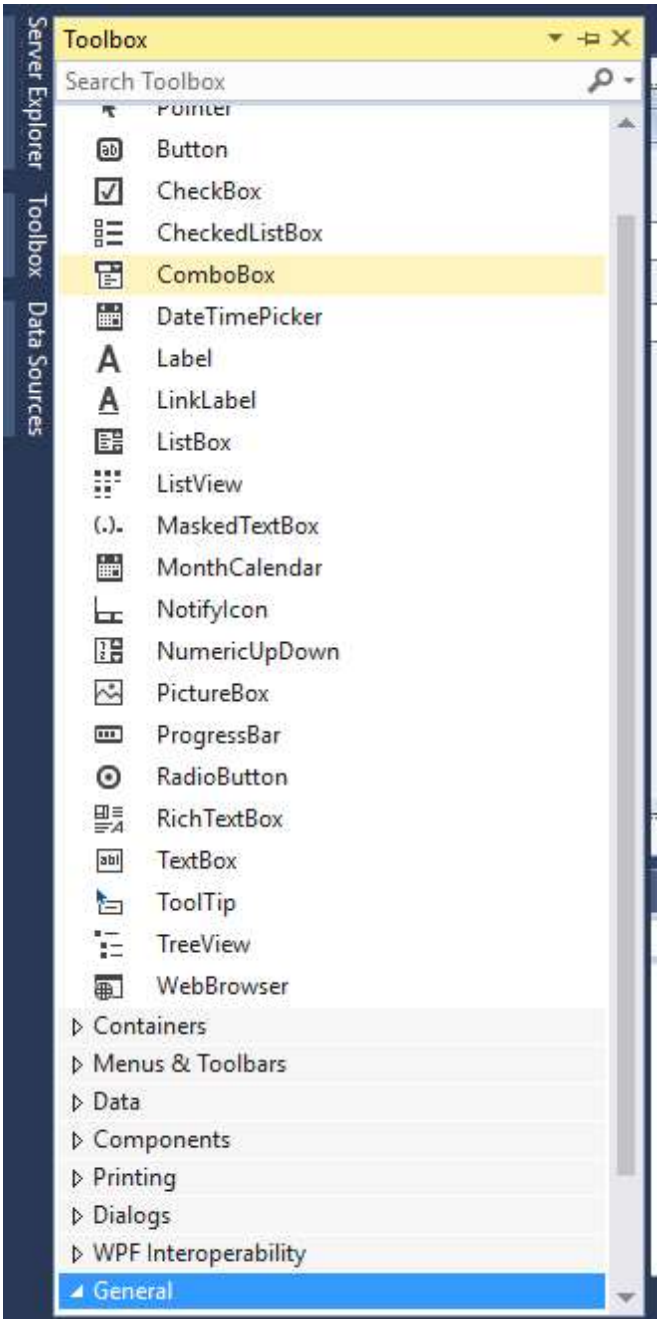
Integrated Development Environment:

An integrated development environment (IDE) is a software suite that consolidates basic tools required to write and test software.

Developers use numerous tools throughout software code creation, building and testing. Development tools often include text editors, code libraries, compilers and test platforms. Without an IDE, a developer must select, deploy, integrate and manage all of these tools separately. An IDE brings many of those development-related tools together as a single framework, application or service. The integrated toolset is designed to simplify software development and can identify and minimize coding mistakes and typos.

Framework:

1. GUI Components:



2. GUI control and their properties

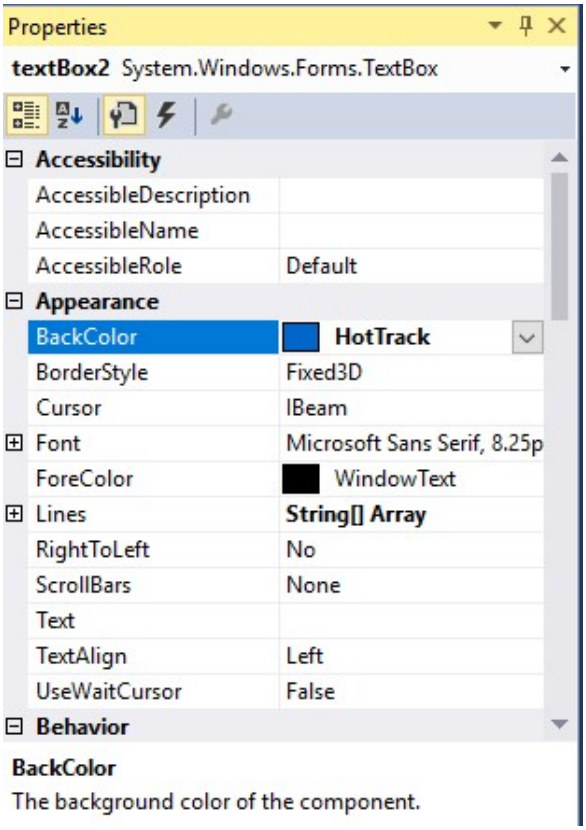


Figure 1 Text box properties

You can see component properties by double click on it.

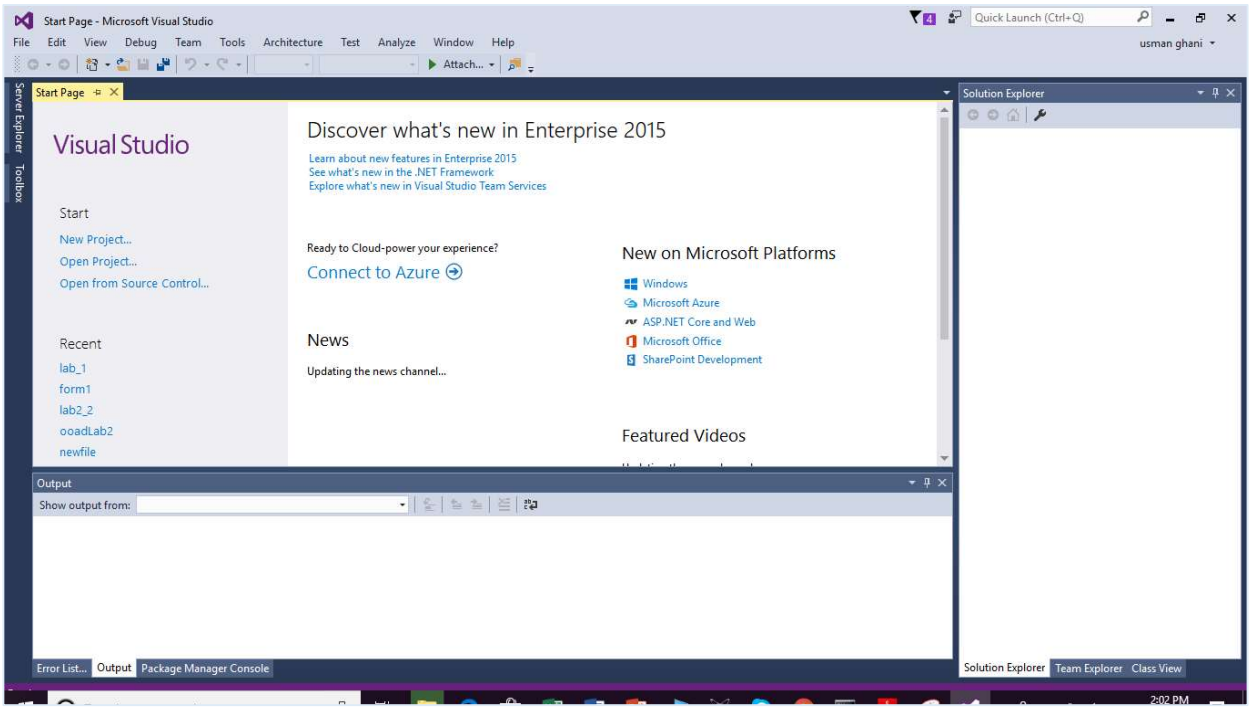
Events:

An event can be mouse click, key press, or a message from the operating system or another program.

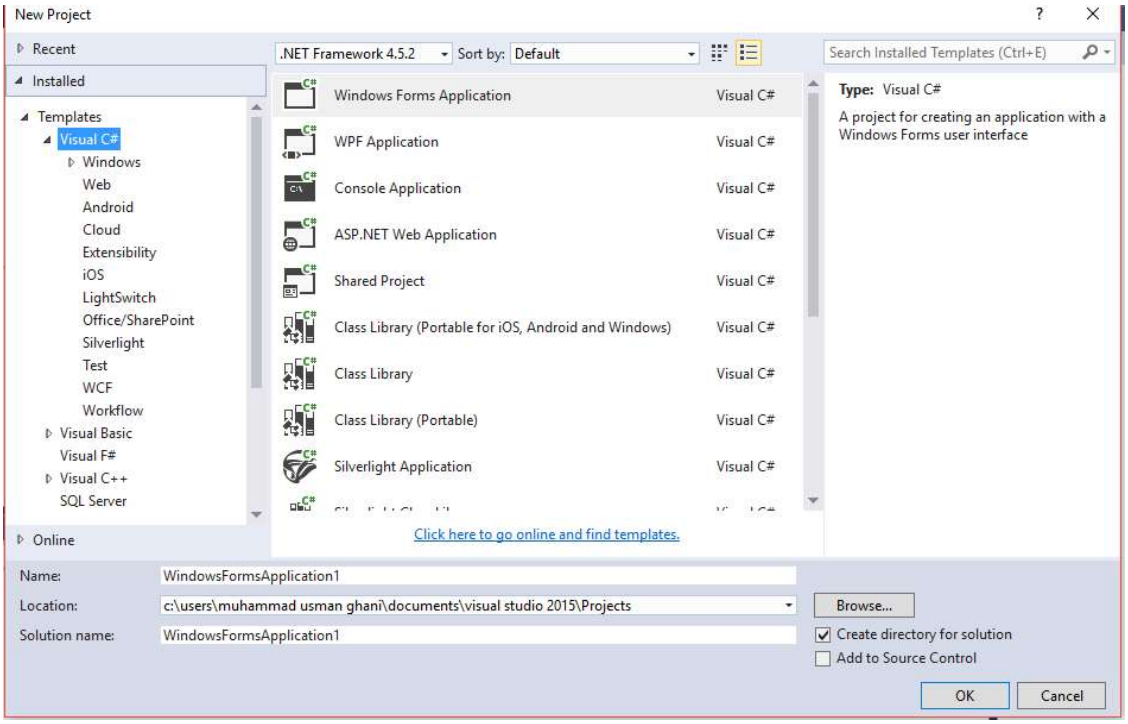
1. Form load event

What event will occur when the form is loaded? Suppose we created form and we want to see "**welcome**" when the form is loaded(form load is an event). We double click on the form to see event function; which code we write here will execute when event is occurred.

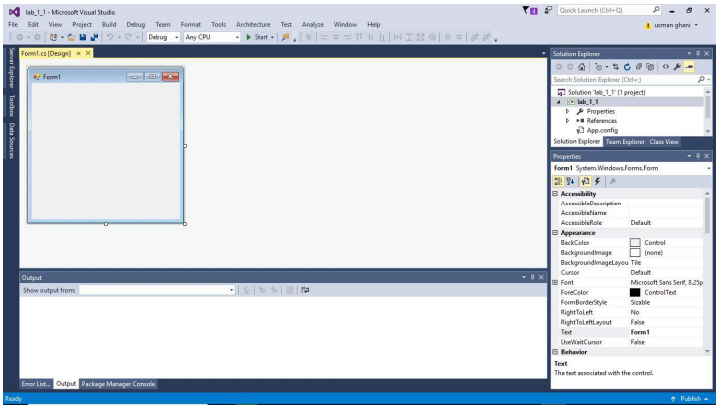
Step 1: open visual studio and click on new project



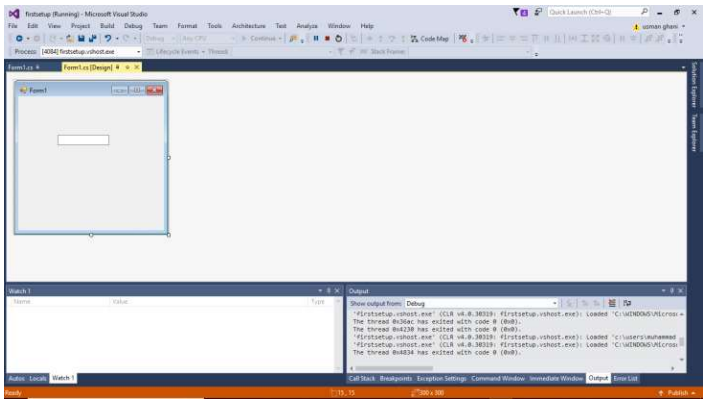
Step 2: select visual C#, write project name and the select windows form application.



Step 3: Form Creation

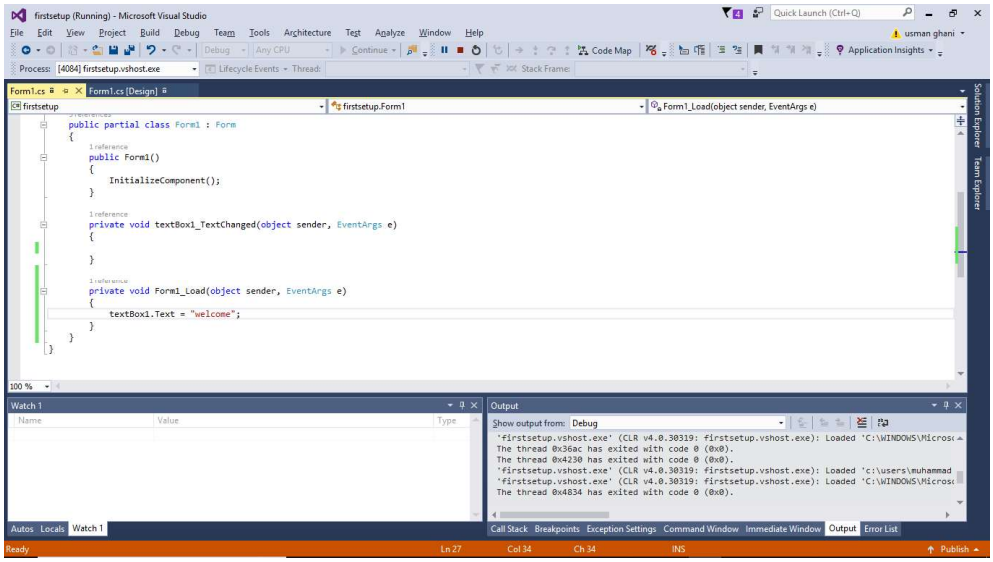


Step4: Select text box from toolbox in the left window



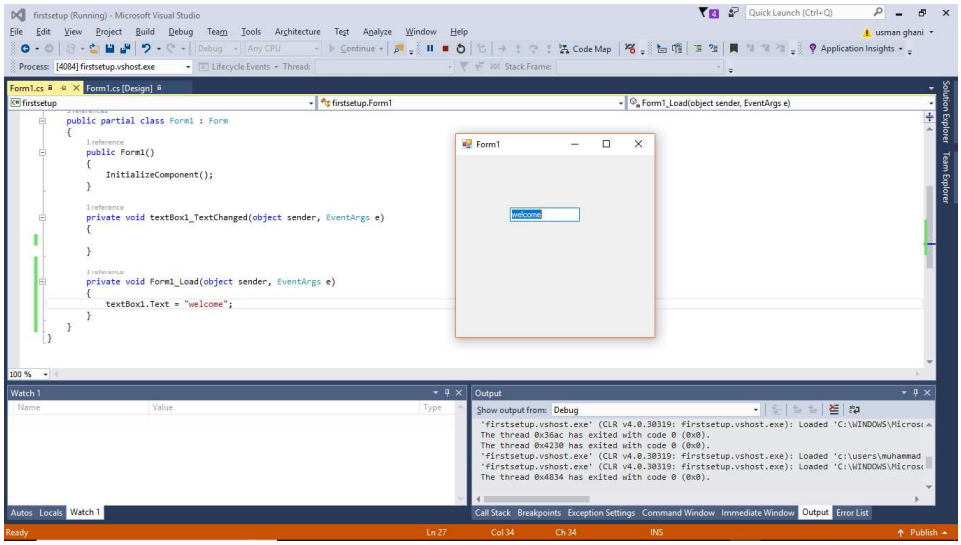
Step 2:

Double click on the form to see “formload” event function



Here we write a code so that when form is loaded welcome show in the text box on the form.

Output:

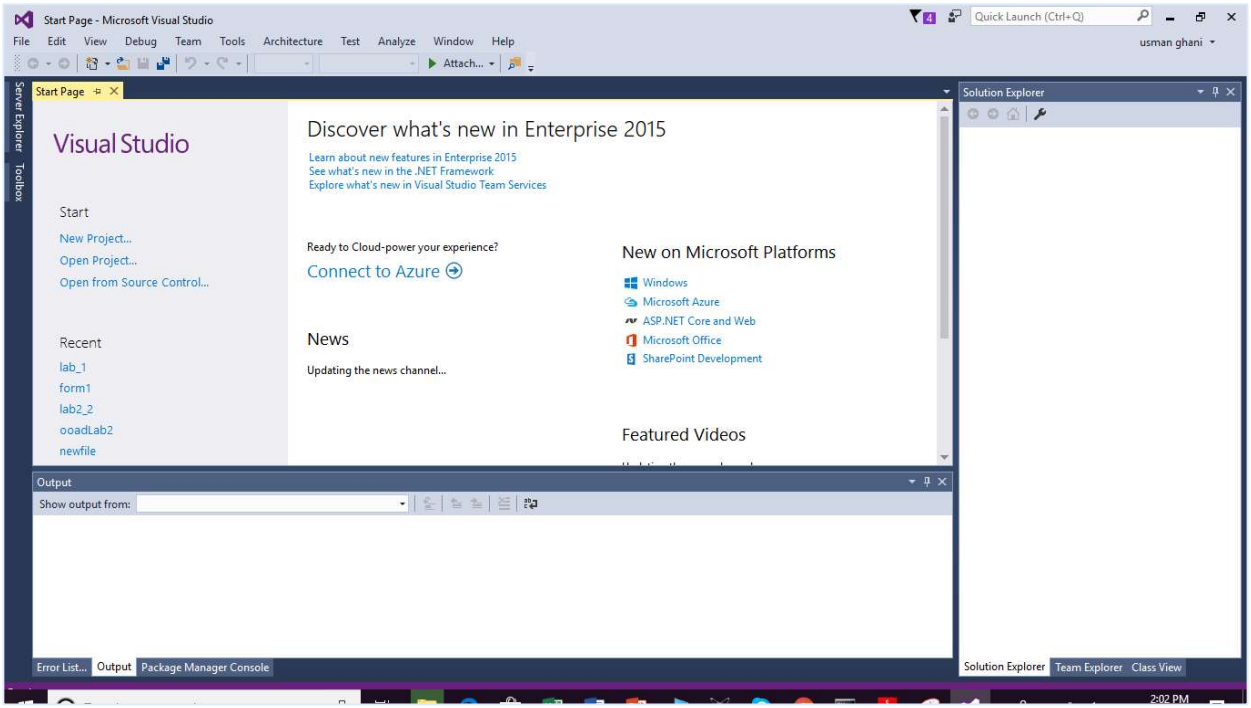


2. Button click event

A Program that show Helloworld in popup window at button click.

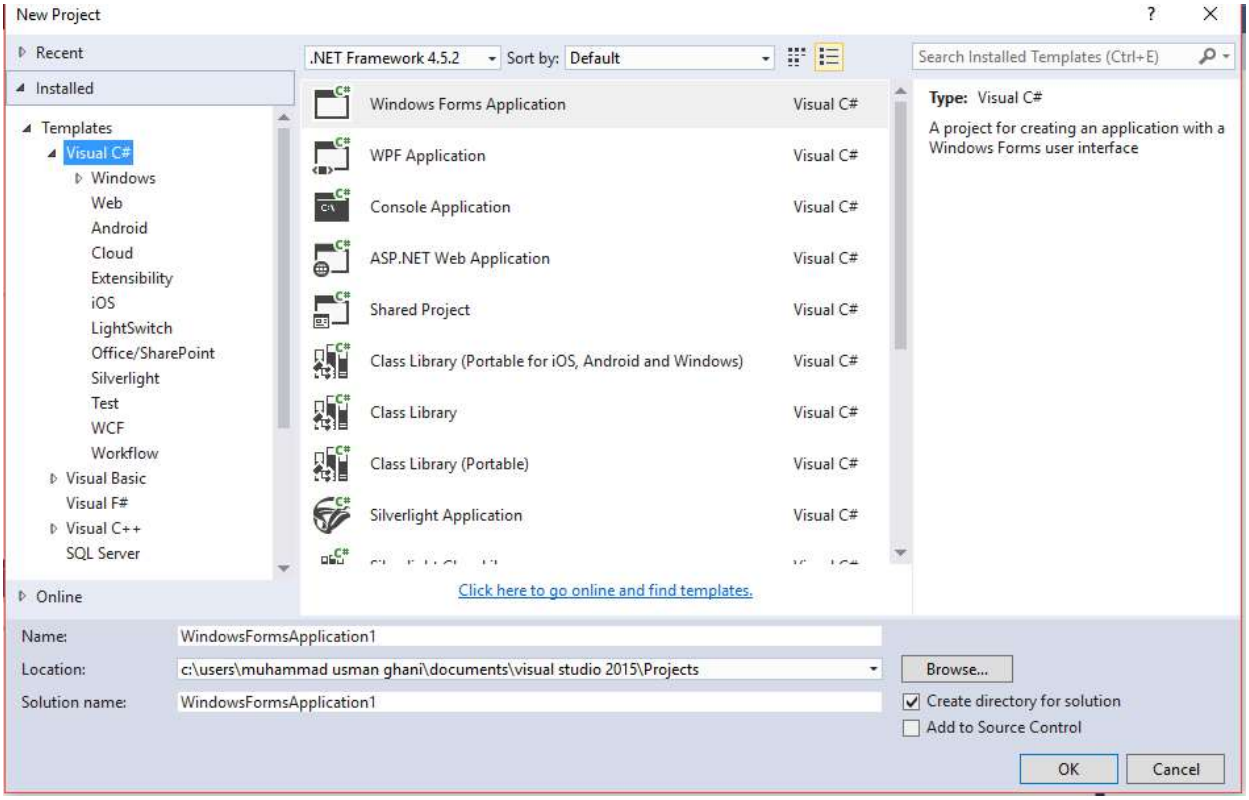
Step 1:

Open visual studio and click on new project



Step 2:

Click on new project in the most left Column.

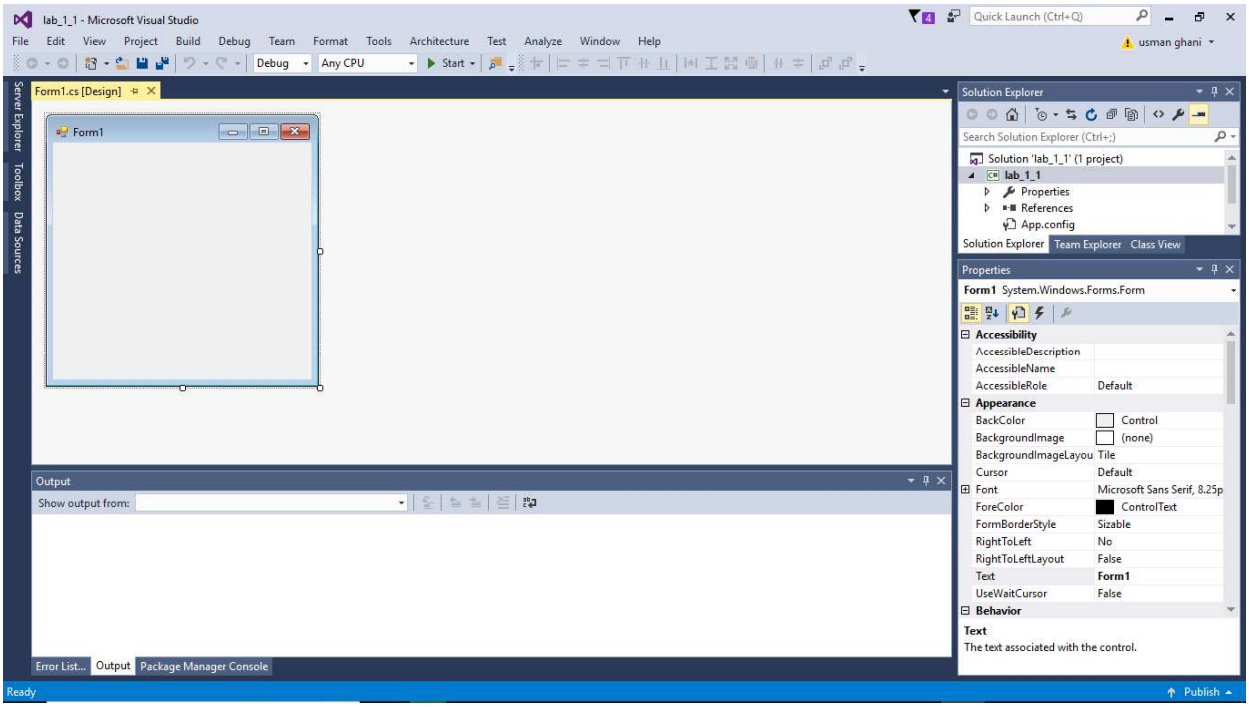


Step3:

In Installed Templates the first option is Visual C# language and then select Windows Forms application.

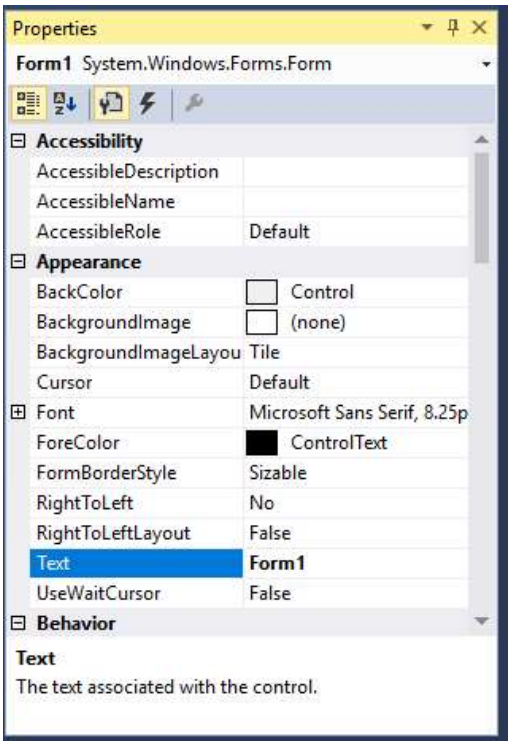
In the following see the Name label and already show the default name WindowFormsApplication1.

You can change the name as you wish or as per program creation you can mention and click and OK.



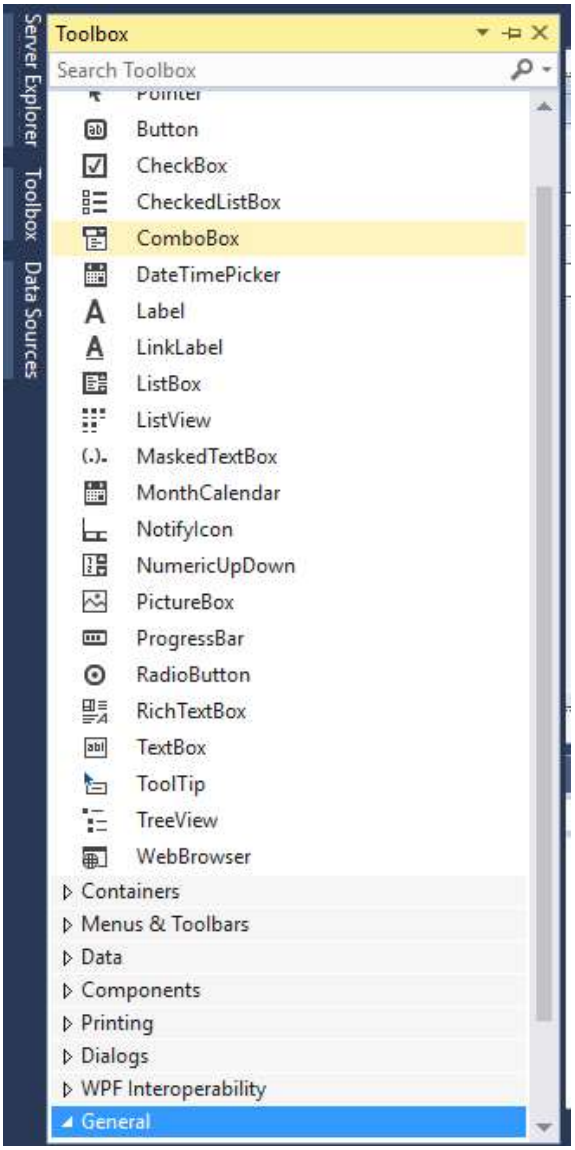
Step 4

In your Windows Forms form set the size as needed. You can set the size and if you want to set the size by properties then right-click on the Windows Forms form and in the last option of show properties click to see the Windows Forms form properties.

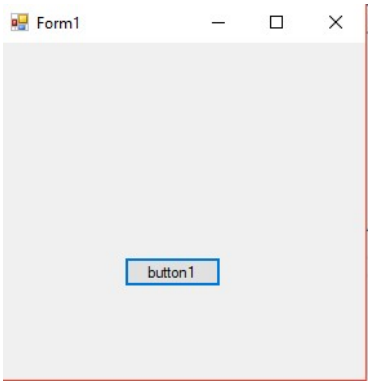


Step 5:

Click on tool box in the main window



Double click on button to make it visible on form. After clicking you will see button on form.

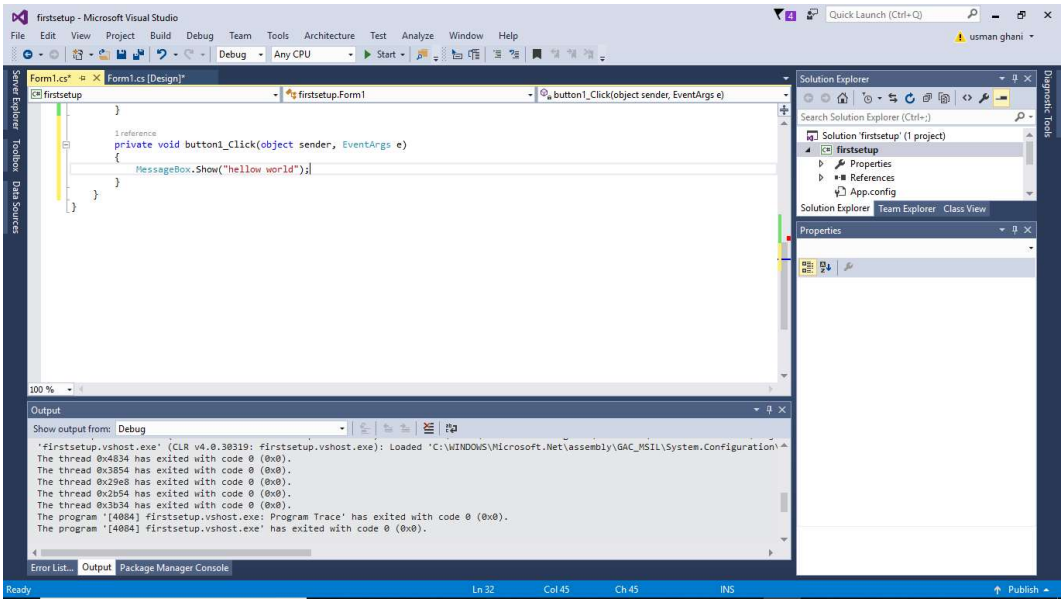


Step 6:

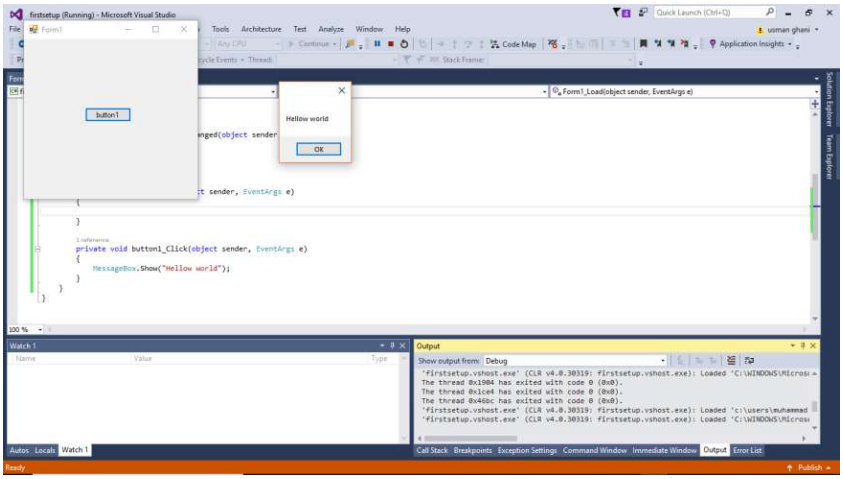
Now to double click on the button you will see the button1_cick function. (This function will execute when button1 click event will occur.

```
        1 reference
        private void button1_Click(object sender, EventArgs e)
        {

        }
    }
```



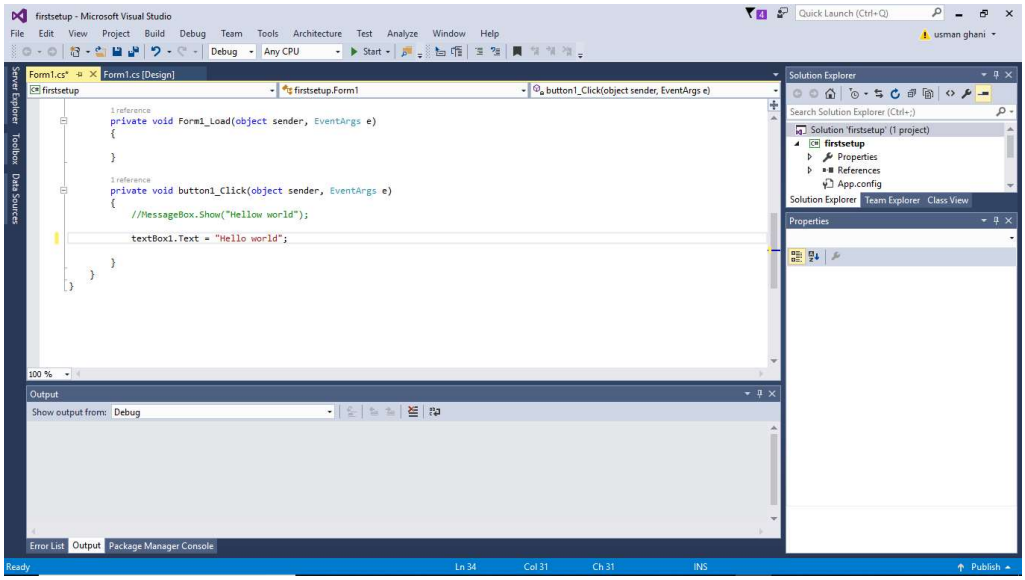
Output:



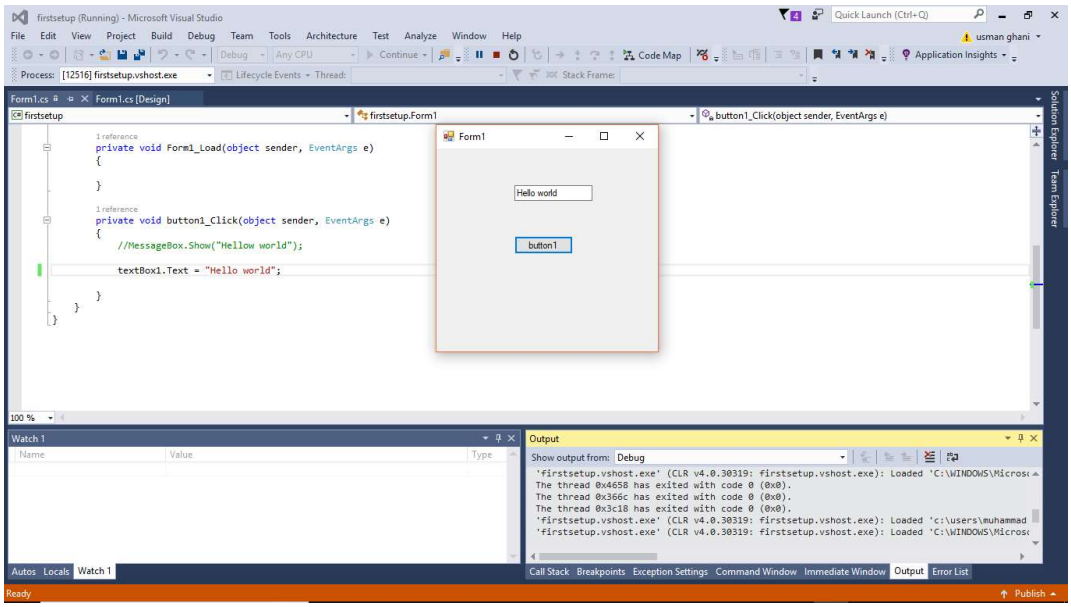
Program 3:

A program that show hello world on button click in the text box

1. **Create new project**
2. **Select button and text box from toolbox.**
3. **Double click on button to see event function.**
4. **Write code here so that when button is pressed "Hello world show in textbox"**



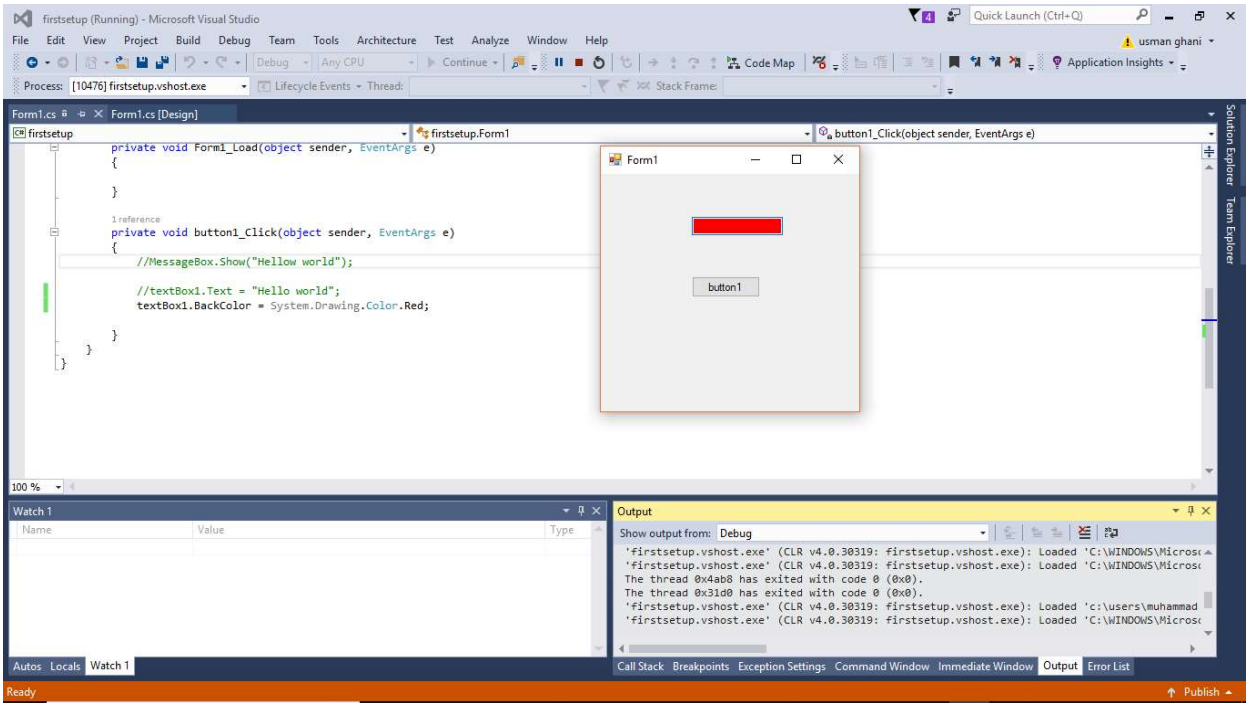
Output:



Program 4:

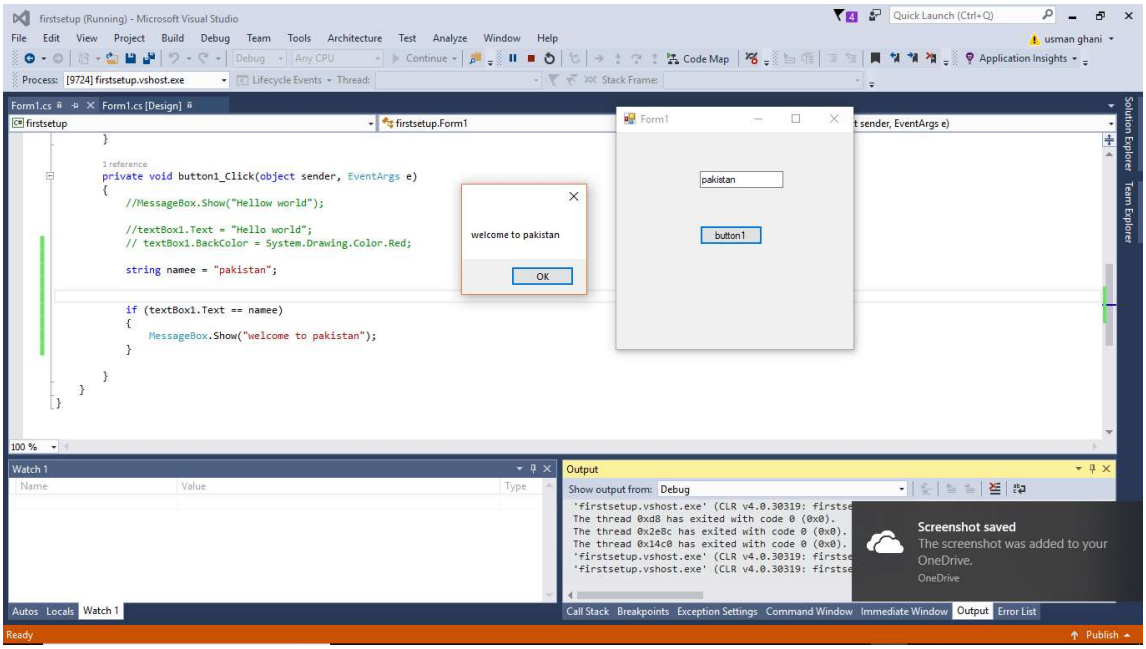
A Program that change the Color of TextBox at Button click

Textbox1.Backcolor=system.Drawing.color.Red;



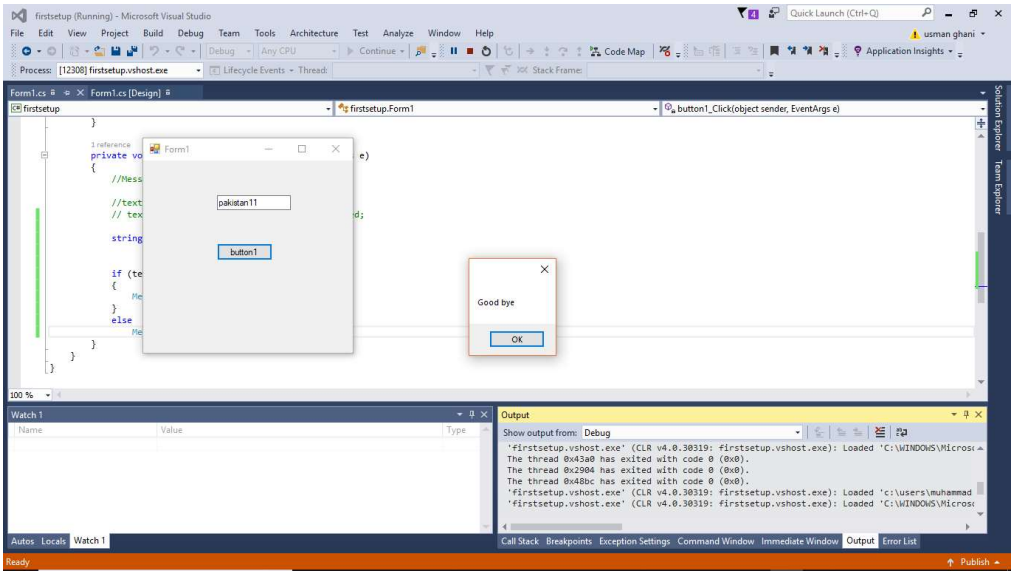
Program 5:

A Program that change the Color of TextBox at Button click



Program 6:

A Program that show the "Welcome" to user if the text of textbox matches with the string "Pakistan" and otherwise Show "Good Bye" in message box

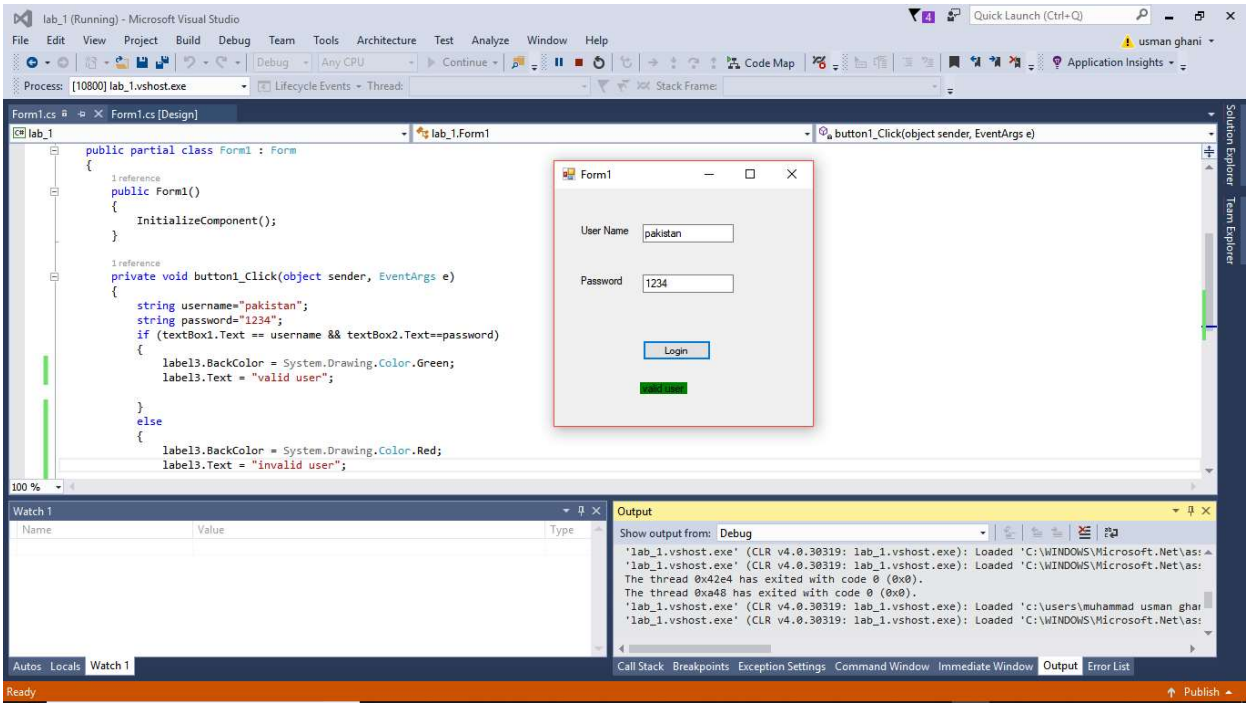


Program 7:

Create Program for Login form that show valid user in green color at the label if the user has entered correct username and password otherwise it shall say invalid username and password is red color.

- **UserName: pakistan**
- **Password: 1234**

Solution



Program 8

Redevelop the program that allow following combination for Logins

Login1

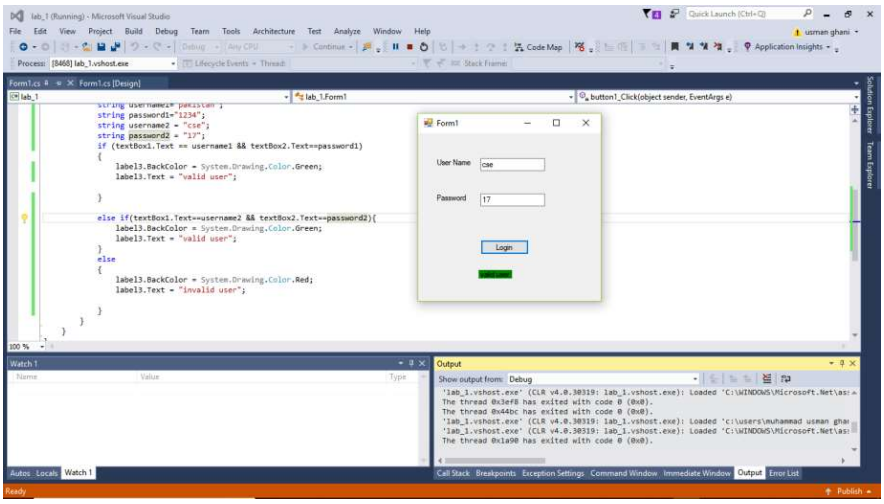
- UserName: Pakistan

- Password:123

Login 2

- UserName: cse
- Password: 17

Output1:



Output2: