Quick Introduction To C# Windows Form.NET

Data Structures

Razi University
2023

Mohammad Mehdi Rajabi
Mehdi Kaliani

Content of Presentation

- What is C# Windows Form
- Create Windows Form Project
- Label
- Button
- TextBox
- Event handling
- VS Error
- Sample Project

C# Windows Form.NET Document

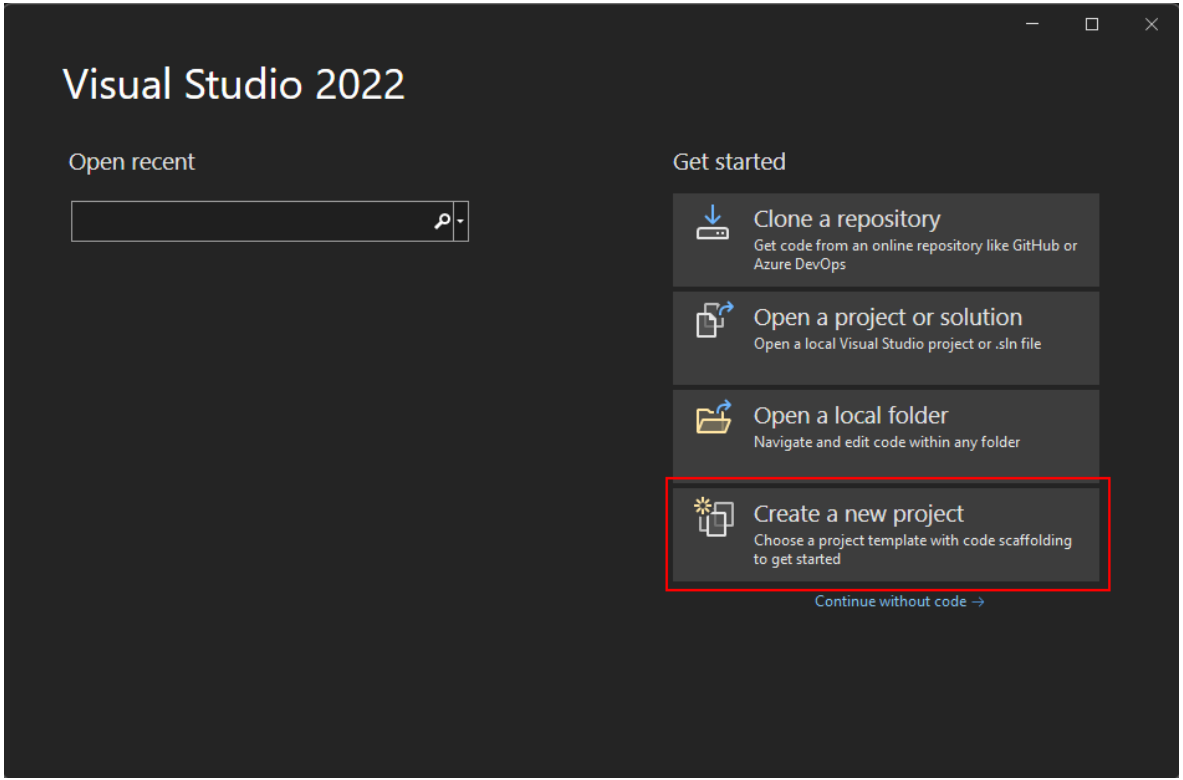
https://learn.microsoft.com/en-us/dotnet/desktop/winforms/overview/?view=netdesktop-7.0

Windows Forms is a **UI framework for building Windows desktop apps**. It provides one of the most productive ways to create desktop apps based on the **visual designer provided in Visual Studio**. Functionality such as **drag-and-drop** placement of visual controls makes it easy to build desktop apps.

With Windows Forms, you develop graphically rich apps that are easy to deploy, update, and work while offline or while connected to the internet. **Windows Forms apps can access the local hardware and file system of the computer where the app is running**.

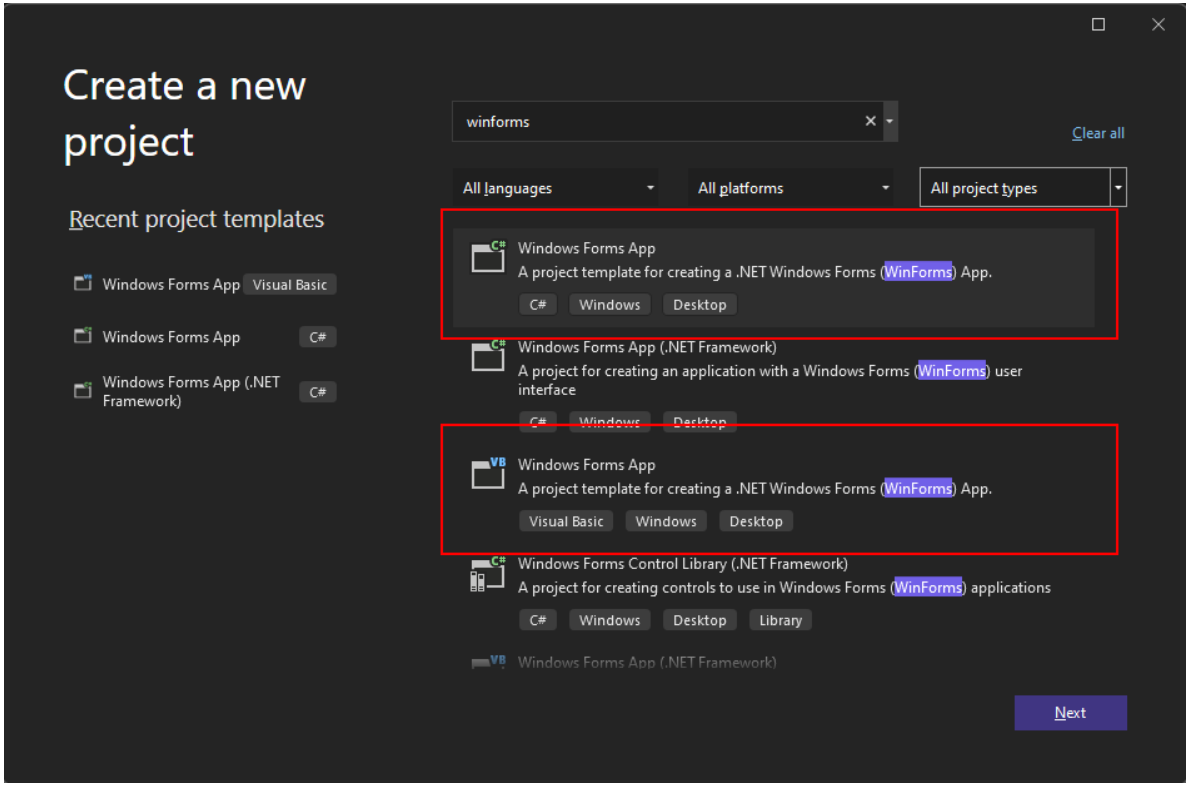
Create Windows Form Project

- ✓ Open Visual Studio.
- ✓ Select Create a new project.



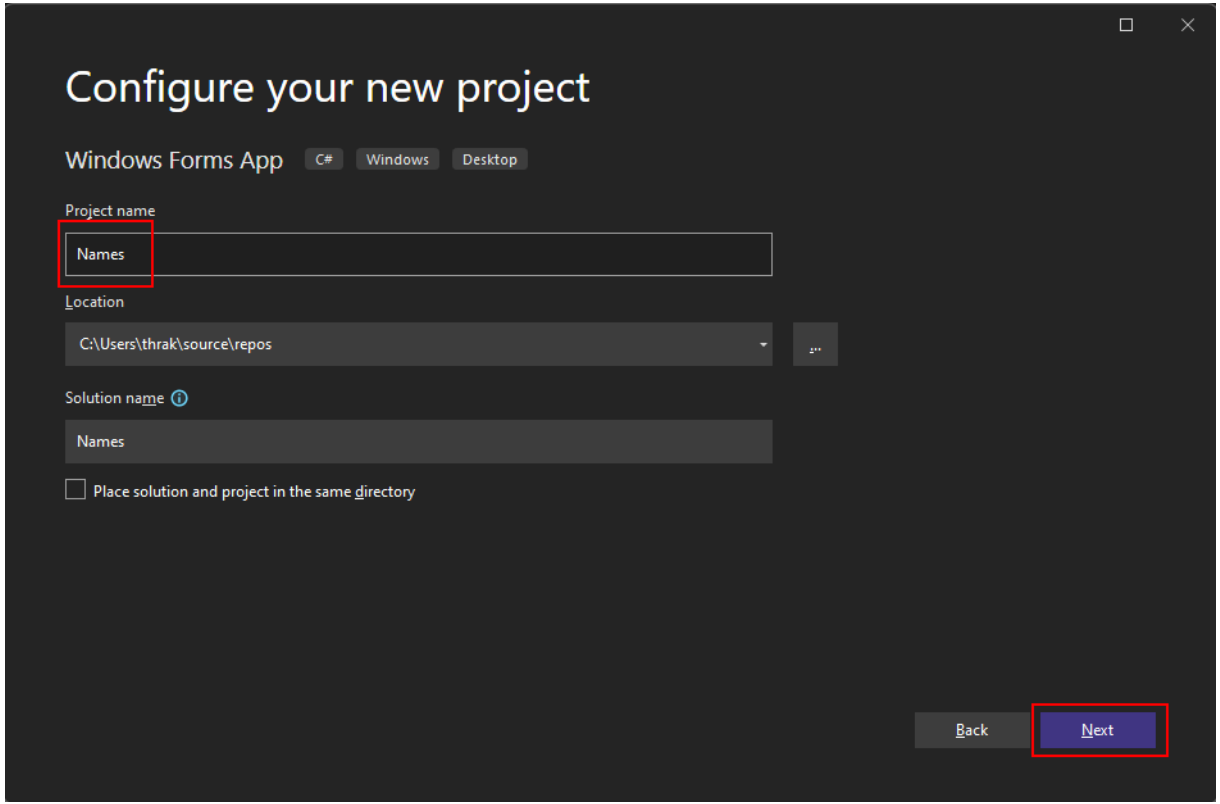
Create Windows Form Project

- ✓ In the **Search for templates box**, type **winforms**, and wait for the search results to appear.
- ✓ In the code language dropdown, choose C# or Visual Basic.
- ✓ In the **list of templates**, select **Windows Forms App** and then click Next.



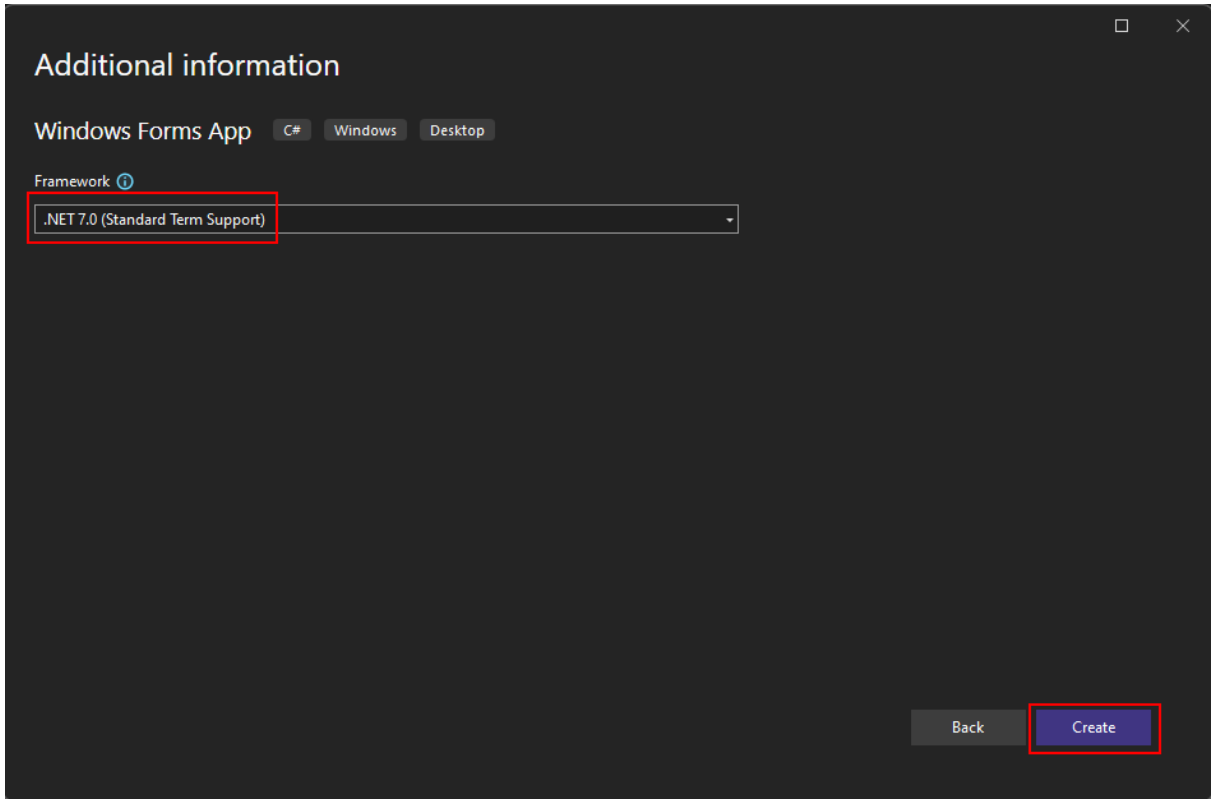
Create Windows Form Project

- ✓ In the Configure your new project window, **set the Project name to Names and click Next**.
- ✓ You can also save your project to a **different folder by adjusting the Location path**.



Create Windows Form Project

- ✓ Finally, in the Additional information window, **select .NET version.**



- **Solution Explorer**

All of your **project files, code, forms, resources,** will appear in this pane.

- **Properties**

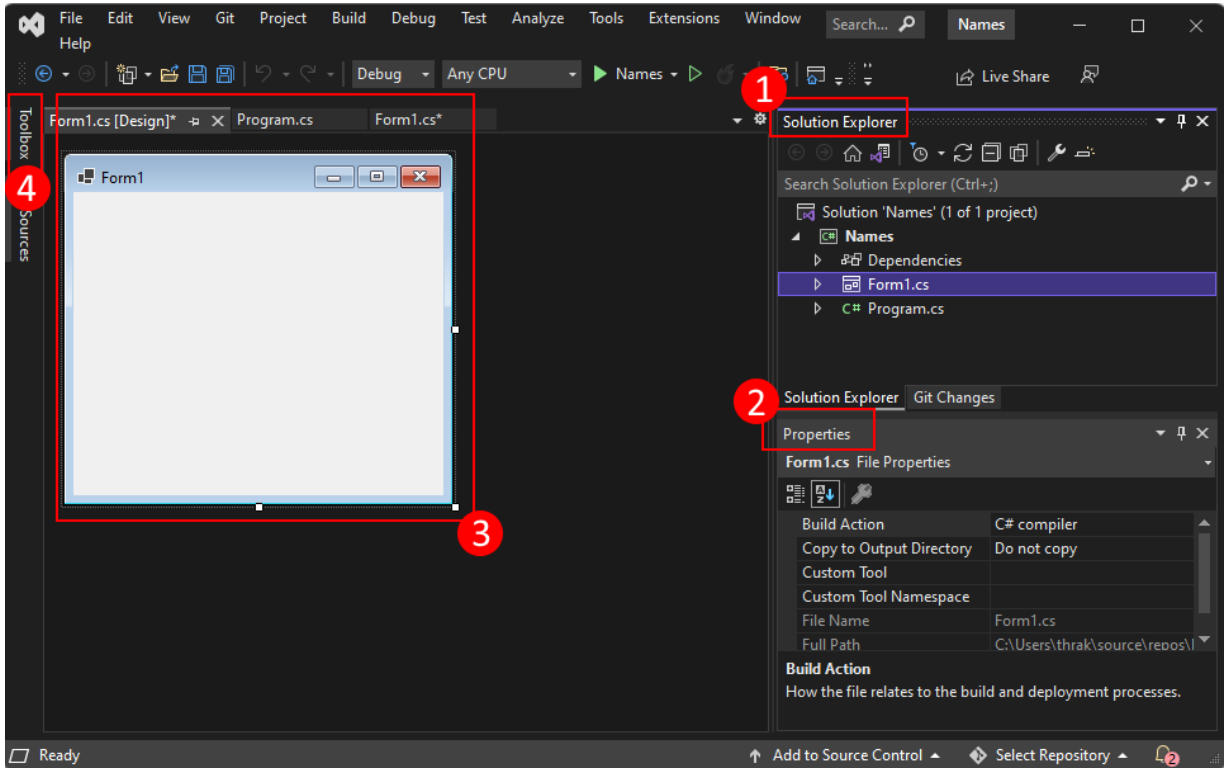
This pane **shows property settings you can configure based on the item selected**. For example, if you select an item from Solution Explorer, you'll see property settings related to the file. If you select an object in the Designer, you'll see settings for the control or form.

- **Form Designer**

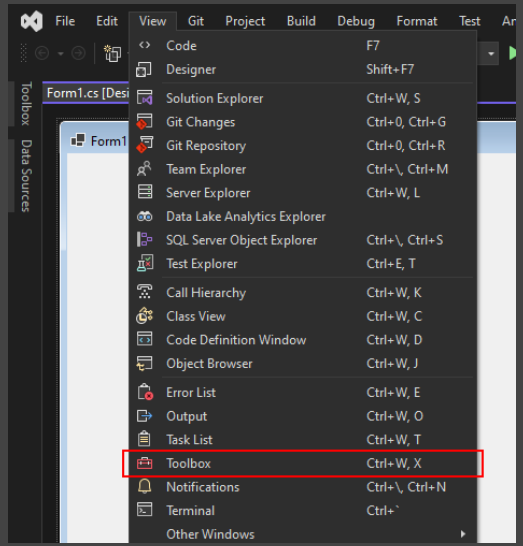
This is the designer for the form. It's interactive and you can **drag-and-drop objects from the Toolbox.** By selecting and moving items in the designer, you can visually compose the user interface (UI) for your app.

- **Toolbox**

The toolbox **contains all of the controls you can add to a form**. To add a control to the current form, double-click a control or drag-and-drop the control.

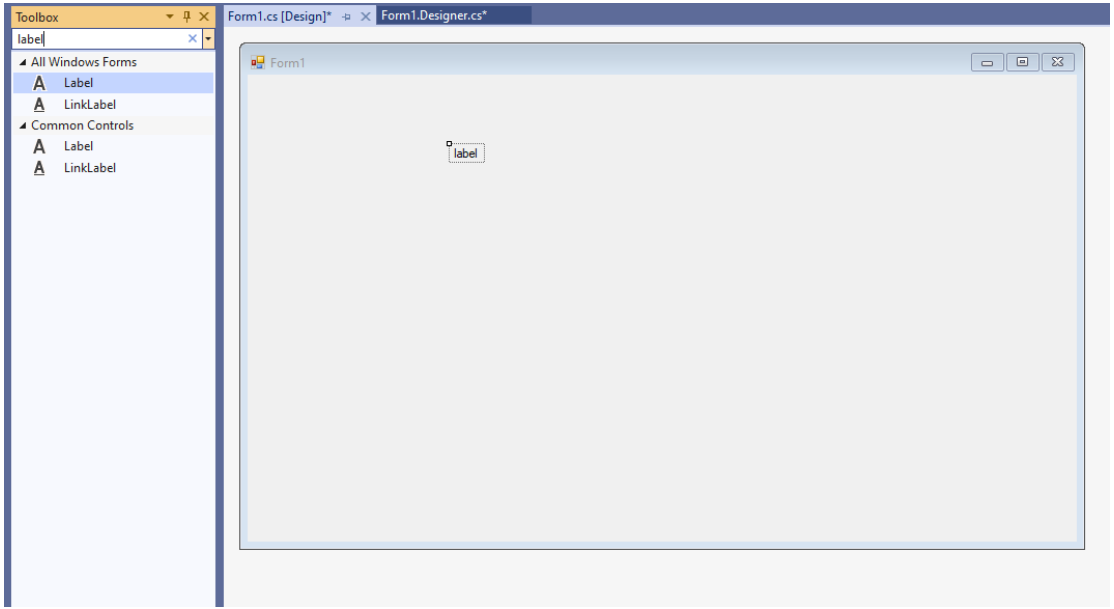


If the toolbox isn't visible, you can display it through the View > Toolbox menu item.



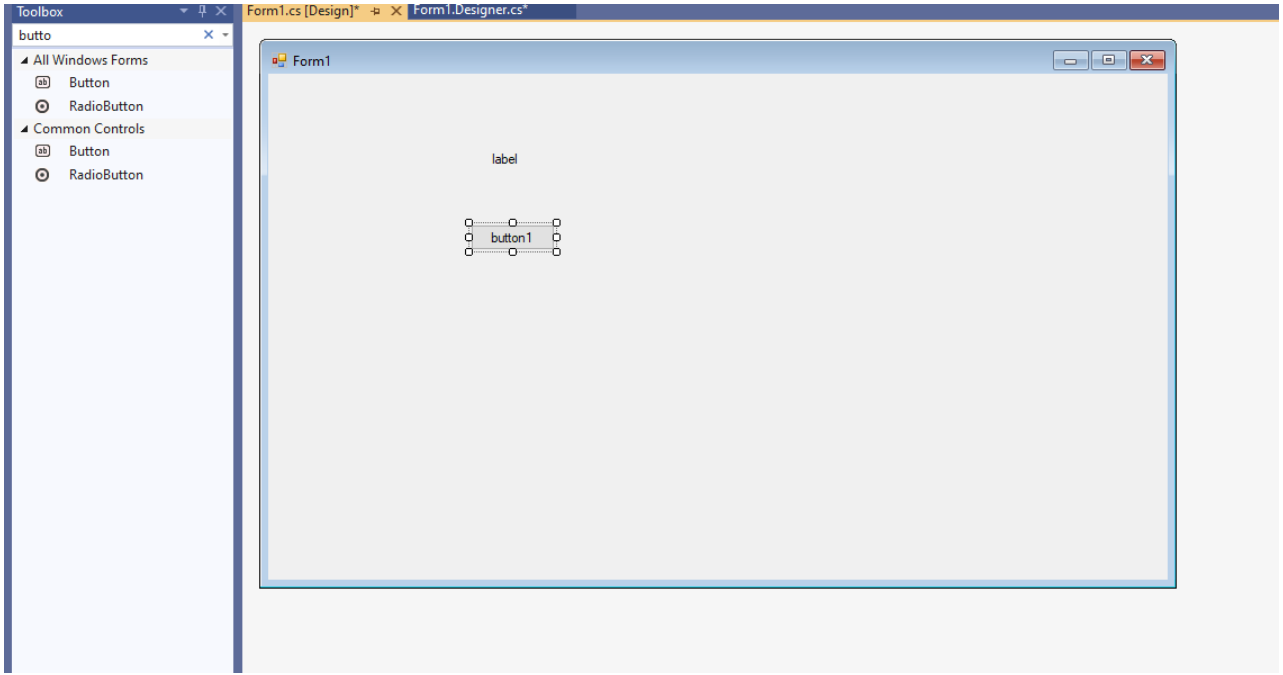
Label

Windows Forms Label controls are **used to display text that cannot be edited by the user**. They're **used to identify objects on a form** and to provide a description of what a certain control represents or does. For example, you can use labels to add descriptive captions to text boxes, list boxes, combo boxes, and so on. You **can also write code that changes the text displayed by a label in response to events at run time**.



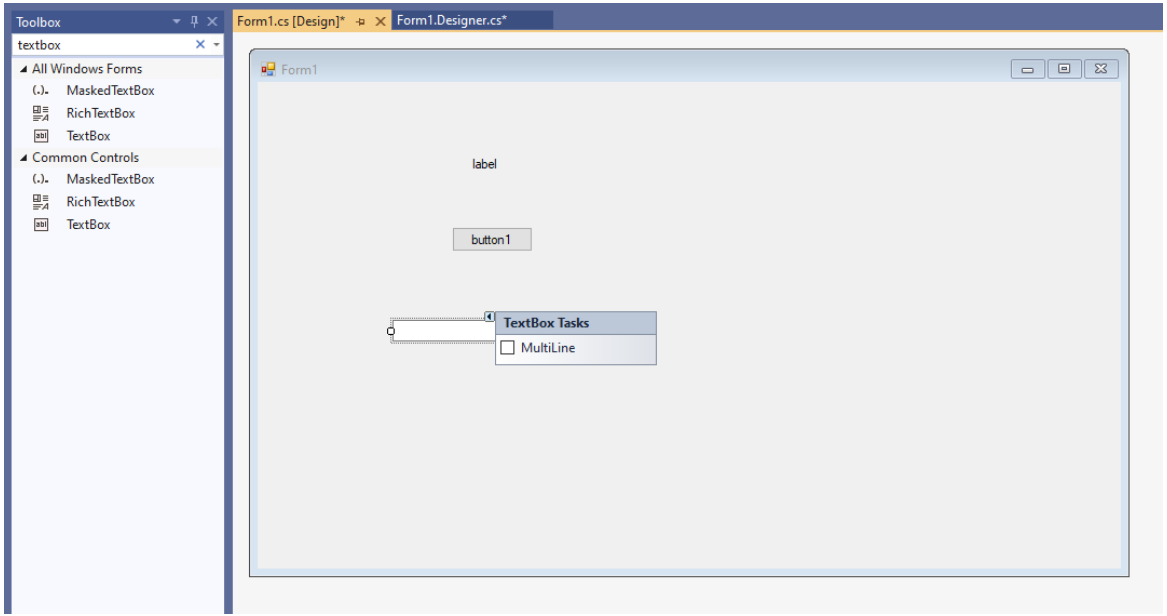
Button

The Windows Forms Button control **allows the user to click it to perform an action**. When the button is clicked, it looks as if it is being pushed in and released. Whenever the **user clicks a button, the Click event handler is invoked**. You place code in the Click event handler to perform any action you choose.



TextBox

Windows Forms text boxes are **used to get input from the user or to display text**. The TextBox control **is generally used for editable text**, although **it can also be made read-only**. Text boxes can display **multiple lines**, wrap text to the size of the control, and add basic formatting. The TextBox control provides a single format style for text displayed or entered into the control. To display multiple types of formatted text, use the RichTextBox control.



Event handling

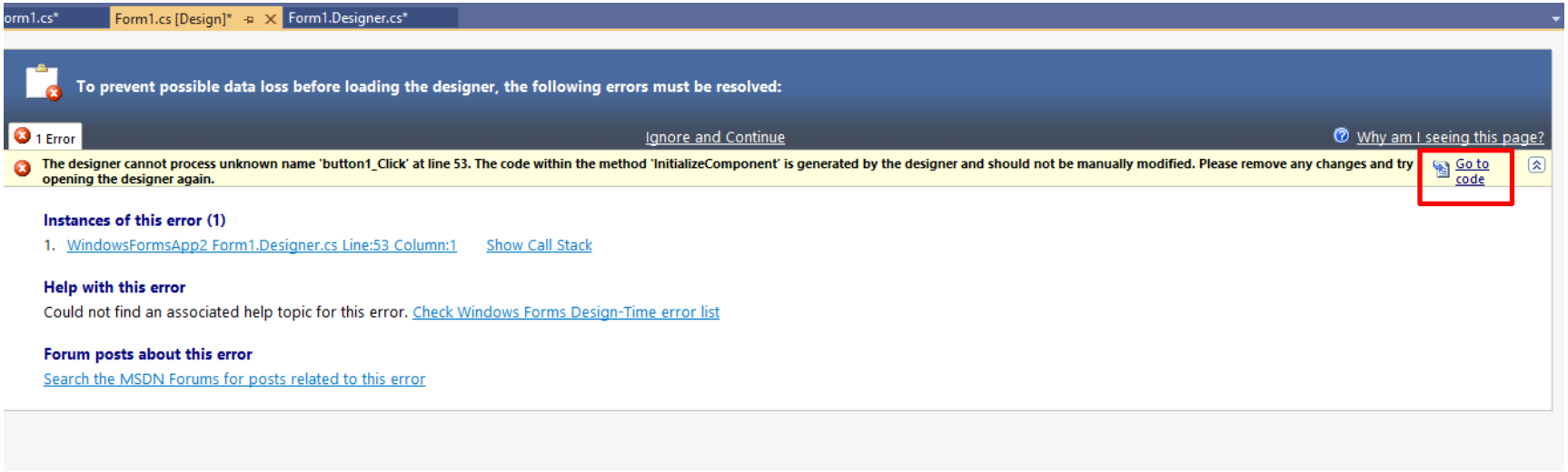
Controls **provide events that are raised when the user interacts with the control or when the state of the control changes**.

Common events

Controls provide a set of common events through the base class: Control. Not every control responds to every event. For example, the Label control doesn't respond to keyboard input, so the Control.PreviewKeyDown event isn't raised. Most shared events fall under these categories:

- **Mouse events**
- **Keyboard events**
- **Property changed events**
- **Other events**

VS Error


orm1.cs*
Form1.cs [Design]*
Form1.Designer.cs*
To prevent possible data loss before loading the designer, the following errors must be resolved:
1 Error
Ignore and Continue
Why am I seeing this page?
The designer cannot process unknown name 'button1_Click' at line 53. The code within the method 'InitializeComponent' is generated by the designer and should not be manually modified. Please remove any changes and try opening the designer again.
Go to code
Instances of this error (1)
1. WindowsFormsApp2 Form1.Designer.cs Line:53 Column:1
Show Call Stack
Help with this error
Could not find an associated help topic for this error.
Check Windows Forms Design-Time error list
Forum posts about this error
Search the MSDN Forums for posts related to this error

Sample Project

GitHub repository

https://github.com/Mohammad-Mehdi-Rajabi/windows-form-sample-for-razi-uni-data-structures-course

End