CSE 1322L - Lab 12

Introduction

In this lab, you will create a simple summation application with a GUI (Graphical User Interface). The application must have boxes where the user can enter numbers, buttons for performing the summation and clearing the results, and a label to display the result of the summation.

Note that this assignment must be written in C#. Also note that **it is recommended that you do this assignment in Visual Studio** (not Visual Studio Code), as Visual Studio has tools to facilitate creating GUI elements and adding functionality to them.

Instructions on how to install and use Visual Studio can be found in the appendix, after the lab's sample outputs.

Requirements

The features described below must be in your program (any fields not delineated should be left with their default values):

- Form properties
  - **Appearance**: "Text" must be "FYE Summator"
  - **Layout**: "Size" must be "360, 110"
- Left-most TextBox properties
  - **Appearance**: "Text" must be an empty string
  - **Design**: "(Name)" must be "operand1txtbox"
  - **Layout**: "Location" must be "12, 12" and "Size" must be "100, 23"
- Right-most TextBox properties
  - **Appearance**: "Text" must be an empty string
  - **Design**: "(Name)" must be "operand2txtbox"
  - **Layout**: "Location" must be "118, 12" and "Size" must be "100, 23"
- Label properties
  - **Appearance**: "Text" must be an empty string
  - **Design**: "(Name)" must be "resultlbl"
  - **Layout**: "AutoSize" must be "True", "Location" must be "224, 15"
- Left-most Button properties
  - **Appearance**: "Text" must be "Sum"
  - **Design**: "(Name)" must be "sumbtn"
  - **Layout**: "Location" must be "12, 41" and "Size" must be "74, 23"
- Right-most Button properties
  - **Appearance**: "Text" must be "Clear"
  - **Design**: "(Name)" must be "clearbtn"

    - **Layout**: “Location” must be “92, 41” and “Size” must be “74, 23”

When clicked, the buttons perform the following actions:

- **clearbtn**: operand1txtbox, operand2txtbox, and resultlbl have their Text set to an empty string.
- **sumbtn**:
    - Performs an arithmetic addition using the contents of operand1txtbox and operand2txtbox, displaying the results on resultlbl. You can assume the user will only enter whole numbers.
    - If either TextBox does not contain a number, resultlbl should instead display “That is not a number!”. This must be performed using a TRY-CATCH and the specific exception thrown must be caught (i.e.: do not catch an Exception type exception).

Deliverables

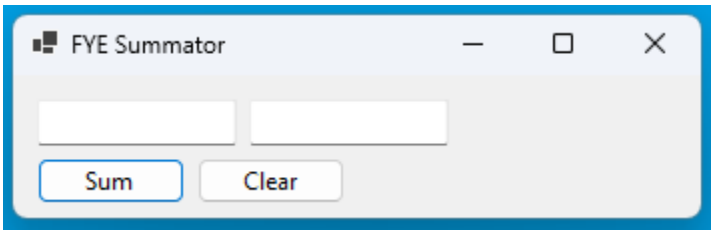
- Form1.cs
- Form1.Designer.cs
- Program.cs

Considerations

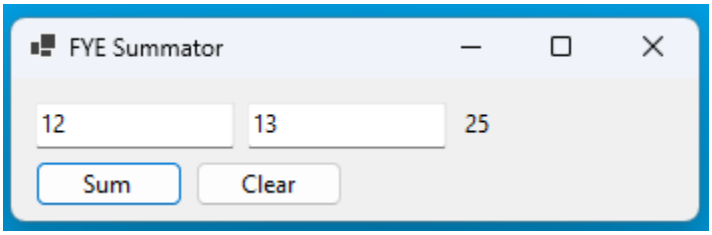
- As a general rule, do not delete any code that you haven’t personally written. Doing so might break your form. If you break your form, it will likely be easier to just restart the project from scratch rather than trying to fix it. Avoid renaming the files as well.
- The different files you have to submit have different purposes
    - Program.cs is your driver, which you should not edit.
    - Form1.cs contains the behavior of your form. Whenever an action is performed on a form element (e.g.: clicking a button), your form will check this file to figure out what actions should be performed
    - Form1.Designer.cs contains the appearance of your form. You should not edit this file manually. Doing so may break it. However, notice that as you edit the properties of a GUI element, this file will automatically edit itself to accommodate your changes.
- If you are using Visual Studio to do this lab, you’ll notice that you will be writing very little code. Most of your effort will be spent adding and editing the GUI elements, which should be done using Visual Studio’s GUI.
- The contents of Labels and TextBoxes are always stored as strings. As such, you will need to perform some conversions when retrieving and updating these elements.
- You will likely use this method to convert the content of a TextBox to an integer. Check its documentation to figure out which exception you should catch if the user types a non-number.

Sample Output

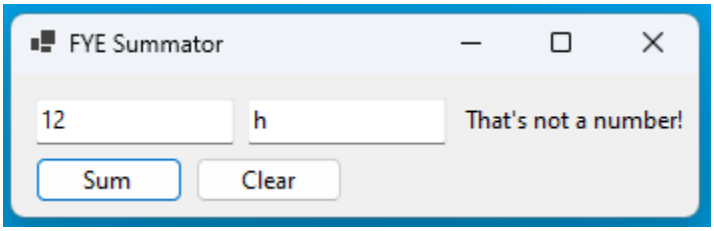
Initial state or after clicking “Clear”



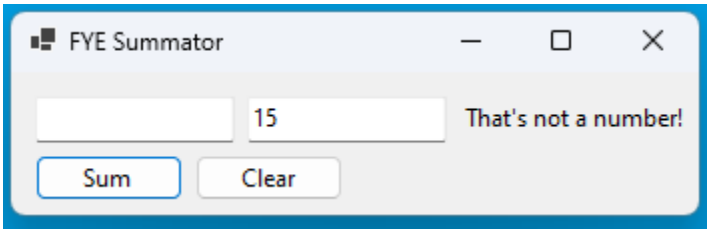
User enters numbers in the boxes and clicks “Sum”



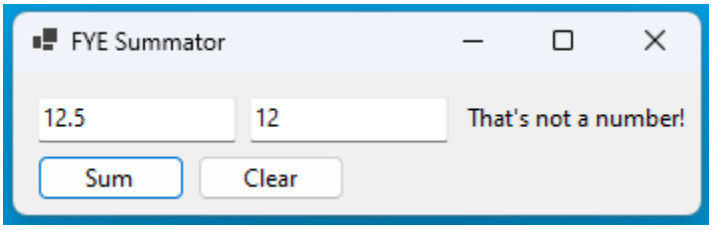
Either field contains a non-number and “Sum” is clicked



Either field is empty and “Sum” is clicked



Either field has a non-integer and “Sum” is clicked

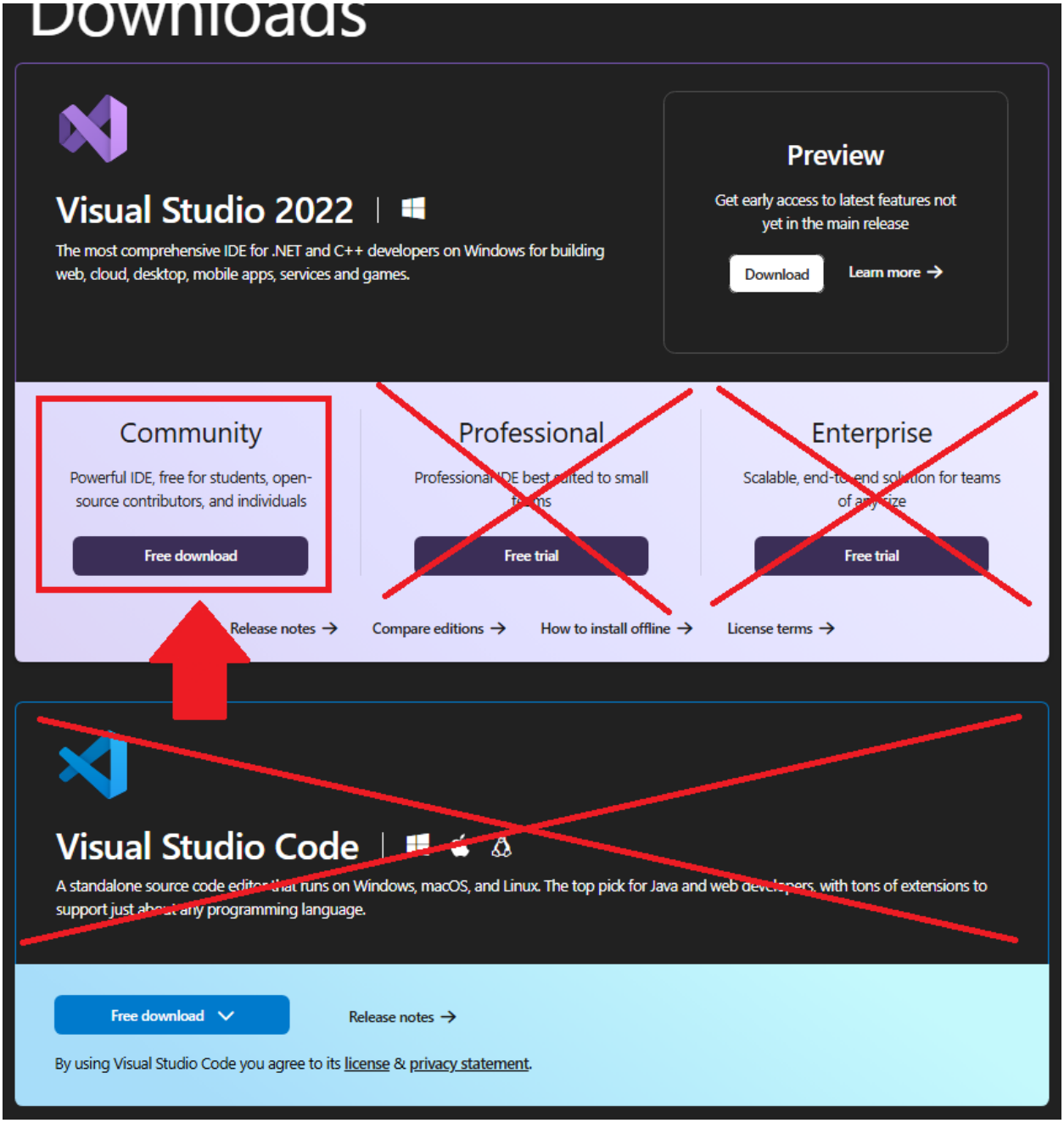


Appendix

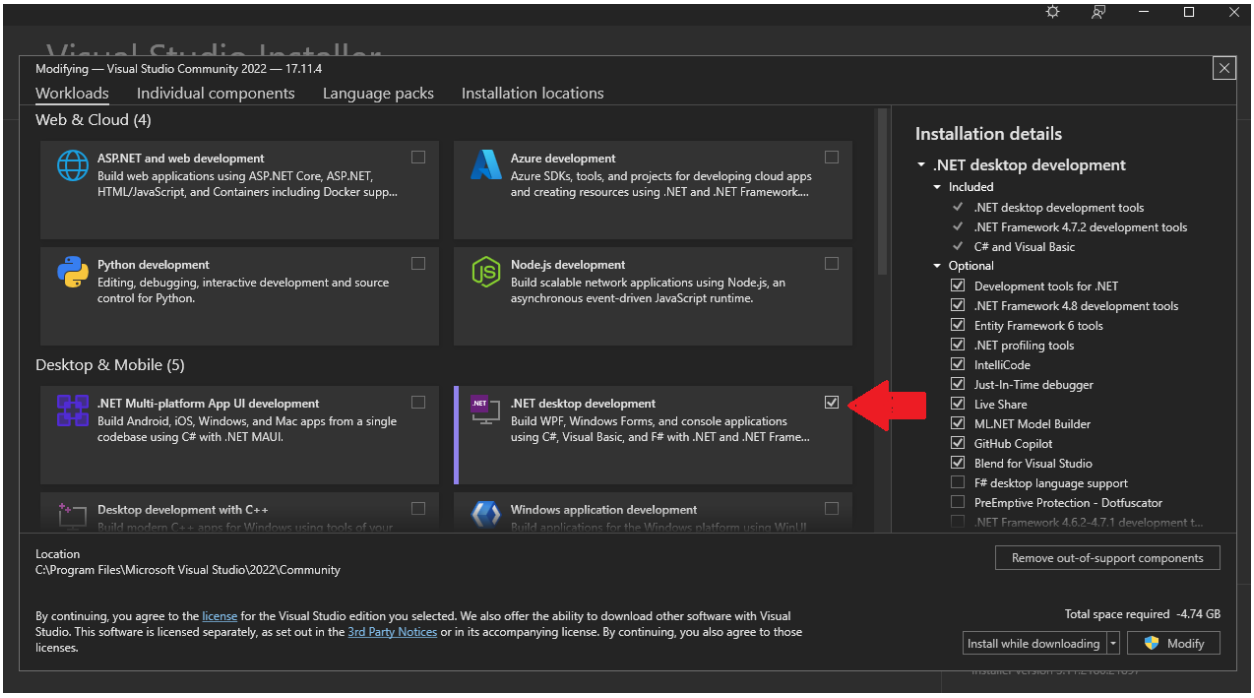
The following pages provide instructions on how to install and use Visual Studio in so far as is required by this lab.

Installing Visual Studio

The current latest version of Visual Studio is 2022 (as of Spring 2025). You can find it here. Note that **you must install Visual Studio's Community Edition**, not any of the other editions and definitely not Visual Studio Code.



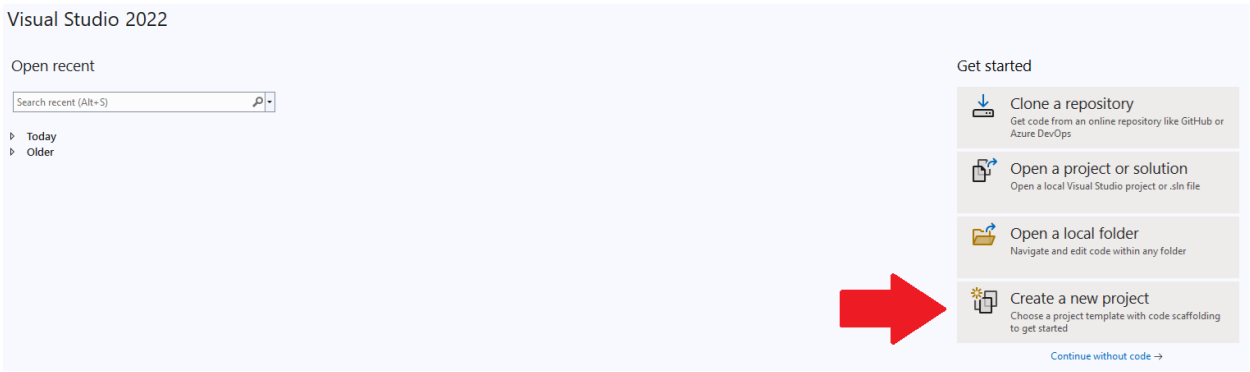
Once you've downloaded Visual Studio 2022 Community Edition's installer, run it. Proceed with the installation until you are asked what components you would like to install. Under "Workloads", make sure only ".NET desktop development" is selected and then click the button at the bottom right (The one in the image below says "Modify" because the computer which took this image already has Visual Studio installed. Yours will probably say "Install" or "Next").



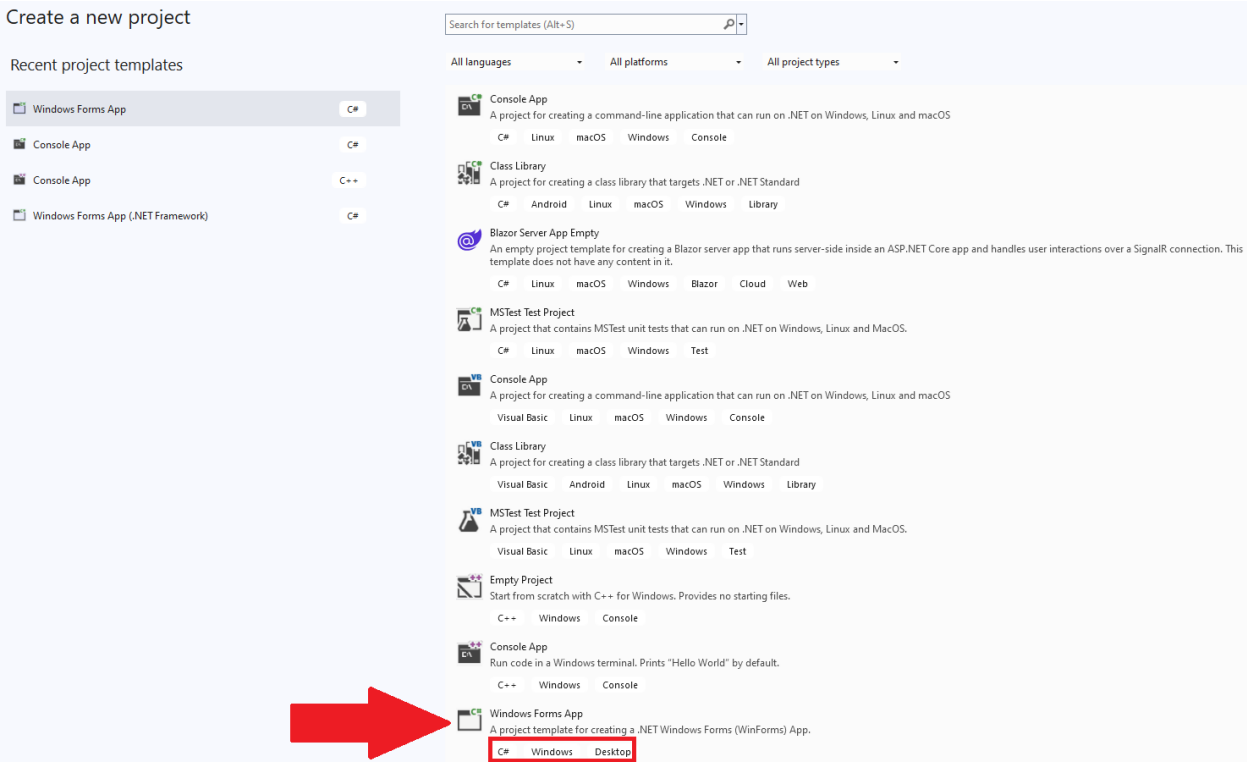
Wait for the installation to be finished.

Creating a New Project

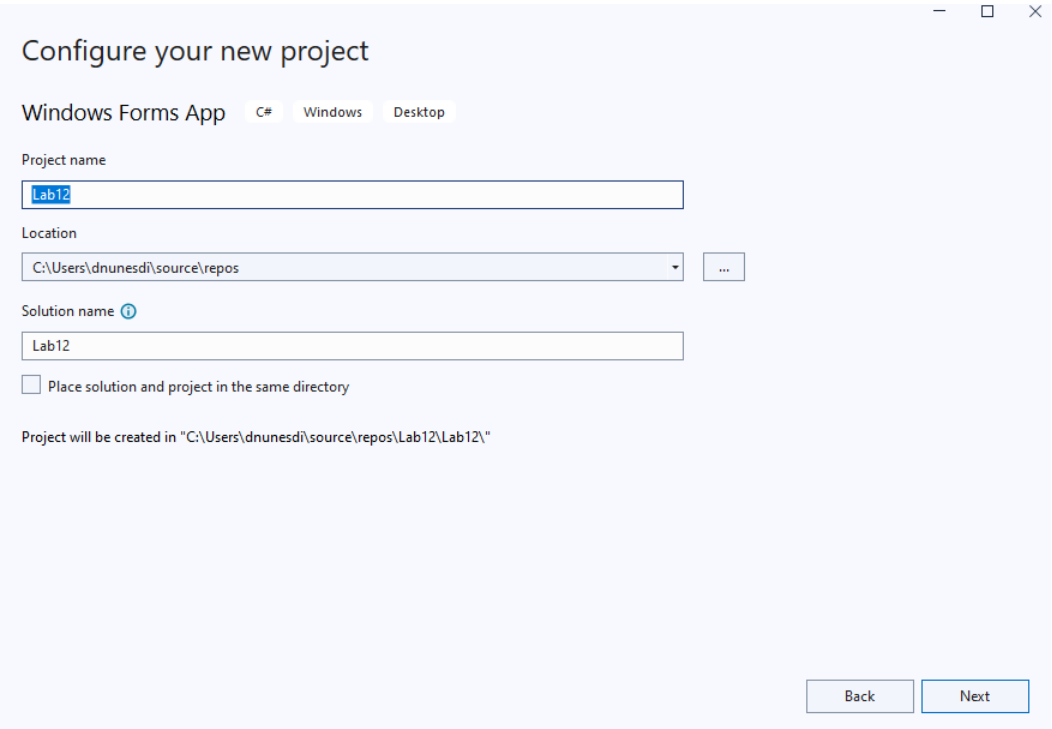
After finishing the installation or when opening Visual Studio for the first time, the following window will appear. Click "Create a new project".



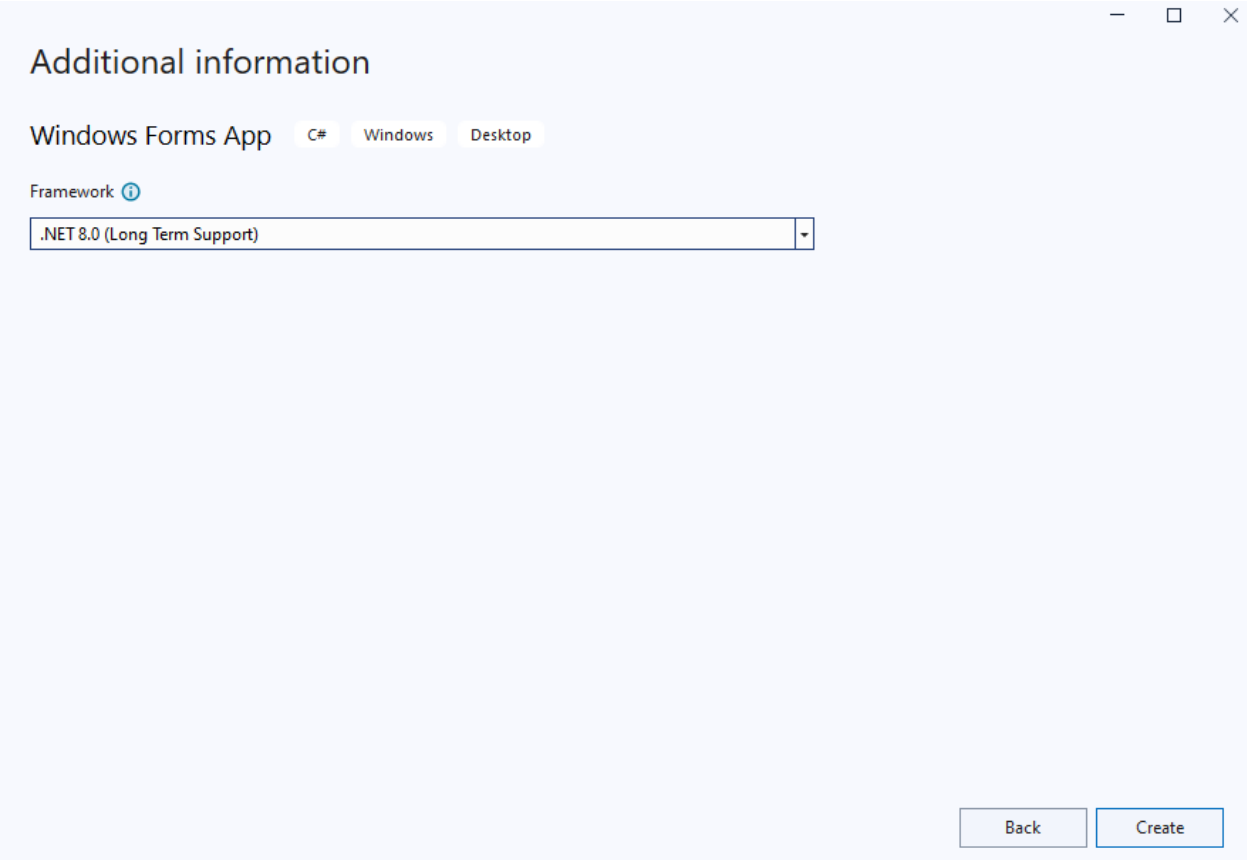
In the following window, select “Windows Forms App” from the templates on the right. Be sure that you are selecting one that uses “C#”, “Windows” and “Desktop” as its keywords. After selecting it, click “Next”.



In the next screen, name your project “Lab12” and then click “Next”.



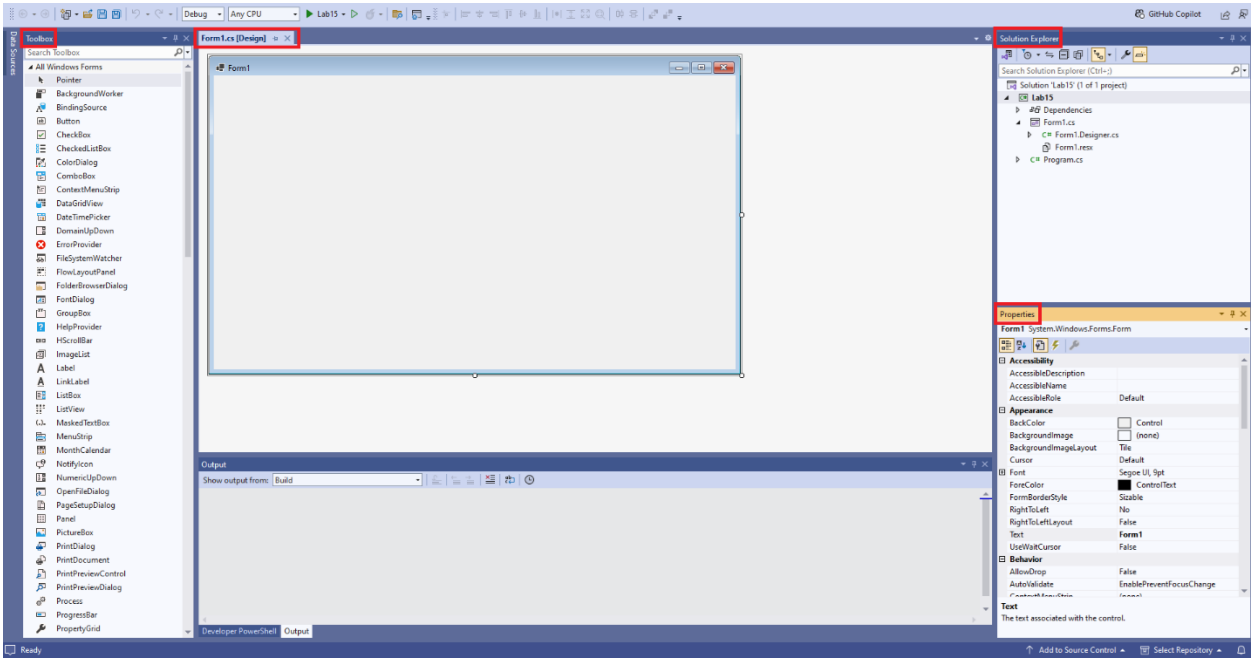
In the next screen, select “.NET 8.0 (Long Term Support)” as your framework. Click “Create”.



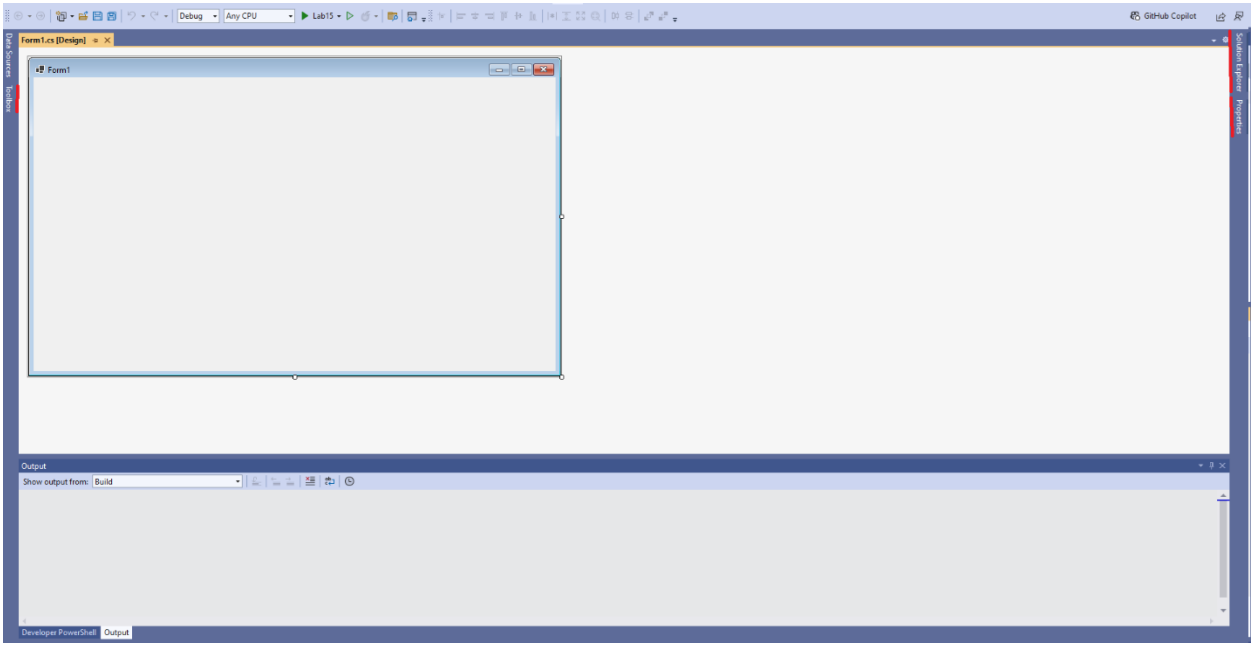
Ensuring Tools and Properties are visible

When working on a Windows Forms project, your IDE should ideally be displaying the following components:

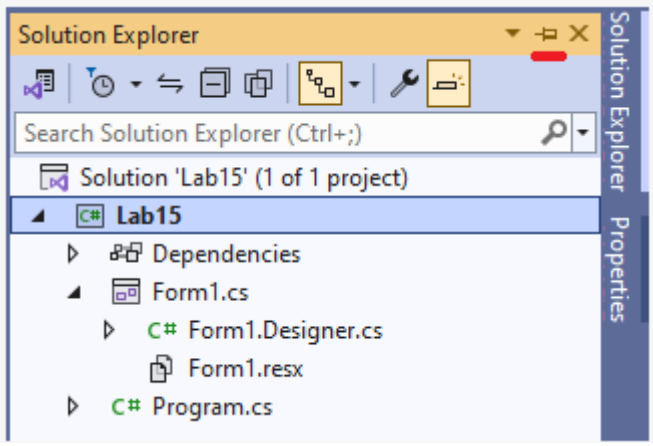
- **Toolbox**, which is where you can select new GUI elements to add. This is usually at the left
- The **Solution Explorer**, which shows your project files. This is usually to the top right.
- The **Properties** window, which shows the properties of the currently-selected element. This is usually at the bottom right.
- The file you currently have opened, usually at the center. When starting a new project, a new form will be created automatically (called “Form1”). Its Designer window will usually be opened by default.



It is possible to have these windows unpinned, to give you more space to work. Notice that they are still present at the margins, but will only remain visible while selected.

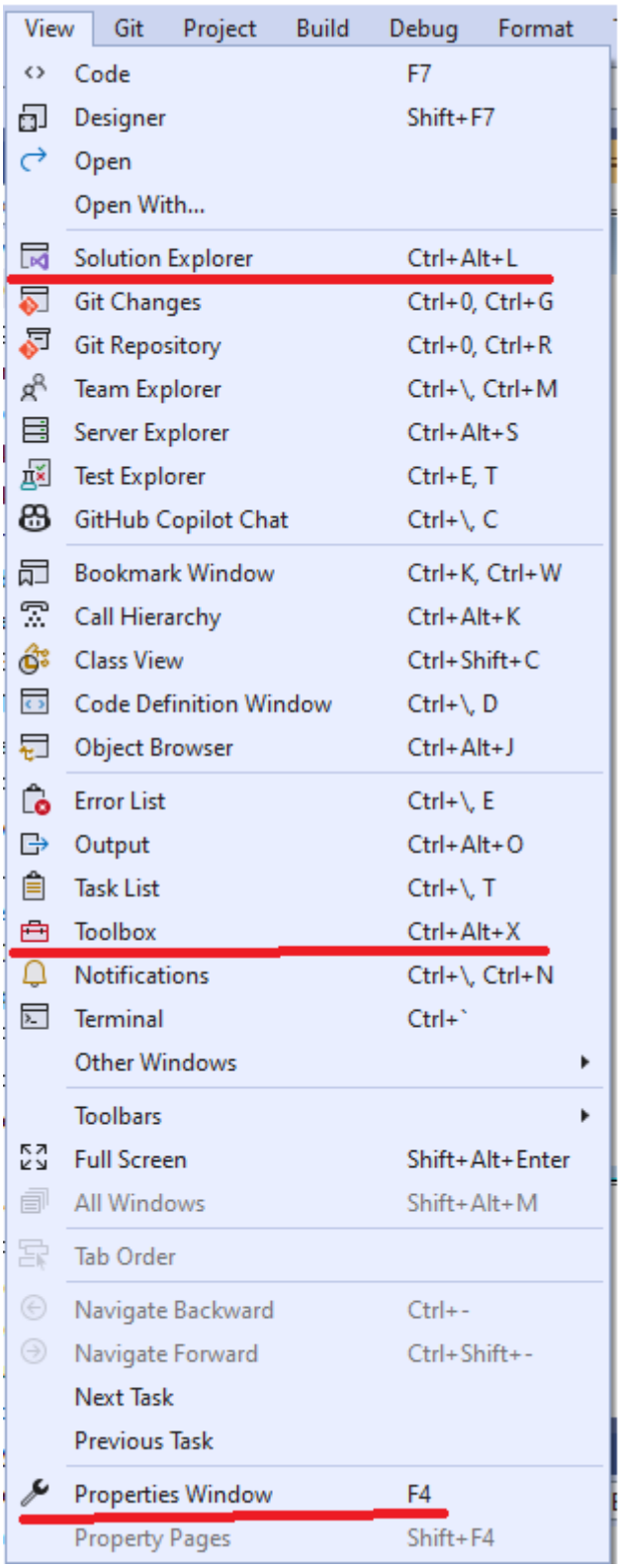


To pin these windows again (i.e.: keep them open even when you don't have them selected), click the button to reveal the window and then click the little pin at the top right.



If you prefer these windows in different spots on your screen, you can drag and drop them to your desired location.

If you close any of these windows by accident, you can always re-open them through the “View” menu at the top left.

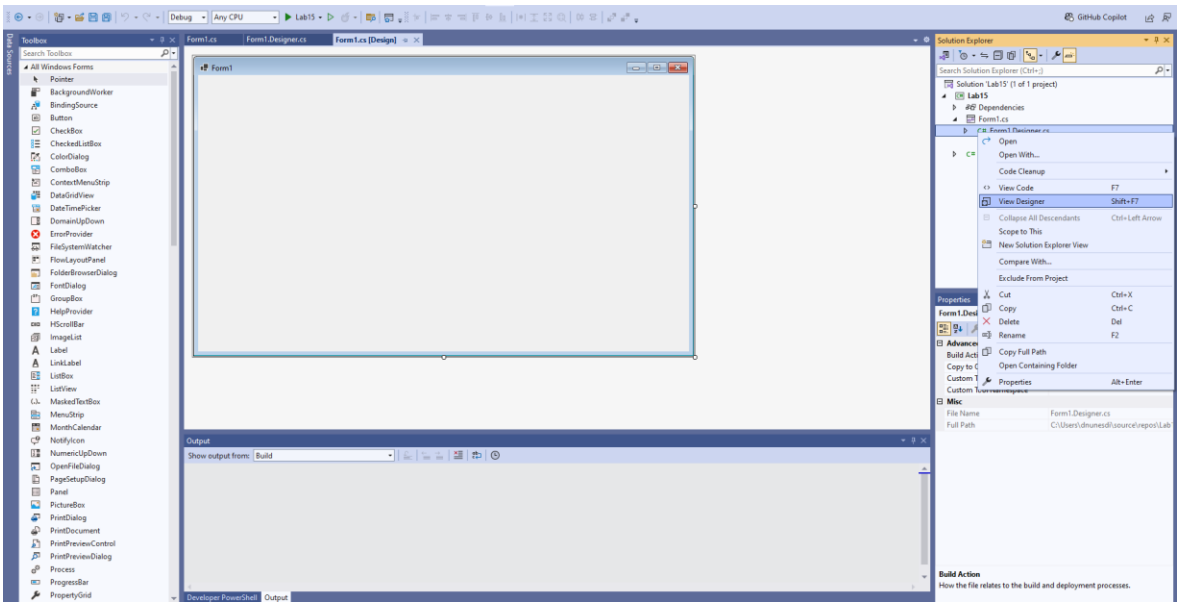


Adding GUI elements

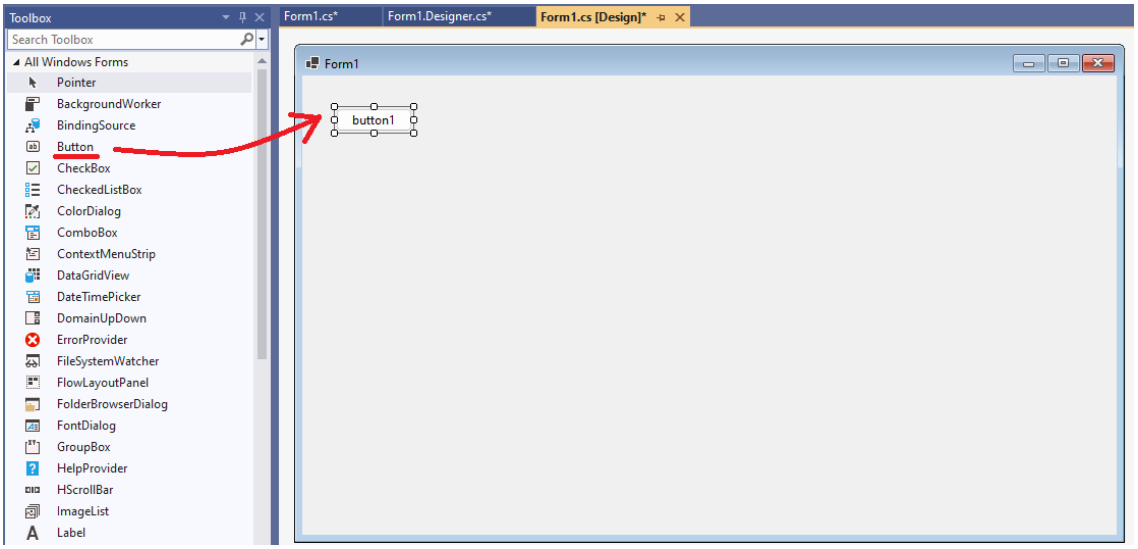
To add GUI elements to your form, you first make sure that the following are open:

- The Toolbox
    - See the previous section on how to display this
- The Designer for your form
    - On the Solution Explorer, right click your designer file (it will probably be called “Form1.Designer.cs”) and then select “View Designer”

Your workspace should look like this:

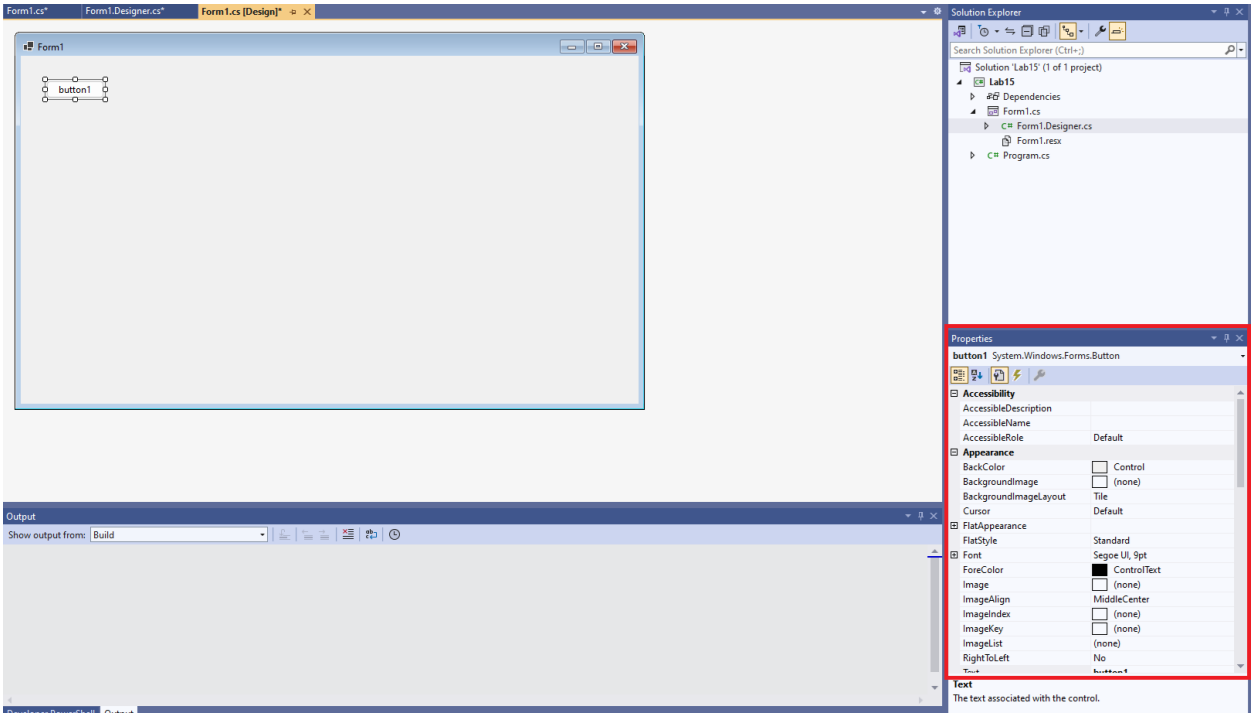


To add a GUI element, simply drag it from your Toolbox into your form in the middle of the screen.



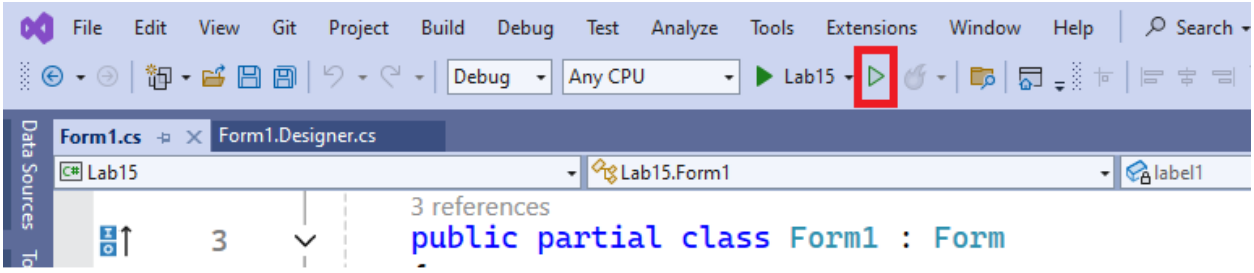
Editing GUI element properties

All GUI elements have properties. You can edit them by first clicking the GUI element on the form and then looking at the Properties window (if you cannot see the Properties window, check the section “Ensuring Tools and Properties is visible”).



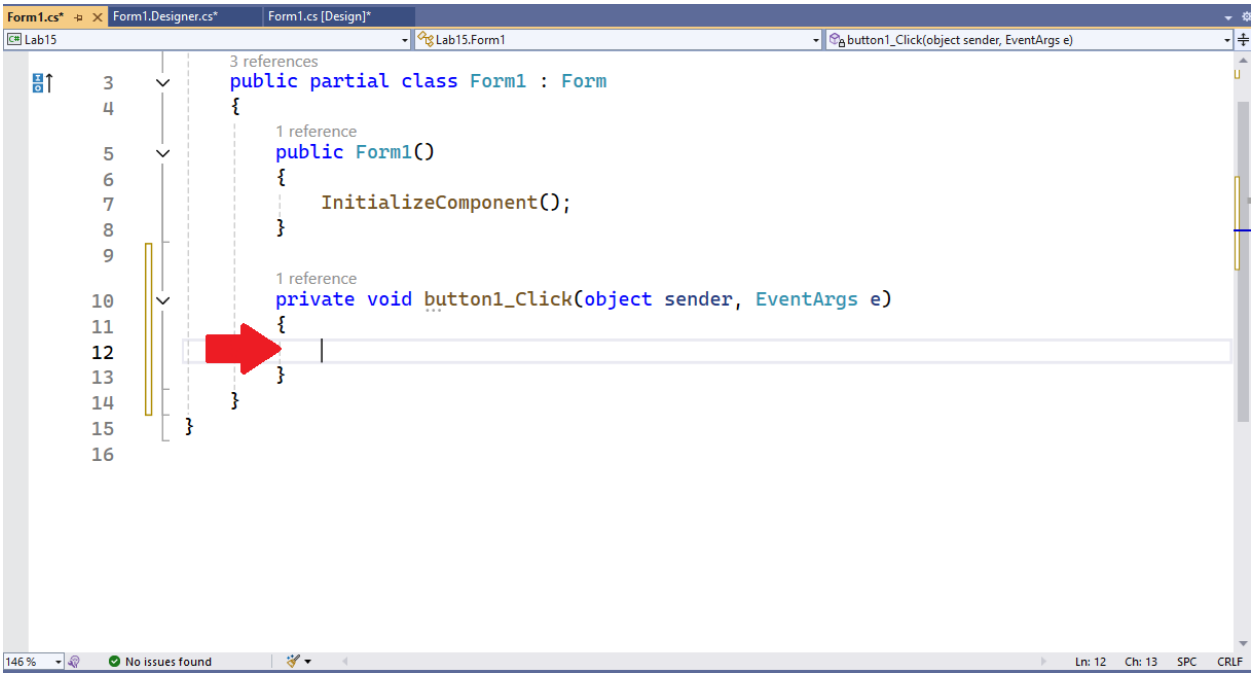
Simply change the properties of your GUI elements as instructed by the lab. Any properties not mentioned in the lab instructions should ideally be left at their default values.

You can test your GUI at any time by running the project. Click the light-green play button at the top of your IDE.



Adding functionality to GUI elements

To add functionality to a GUI element, double click the element. A new window will appear, and your cursor will be located inside a method. Whatever code you add in this method will be executed when the GUI element is clicked. **Avoid deleting any code you haven't written as that will likely break your project**.

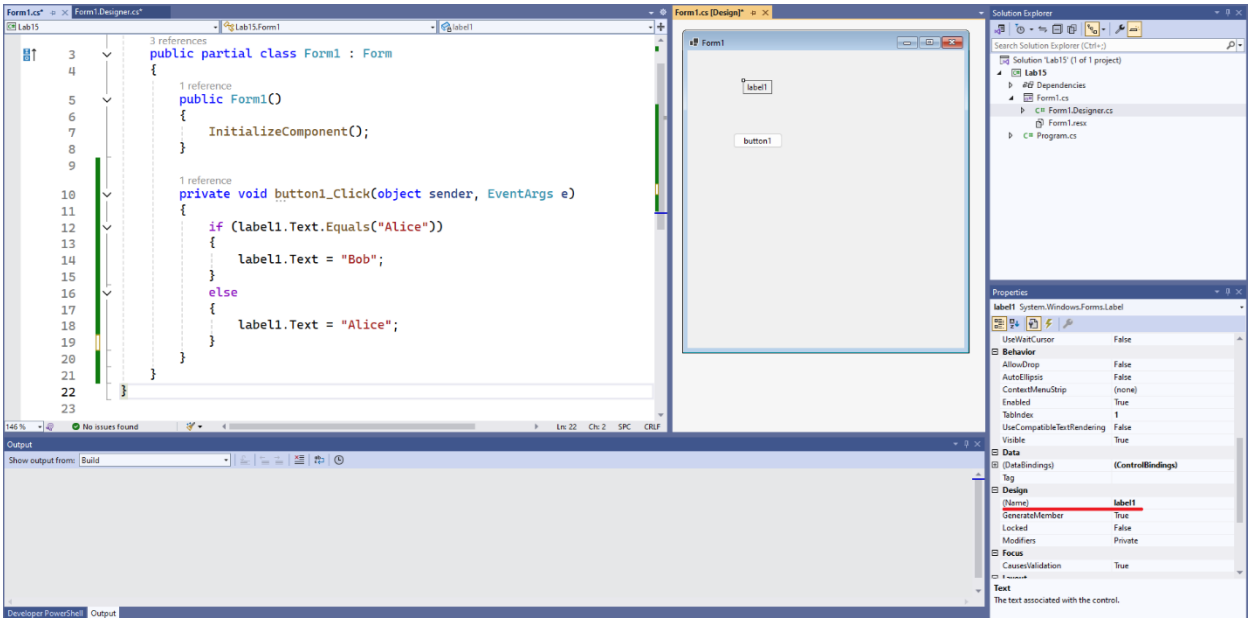


Every GUI element has a Name, which can be found in its Properties window under Design → (Name). This (Name) is how the code identifies the underlying object associated with the element on the form.

Note that (Name) is a different field than Appearance → Text. The Text field holds what text the element displays.

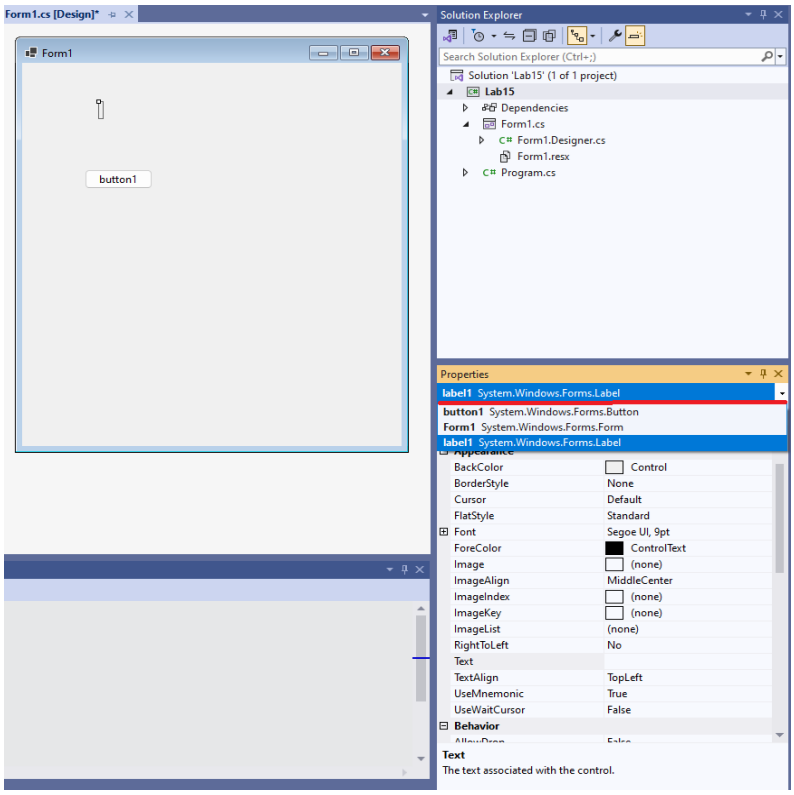
A Label element has been added to the form below. When selecting the Label, we can see that its (Name) is “label1”. We can then add code in the Button’s click method to change the text on the Label when the button is clicked.

In the example below, the Label will initially start with the words “label1”. If clicked, it will become “Alice” and, if clicked again, it will become “Bob”. Further clicks will alternate the Label’s text between “Alice” and “Bob”.



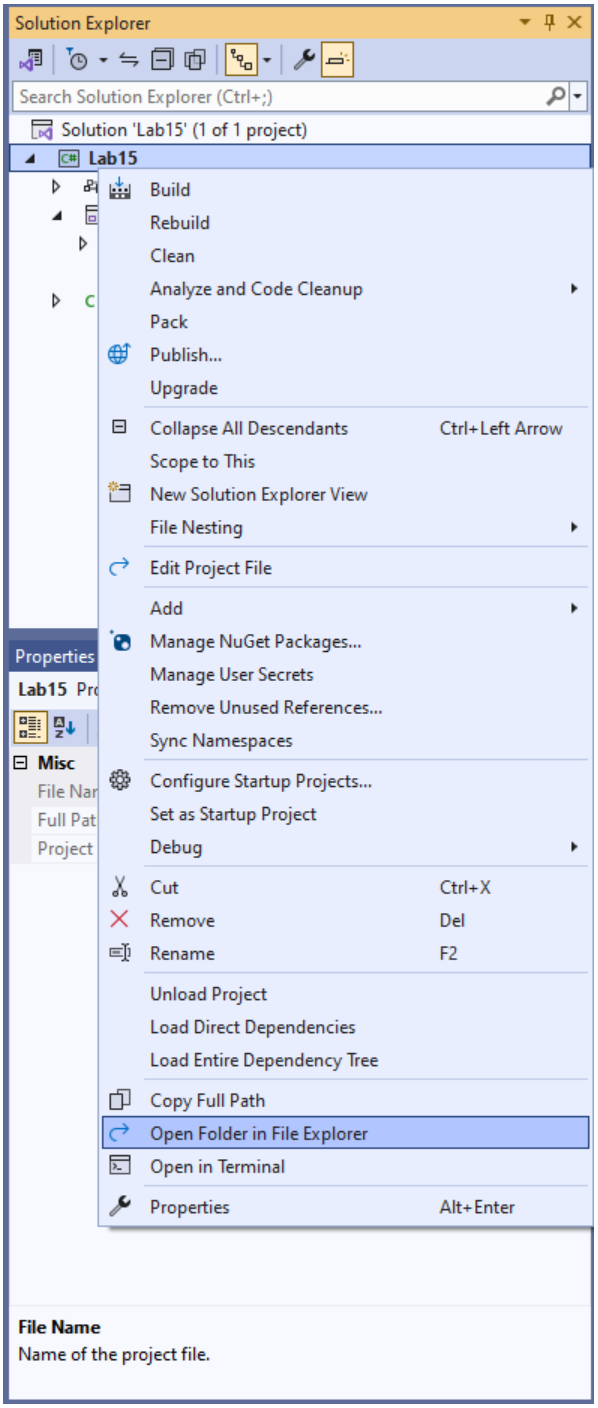
If you, at any point, misplace a GUI element and are having trouble finding it (**e.g.: a Label whose Text field is an empty string**), do the following:

- Click anywhere on your form in the Designer view.
- In the Properties window, click the Form’s name at the top (it will probably “Form1 System.Windows.Forms.Form”)
- In the dropdown menu, select the element you are trying to locate



Retrieving project files

Once you are done writing your solution, you must then retrieve your project files so they can be submitted via Gradescope. To do so, right click your project's name on the Solution Explorer and select "Open Folder in File Explorer".



A window will open showing your files. Simply drag and drop them onto Gradescope. **Remember to only submit the files requested by the lab deliverables.**