GRAND CANYON UNIVERSITY™

# CST-150 Activity 1 Guide

## Contents

# Parts 1 and 2

# Tools Installation

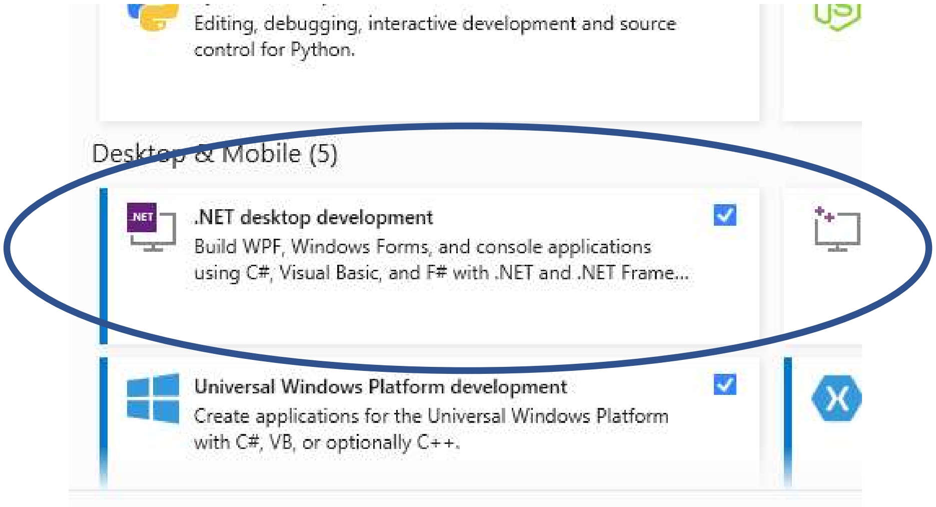
## Overview

In this activity, students will install the development tool and validate the tool installation with a simple application.

## Execution

Execute this assignment according to the following guidelines:

1. Install Visual Studio
   a. Be sure to download or update to the version specified by your instructor.
   b. Mac – Be sure everyone has either Parallels or VMWare Fusion installed.
   c. Visit "Student Success Center" in the digital classroom and search "Visual Studio."
   d. Install Visual Studio.
   e. Install the required workload ".NET desktop development" as shown in Figure 1.



Editing, debugging, interactive development and source control for Python.

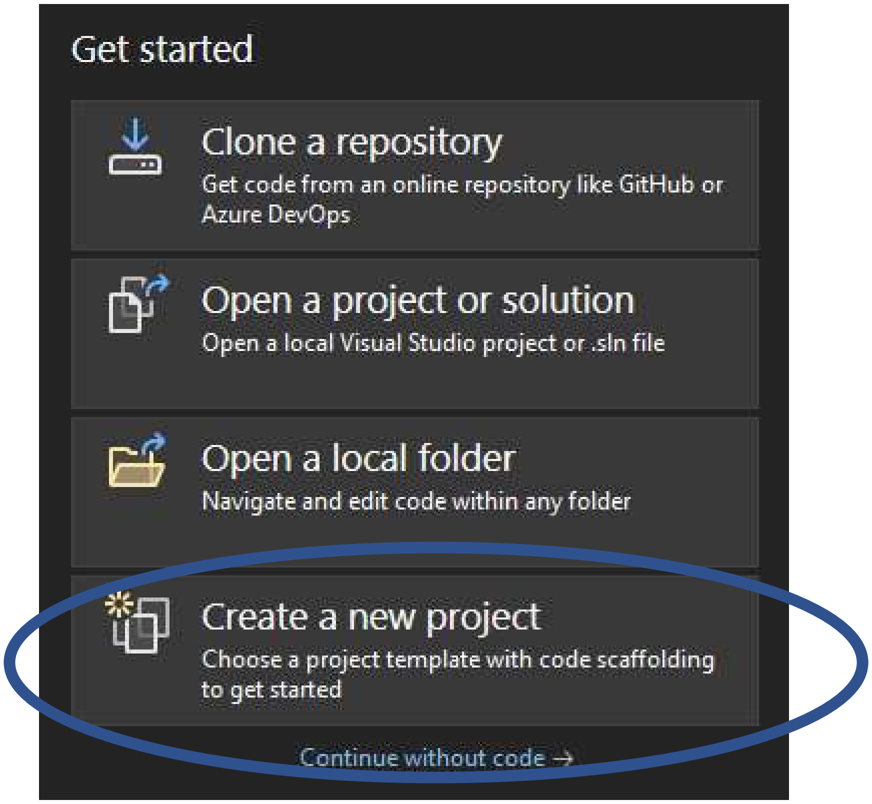
Desktop & Mobile (5)

.NET desktop development ☑
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...

Universal Windows Platform development ☑
Create applications for the Universal Windows Platform with C#, VB, or optionally C++.

*Figure 1: Workload to Include*

2. Write Your First Program
   a. Start Visual Studio and select "Create a new project" as shown in Figure 2.



Get started

Clone a repository
Get code from an online repository like GitHub or Azure DevOps

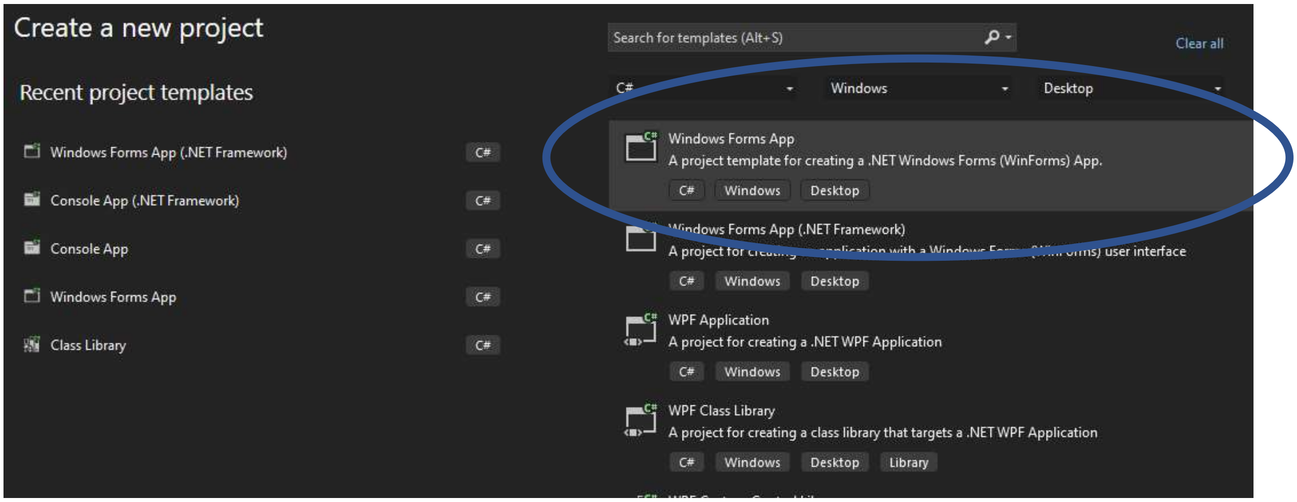
Open a project or solution
Open a local Visual Studio project or .sln file

Open a local folder
Navigate and edit code within any folder

Create a new project
Choose a project template with code scaffolding to get started

*Figure 2: Select "Create a new project."*

b. Select the Project Template "WindowsFormsApp" as shown in Figure 3.



Create a new project

Recent project templates

- Windows Forms App (.NET Framework) C#
- Console App (.NET Framework) C#
- Console App C#
- Windows Forms App C#
- Class Library C#

Windows Forms App
A project template for creating a .NET Windows Forms (WinForms) App.
C# Windows Desktop

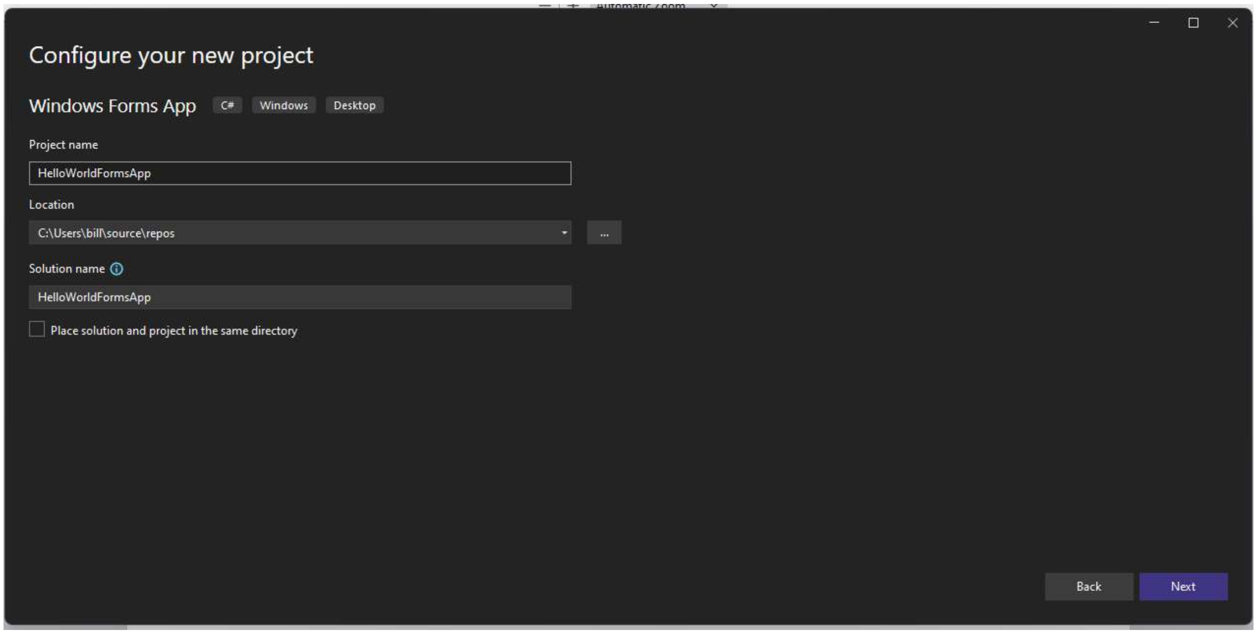
Windows Forms App (.NET Framework)
A project for creating an application with a Windows Forms (WinForms) user interface
C# Windows Desktop

WPF Application
A project for creating a .NET WPF Application
C# Windows Desktop

WPF Class Library
A project for creating a class library that targets a .NET WPF Application
C# Windows Desktop Library

*Figure 3: Select the Project Template.*

c. Enter the Project name "HelloWorldFormsApp," as shown in Figure 4, followed by the "Next" button.



Configure your new project

Windows Forms App

Project name: HelloWorldFormsApp

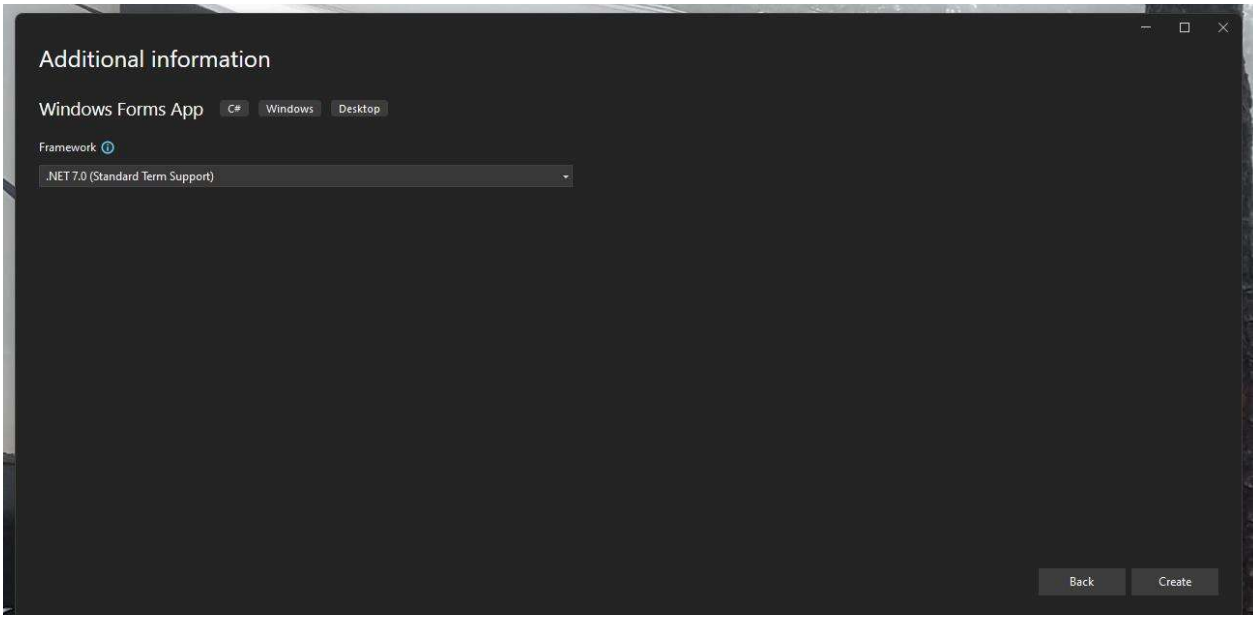
Location: C:\Users\bill\source\repos

Solution name: HelloWorldFormsApp

□ Place solution and project in the same directory

*Figure 4: Enter Project Name.*

d. Select the Framework as shown in Figure 5, followed by the "Create" button.



Additional information

Windows Forms App

Framework: .NET 7.0 (Standard Term Support)

*Figure 5: Select Framework.*

e. In the form, drag a button and label into the form from the toolbox as is shown in Figure 6.

f. For the button, name it "btnHelloButtonTest" in the properties using camelCasing naming convention.

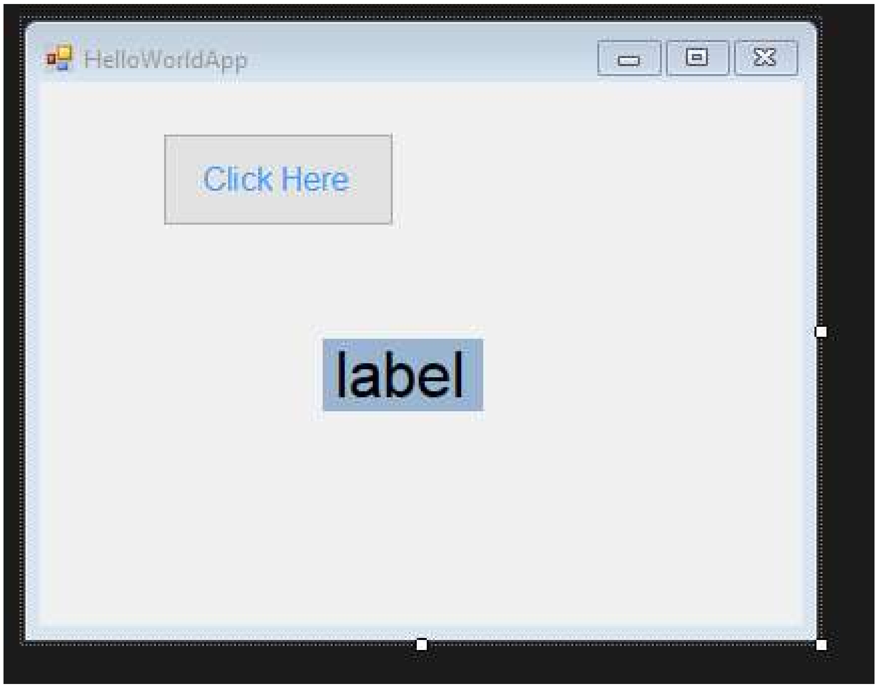
g. Change the font type and size in the properties to "Arial 12pt."

h. Update the Forecolor property to "MenuHighlight."

i. Grab the corner of the button and enlarge it.

j. Change for the form text to "HelloWorldWork".

k. **Note**: If the toolbox is empty, try right-clicking in the toolbox and reset. If that does not work, repair, and install program.



7450

*Figure 6: Form Design.*

l. In the form, drag a label into the form from the toolbox as shown in Figure 6.

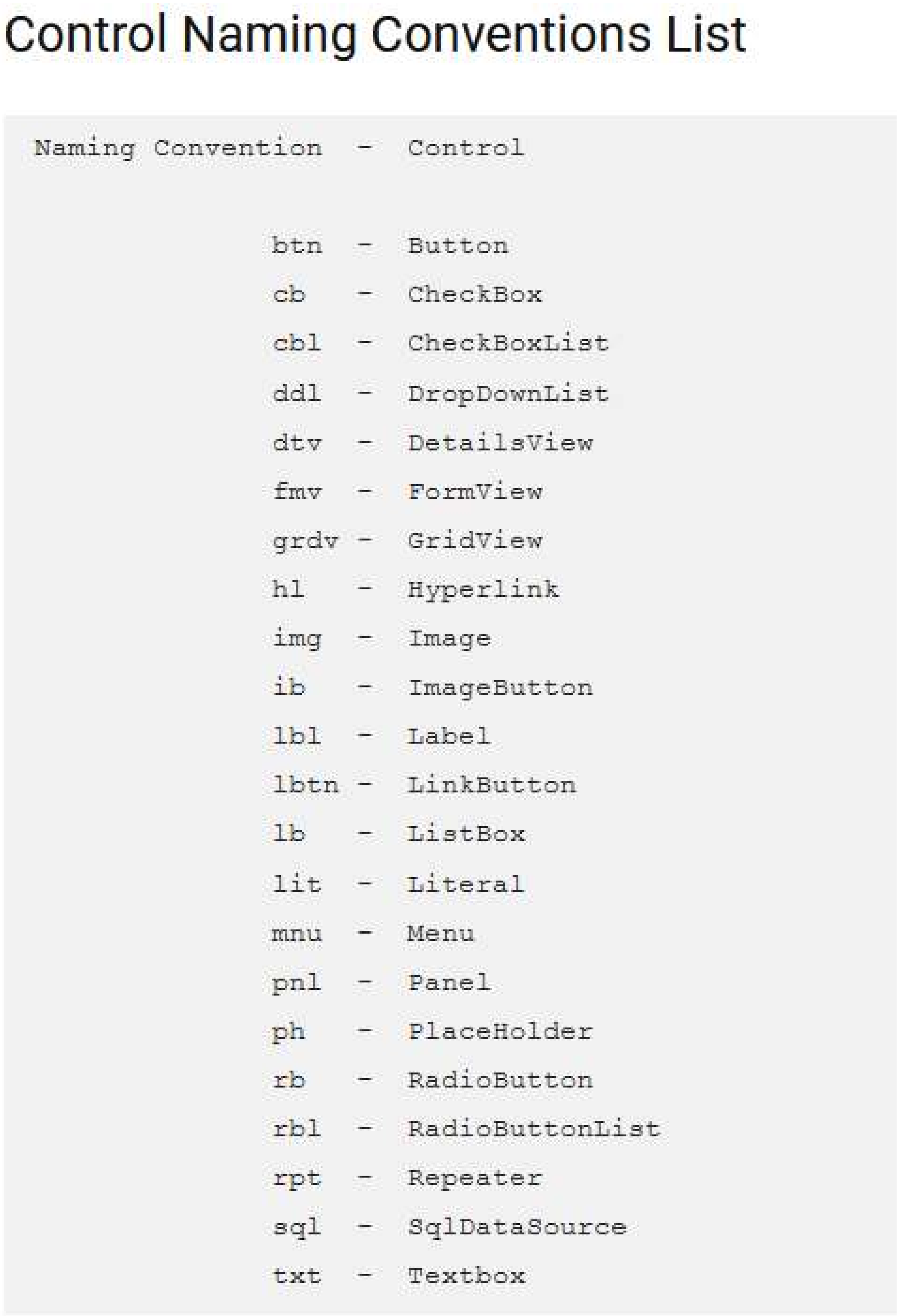
m. Name the label "lblHelloWorldLabel."

n. Change the BackColor property to "ActiveCaption."

o. Update the Font to "Arial, 24pt."

p. Change the ForeColor to "ControlText."

q. The following Figure 7 shows how to label the prefix for controls:



## Control Naming Conventions List

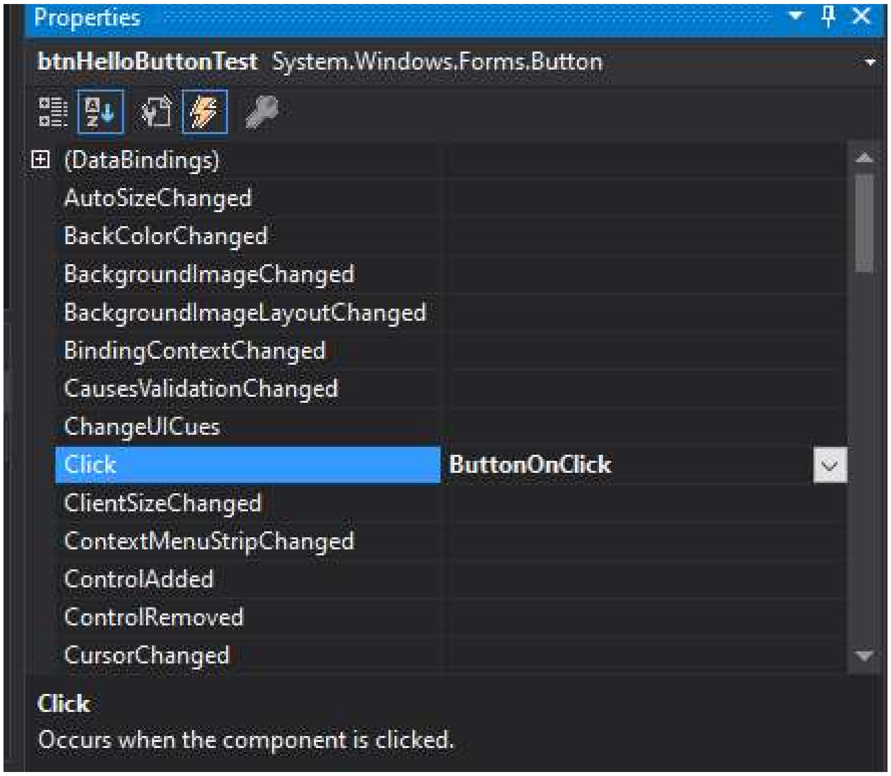
| Naming Convention | - | Control |
|---|---|---|
| btn | - | Button |
| cb | - | CheckBox |
| cbl | - | CheckBoxList |
| ddl | - | DropDownList |
| dtv | - | DetailsView |
| fmv | - | FormView |
| grdv | - | GridView |
| hl | - | Hyperlink |
| img | - | Image |
| ib | - | ImageButton |
| lbl | - | Label |
| lbtn | - | LinkButton |
| lb | - | ListBox |
| lit | - | Literal |
| mnu | - | Menu |
| pnl | - | Panel |
| ph | - | PlaceHolder |
| rb | - | RadioButton |
| rbl | - | RadioButtonList |
| rpt | - | Repeater |
| sql | - | SqlDataSource |
| txt | - | Textbox |

*Figure 7: Naming Conventions.*

r. Now, let's write the code behind.

s. Be sure the button is selected and in the properties window select the events tab.

t. In the Click type "ButtonOnClick" as shown in Figure 8 and hit enter.



Properties

btnHelloButtonTest System.Windows.Forms.Button

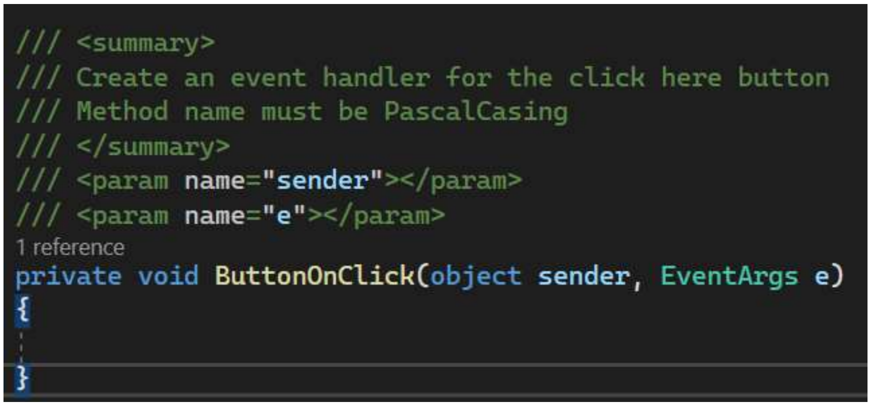
⊞ (DataBindings)
AutoSizeChanged
BackColorChanged
BackgroundImageChanged
BackgroundImageLayoutChanged
BindingContextChanged
CausesValidationChanged
ChangeUICues
Click ButtonOnClick
ClientSizeChanged
ContextMenuStripChanged
ControlAdded
ControlRemoved
CursorChanged

**Click**
Occurs when the component is clicked.

*Figure 8: Click Event Handler*

u. This will create the method using the proper naming convention of PascalCasing as shown in Figure 9.

v. Add in the summary comments "///" for the method as shown in Figure 9.

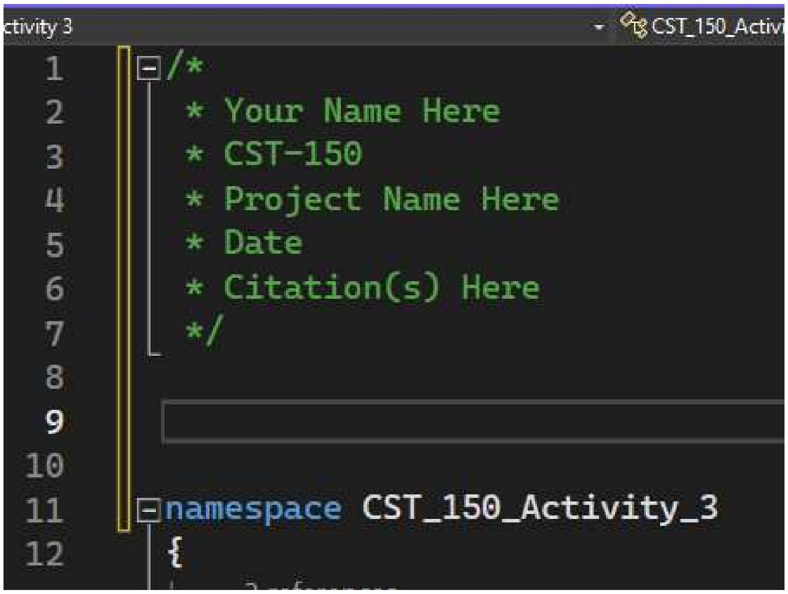


```
/// <summary>
/// Create an event handler for the click here button
/// Method name must be PascalCasing
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
1 reference
private void ButtonOnClick(object sender, EventArgs e)
{

}
```

*Figure 9: Method Level Comments.*

w. Whenever a new cs page is started, the first item we complete is to add the citations at the very top as shown in Figure 10 that start with "/*" and end with "*/".

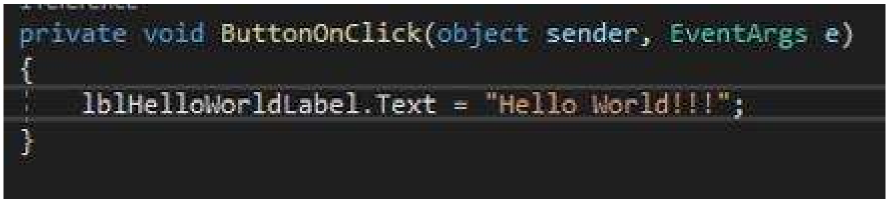


```
/*
 * Your Name Here
 * CST-150
 * Project Name Here
 * Date
 * Citation(s) Here
 */


namespace CST_150_Activity_3
{
```

*Figure 10: Citation at the top.*

x. Insert the block of code between the curly braces to show "Hello World" as it is shown in Figure 11.



```
private void ButtonOnClick(object sender, EventArgs e)
{
    lblHelloWorldLabel.Text = "Hello World!!!";
}
```

*Figure 11: Code to Display Hello World*

y. Now, run the code selecting the icon as shown in Figure 12 and see the results.

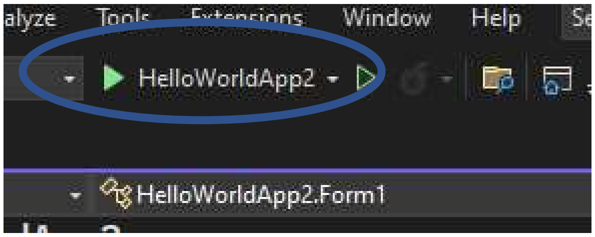


*Figure 12: Run the Code.*

3. Draw a flowchart to visually describe the flow of the program.
4. Submit the activity as described in the digital classroom.

# Part 3

## Earth Weight To Mars

### Overview

Write a Windows Forms Application that prompts the user to enter a value and then converts the value to some other type of units.

### Execution

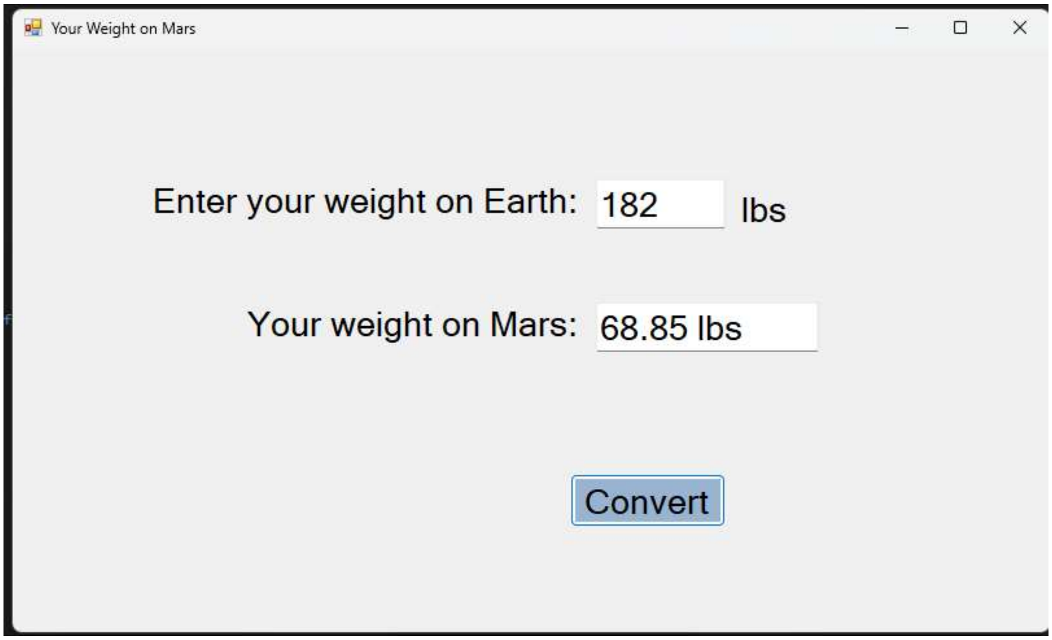
Execute this assignment according to the following guidelines:

1. Functional Specifications

   a. Click on the "Convert" button and the value in the textbox for "Enter your weight on Earth" will be used to calculate "Your weight on Mars" as shown in Figure 13.

   b. The conversion will be displayed in the textbox just to the right of the text "Mars."

   c. Hide "Your weight on Mars" text and textbox until after clicking the button.



Your Weight on Mars

Enter your weight on Earth: 182 lbs

Your weight on Mars: 68.85 lbs

*Figure 13: Activity 1 Example*

2. Technical Specifications

   d. Me = We / ge -- Mass Earth = Weight Earth / Gravitational Acceleration on Earth

   e. ge = 9.81 $m/s^2$ -- meters / seconds squared

   f. gm = 3.711 $m/s^2$ – gm is Gravitational Acceleration on Mars

   g. Wm = Me * gm

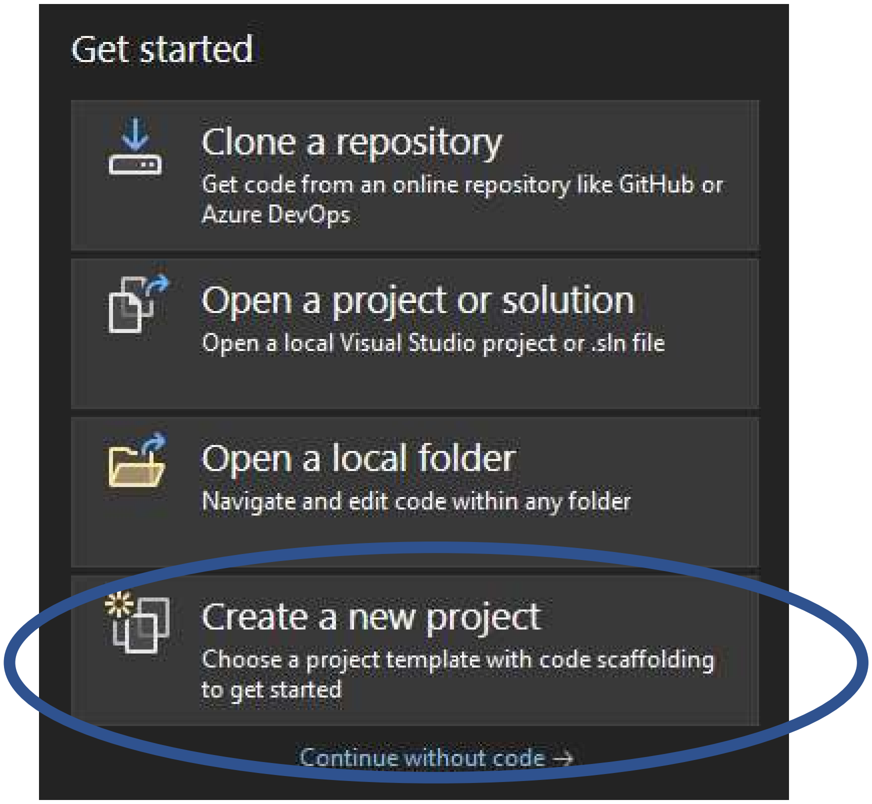
h. Full algorithm shown in Figure 14.

$$\text{Weight On Mars} = \frac{\text{Weight On Earth}}{9.81\ \text{m/s}^2} \times 3.711\ \text{m/s}^2$$

*Figure 14: Algorithm to Calculate Weight on Mars*

3. Start Visual Studio

a. Start Visual Studio and select "Create a new project" as is shown in Figure 15.



Get started

Clone a repository
Get code from an online repository like GitHub or Azure DevOps

Open a project or solution
Open a local Visual Studio project or .sln file

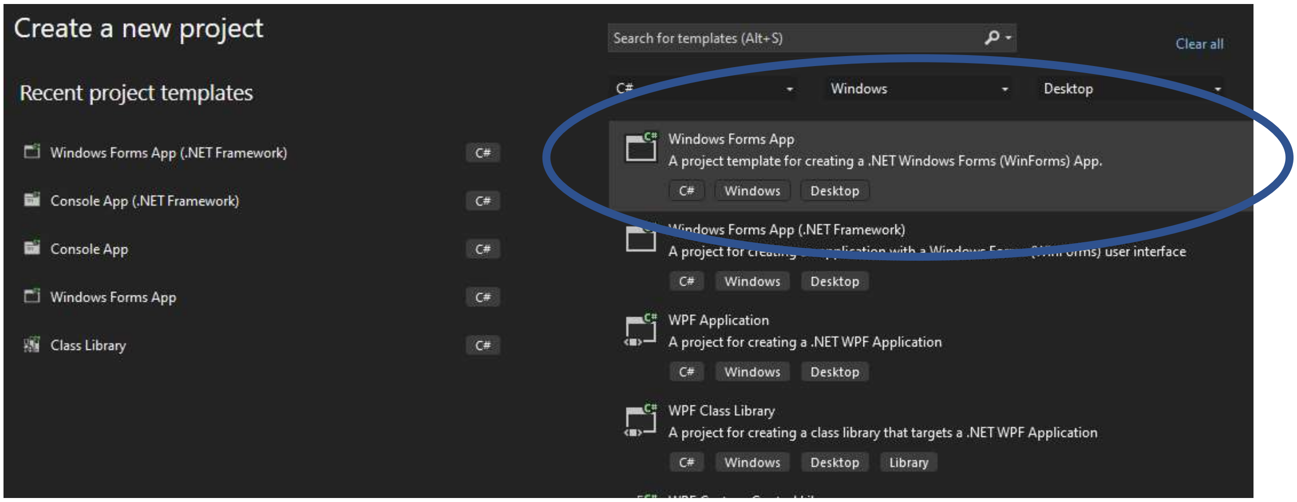
Open a local folder
Navigate and edit code within any folder

Create a new project
Choose a project template with code scaffolding to get started

Continue without code →

*Figure 15: Select "Create a new project."*

b. Select the Project Template "WindowsFormsApp" as shown in Figure 16.


Create a new project

Recent project templates

- Windows Forms App (.NET Framework) C#
- Console App (.NET Framework) C#
- Console App C#
- Windows Forms App C#
- Class Library C#

Windows Forms App
A project template for creating a .NET Windows Forms (WinForms) App.
C# Windows Desktop

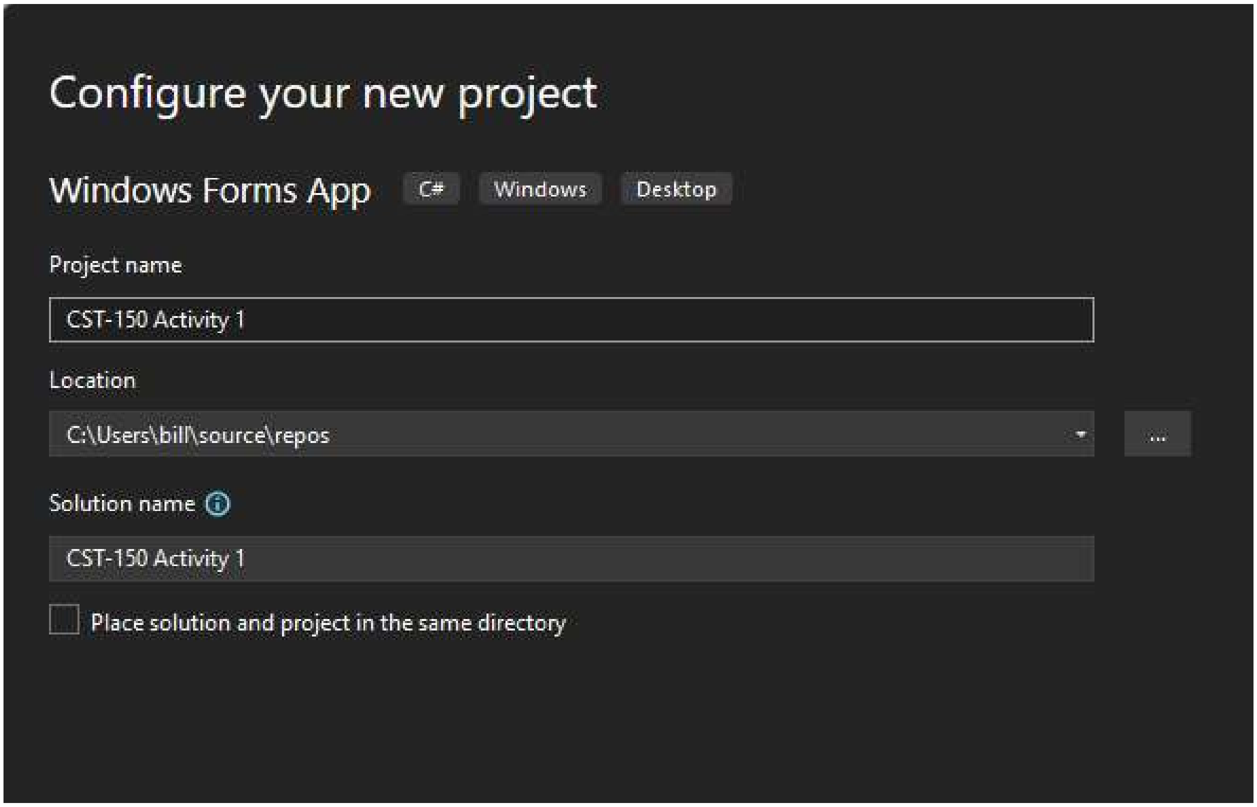
Windows Forms App (.NET Framework)
A project for creating an application with a Windows Forms (WinForms) user interface
C# Windows Desktop

WPF Application
A project for creating a .NET WPF Application
C# Windows Desktop

WPF Class Library
A project for creating a class library that targets a .NET WPF Application
C# Windows Desktop Library

*Figure 16: Select the Project Template.*

[redacted]. Enter the Project name: "CST-150 Activity 1" as shown in Figure 17 followed by the "Next" button. Be sure the check box is not checked for "Place solution and project in the same directory." It will be discovered having the .sln solution file is more convenient if it is not in the same directory as all the project files.


Configure your new project

Windows Forms App C# Windows Desktop

Project name

CST-150 Activity 1

Location

C:\Users\bill\source\repos

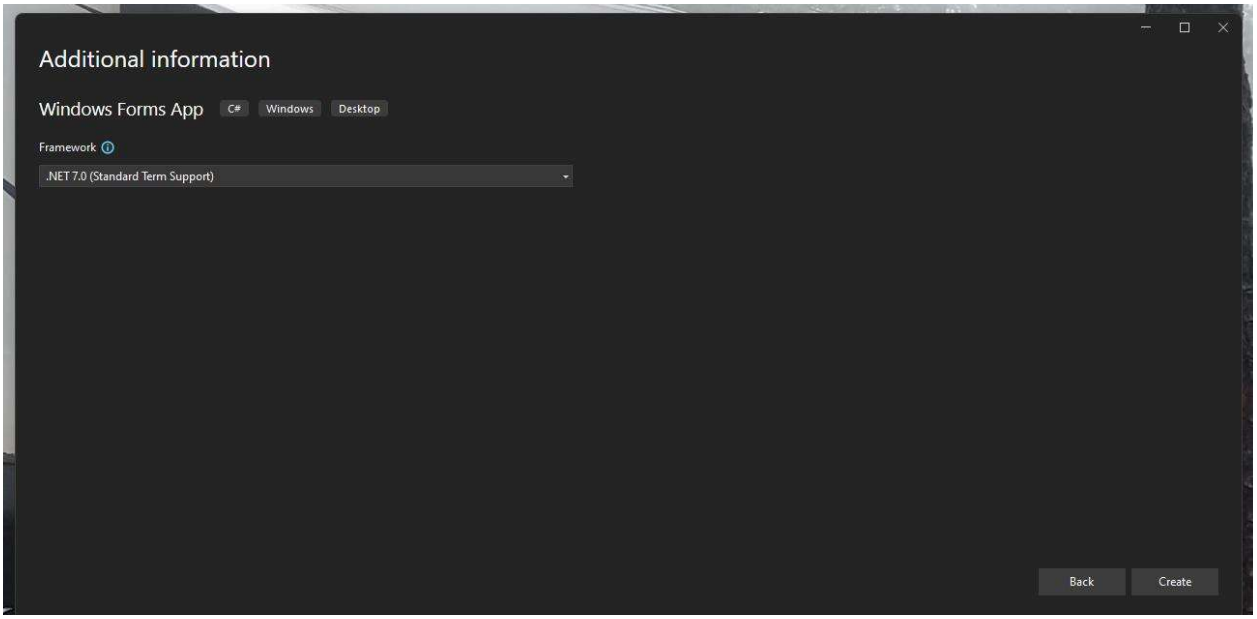
Solution name

CST-150 Activity 1

□ Place solution and project in the same directory

*Figure 17: Enter Project Name.*

d. Select the Framework as is shown in Figure 18 followed by the "Create" button. If the Framework shown in Figure 6 is not available, re-run the Visual Studio installer and select the correct options.



Additional information

Windows Forms App

Framework: .NET 7.0 (Standard Term Support)

*Figure 18: Select Framework.*

5. Front-End Design

a. Layout the Form for the project as shown in Figure 19.

b. Place a label on the form and change the text property to "Enter your weight on Earth."

c. Update the name of the label to "lblEarth."

d. Place a text box just to right of the label on the form and update the name to "txtEarthWeight."

e. Place another label on the form and change the text property to "Your weight on Mars."

f. Update the name of the label to "lblMars."

g. Add a new label to the form just to the right of the last label. We will use this label to show our weight on Mars.

h. Name this label "lblMarsWeight."

i. Add a button to the form and change the text of the button to "Convert."

j. Name the button "btnConvert" using camelCasing.

k. Update the form text so the form now shows, "Your Weight on Mars" in the upper left corner of the form.

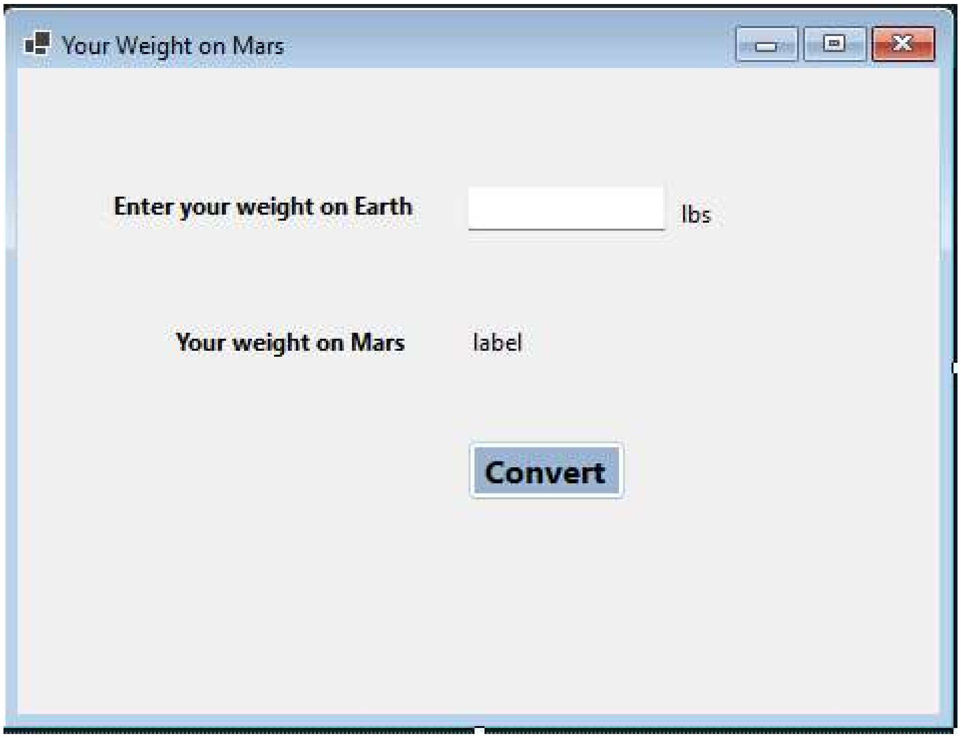
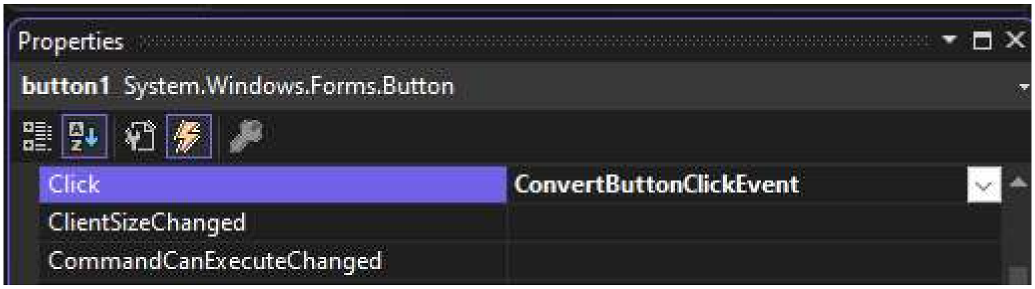


*Figure 19: Front-End Design*

6. Back-End Code

a. On the form, create a Click Event Handler for the button as is shown in Figure 20.

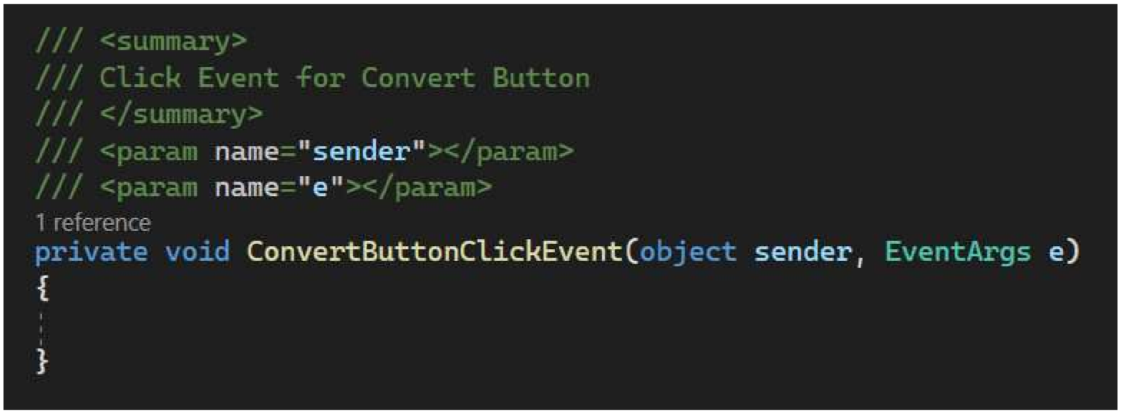


button1 System.Windows.Forms.Button

Click ConvertButtonClickEvent

*Figure 20: Click Event*

b. The Click Event Handler should look like this when complete as shown in Figure 21.

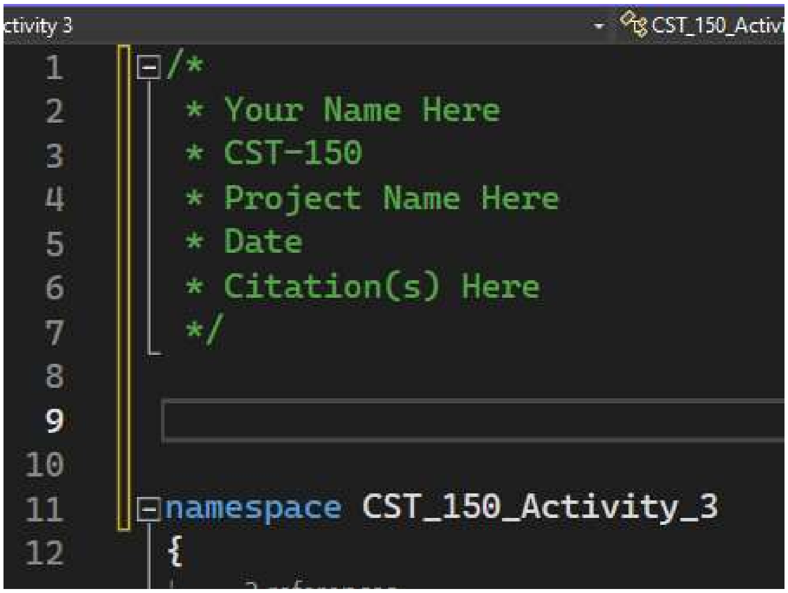


```
/// <summary>
/// Click Event for Convert Button
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
1 reference
private void ConvertButtonClickEvent(object sender, EventArgs e)
{

}
```

*Figure 21: Button Click Event Handler Method in Code Behind*

c. Whenever a new cs page is started, the first item we do is add the citations at the very top as shown in Figure 10.

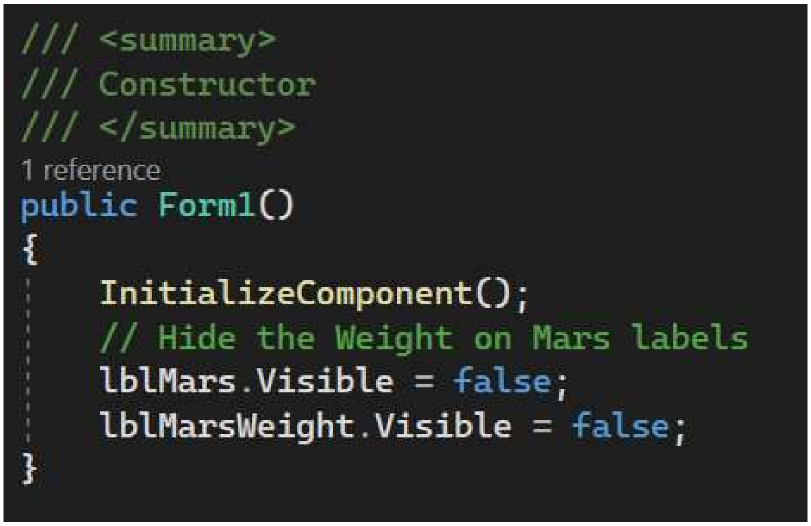


```
/*
 * Your Name Here
 * CST-150
 * Project Name Here
 * Date
 * Citation(s) Here
 */


namespace CST_150_Activity_3
{
```

*Figure 22: Citation at the top.*

d. Make the properties not visible for the Mars label and textbox as shown in Figure 23.



```
/// <summary>
/// Constructor
/// </summary>
1 reference
public Form1()
{
    InitializeComponent();
    // Hide the Weight on Mars labels
    lblMars.Visible = false;
    lblMarsWeight.Visible = false;
}
```

*Figure 23: Hide the Form Controls*

e. Run the app and verify the results making sure it looks like Figure 24.

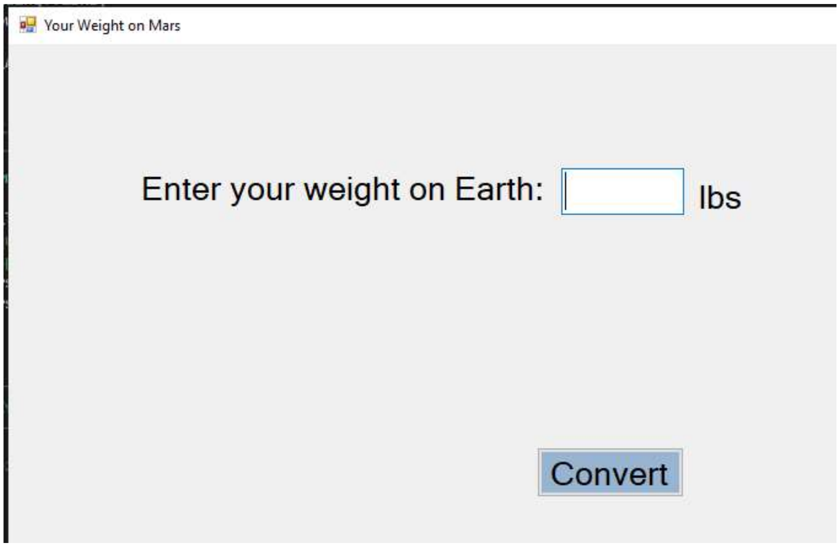
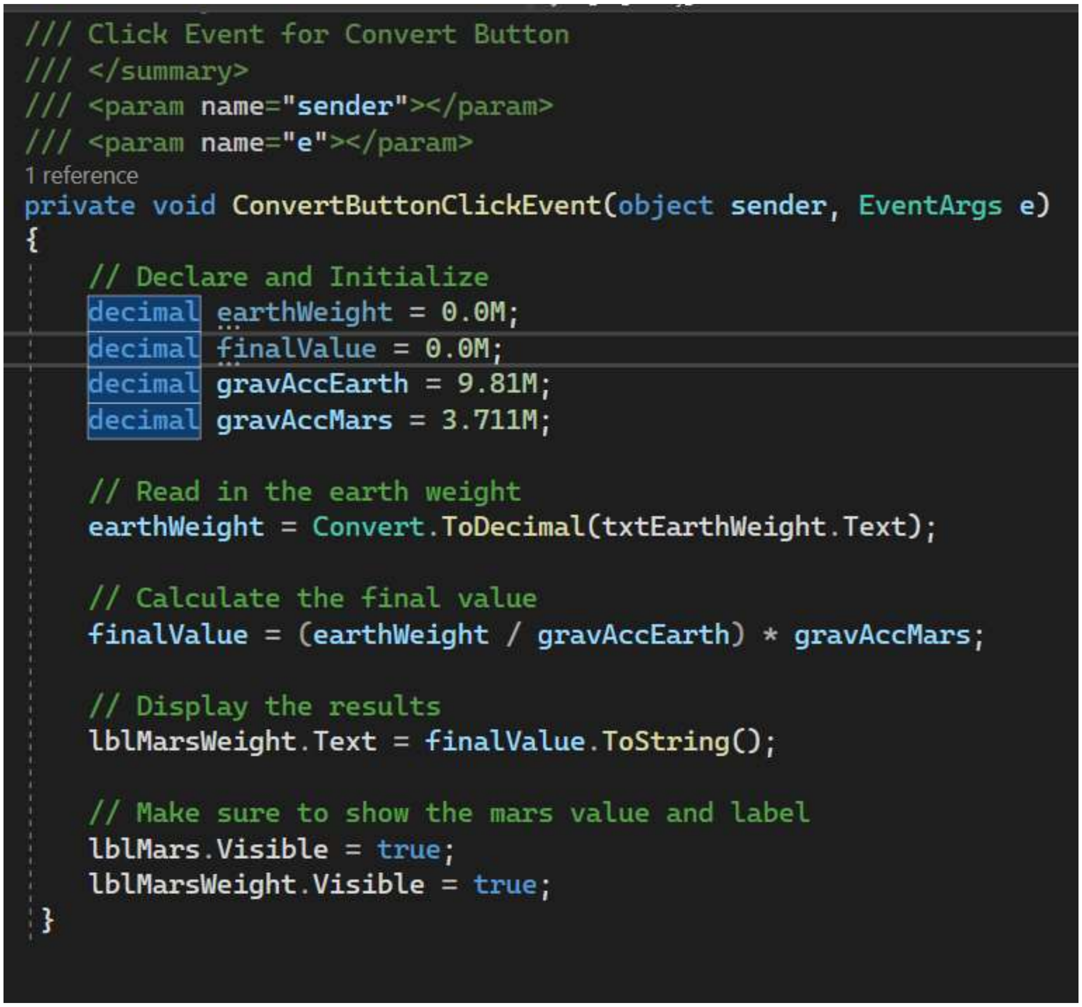


*Figure 24: Result Should Look Like This.*

f. Now, add the code to the Button Click Event Handler as shown in Figure 25 that converts the earth weight to mars. Do not forget the "M" that is required to designate a decimal in C#.

g. To display the results in the lblMarsWeight.Text, it is required to make the decimal finalValue into a string, so it can be accepted by the label. Do this by using the method .ToString() that makes the decimal a string.

h. Then when done with the algorithm, make the labels visible.



```
/// Click Event for Convert Button
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
1 reference
private void ConvertButtonClickEvent(object sender, EventArgs e)
{
    // Declare and Initialize
    decimal earthWeight = 0.0M;
    decimal finalValue = 0.0M;
    decimal gravAccEarth = 9.81M;
    decimal gravAccMars = 3.711M;

    // Read in the earth weight
    earthWeight = Convert.ToDecimal(txtEarthWeight.Text);

    // Calculate the final value
    finalValue = (earthWeight / gravAccEarth) * gravAccMars;

    // Display the results
    lblMarsWeight.Text = finalValue.ToString();

    // Make sure to show the mars value and label
    lblMars.Visible = true;
    lblMarsWeight.Visible = true;
}
```

*Figure 25: Code that has the logic to convert earth to mars weight.*

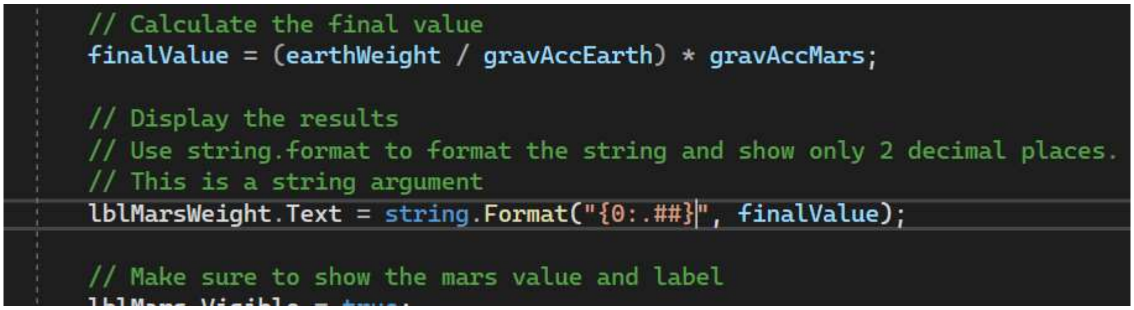
7. Format the Results

    a. Update the line "lblMarsWeigth.Text = finalValue.ToString()" so the results are properly formatted as shown in Figure 26.

    b. Format the Mars weight number.

    c. {0 is call "Zero Placeholder" replaces the zero with the corresponding digit if it exists. If you put {00000: then there would be leading 0s left in the format).

    d. Two decimal places (.##) replaces the "#" symbol with the corresponding digit if one is present; otherwise, no digit appears in the result string.



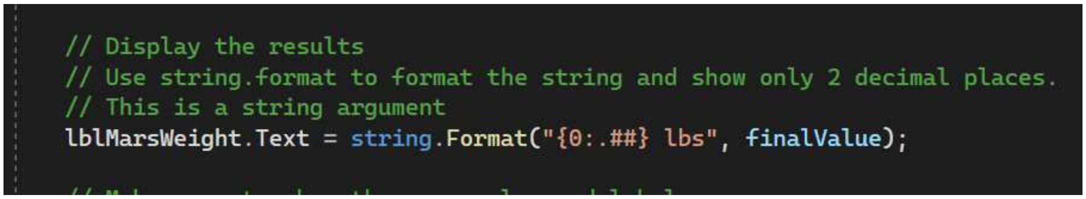
```
// Calculate the final value
finalValue = (earthWeight / gravAccEarth) * gravAccMars;

// Display the results
// Use string.format to format the string and show only 2 decimal places.
// This is a string argument
lblMarsWeight.Text = string.Format("{0:.##}", finalValue);

// Make sure to show the mars value and label
lblMars.Visible = true;
```

*Figure 26: Format the Result*

e. Now, add the units as is shown in Figure 27.

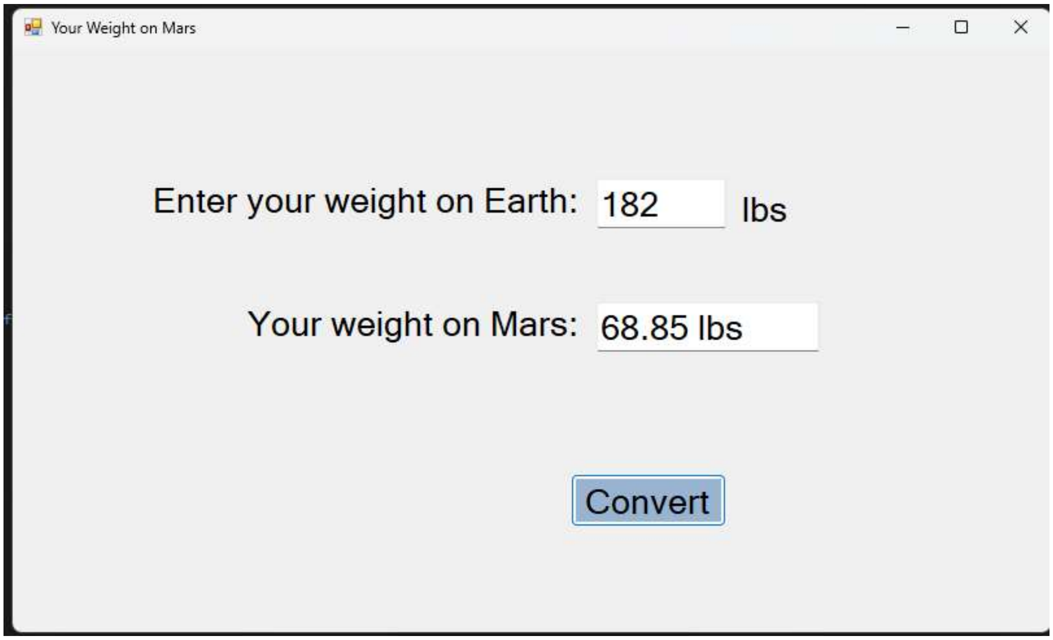


```
// Display the results
// Use string.format to format the string and show only 2 decimal places.
// This is a string argument
lblMarsWeight.Text = string.Format("{0:.##} lbs", finalValue);

// Make sure to show the mars value and label
```

*Figure 27: Add Units*

f. The end result should look like Figure 28, with the exception that yours must have the earth weight as provided by your instructor to get full points for the application.



Your Weight on Mars

Enter your weight on Earth: 182 lbs

Your weight on Mars: 68.85 lbs

*Figure 28: End Result*

[redacted]. Submit the Activity as described in the digital classroom.