

Name: Eric Gathinji

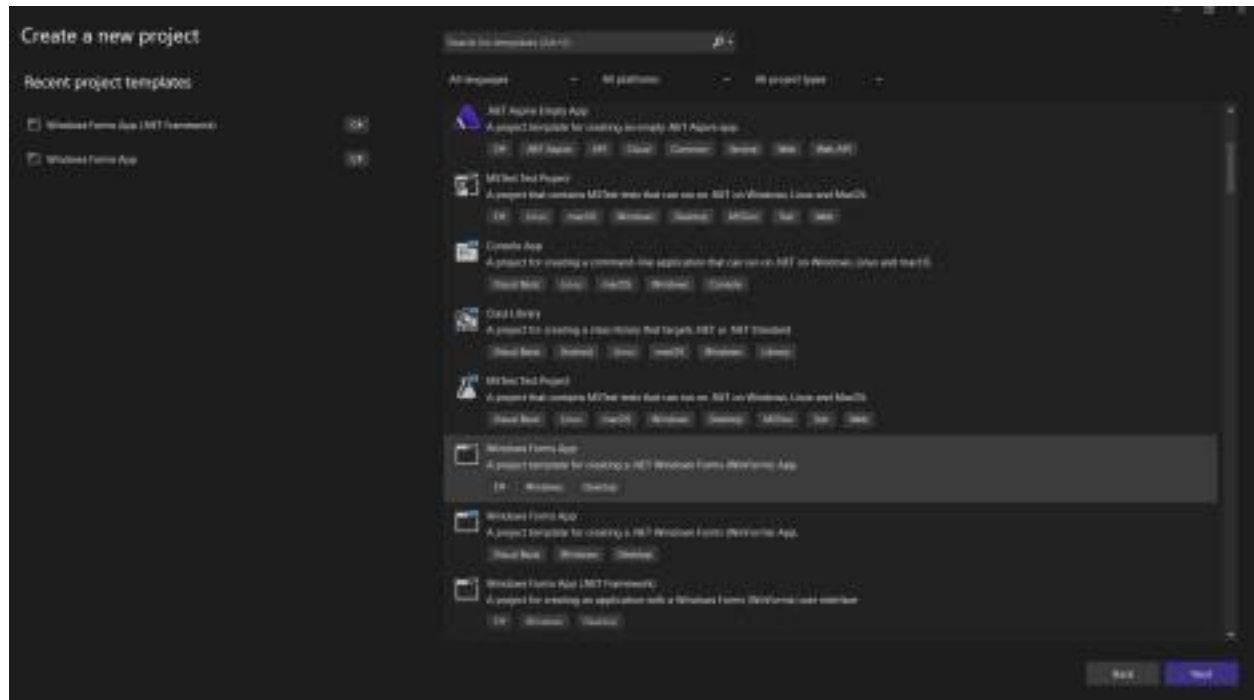
Programming in C# CST-150-0500

Grand Canyon University

23rd June 2025

Activity 1

Part 1



Visual Studio Installer with the ".NET desktop development" workload selected.

This screenshot shows the Visual Studio Installer during the setup process, where the required workload for Windows Forms development is selected.

The ".NET desktop development" workload is essential because it provides all the necessary tools, libraries, and templates to build Windows desktop applications using C#. Selecting this workload ensures that the developer can create projects like Windows Forms Apps, which are required for this course.

Without this step, Visual Studio would not support form-based graphical user interfaces. This configuration is crucial for completing the CST-150 Activity 1 assignment.

Part 2.

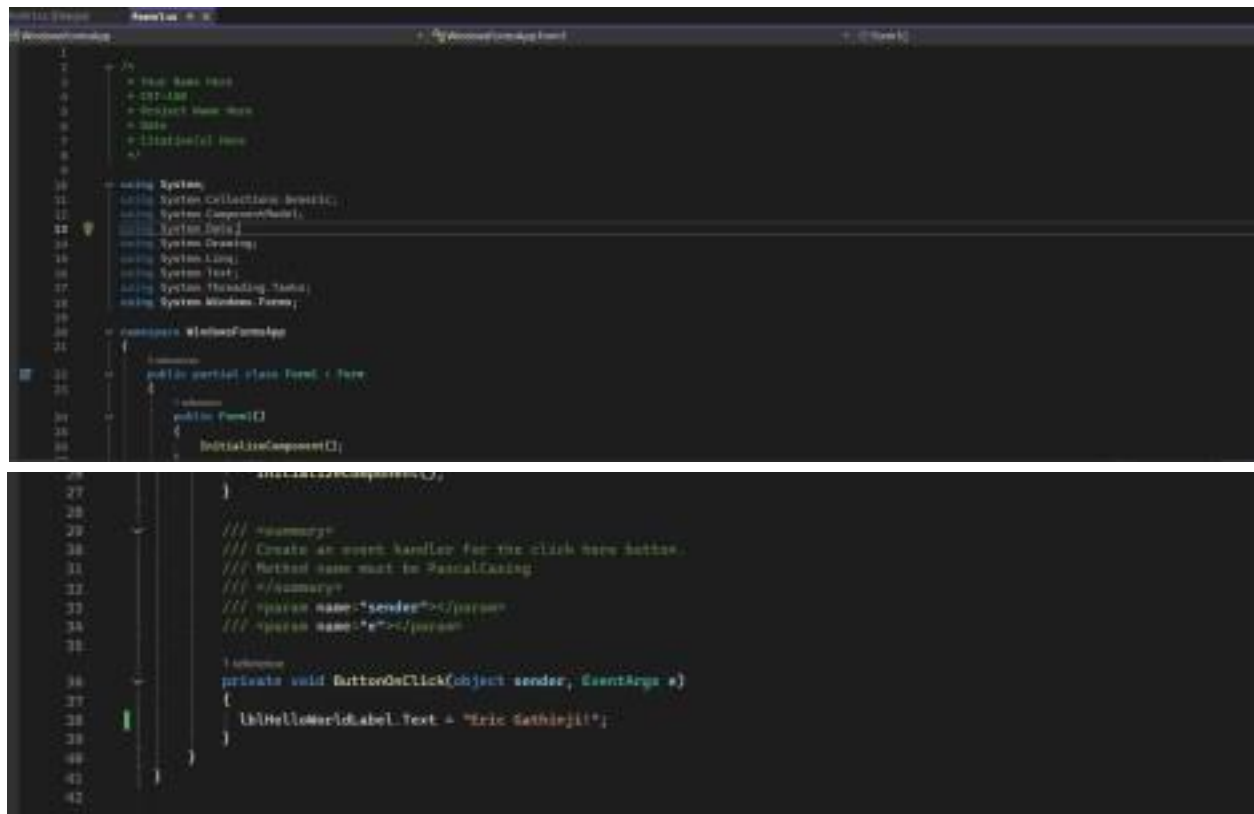
Flowchart representing the application's behavior upon button click **Explanation:**

This flowchart models the logical sequence of the Windows Forms application. The application starts by displaying the form with a button and label. When the user clicks the button, the label is updated with a message. This visual representation helps understand the program's control flow before implementation. It ensures clarity in planning and aligns with best practices in software development.



Windows Form displaying the designer view with my name on the form.

This screenshot shows the Windows Form in design mode. A label control has been added and populated with my name to personalize the interface. Additional controls like a button and another label are placed on the form to support the functionality of displaying a message when the button is pressed. The layout and naming follow C# naming conventions such as `lblHelloWorldLabel` and `btnHelloButtonTest`. This graphical interface is user-friendly and easy to interact with.



```
1
2
3 // This file contains the logic for the application.
4
5 // Project Name: MyForm
6
7 // Initial setup
8
9
10 using System;
11 using System.Collections.Generic;
12 using System.ComponentModel;
13 using System.Drawing;
14 using System.Linq;
15 using System.Text;
16 using System.Threading.Tasks;
17 using System.Windows.Forms;
18
19 namespace WindowsFormsApp1
20 {
21     public partial class Form1 : Form
22     {
23         public Form1()
24         {
25             InitializeComponent();
26         }
27
28         /// <summary>
29         /// Create an event handler for the click here button.
30         /// Method name must be PascalCasing.
31         /// </summary>
32         /// <param name="sender">sender</param>
33         /// <param name="e">e</param>
34
35         private void ButtonOnClick(object sender, EventArgs e)
36         {
37             lblHelloWorldLabel.Text = "Eric Gathje!!";
38         }
39     }
40 }
41
42
```

Code behind with proper citation and a commented button click event.

Explanation:

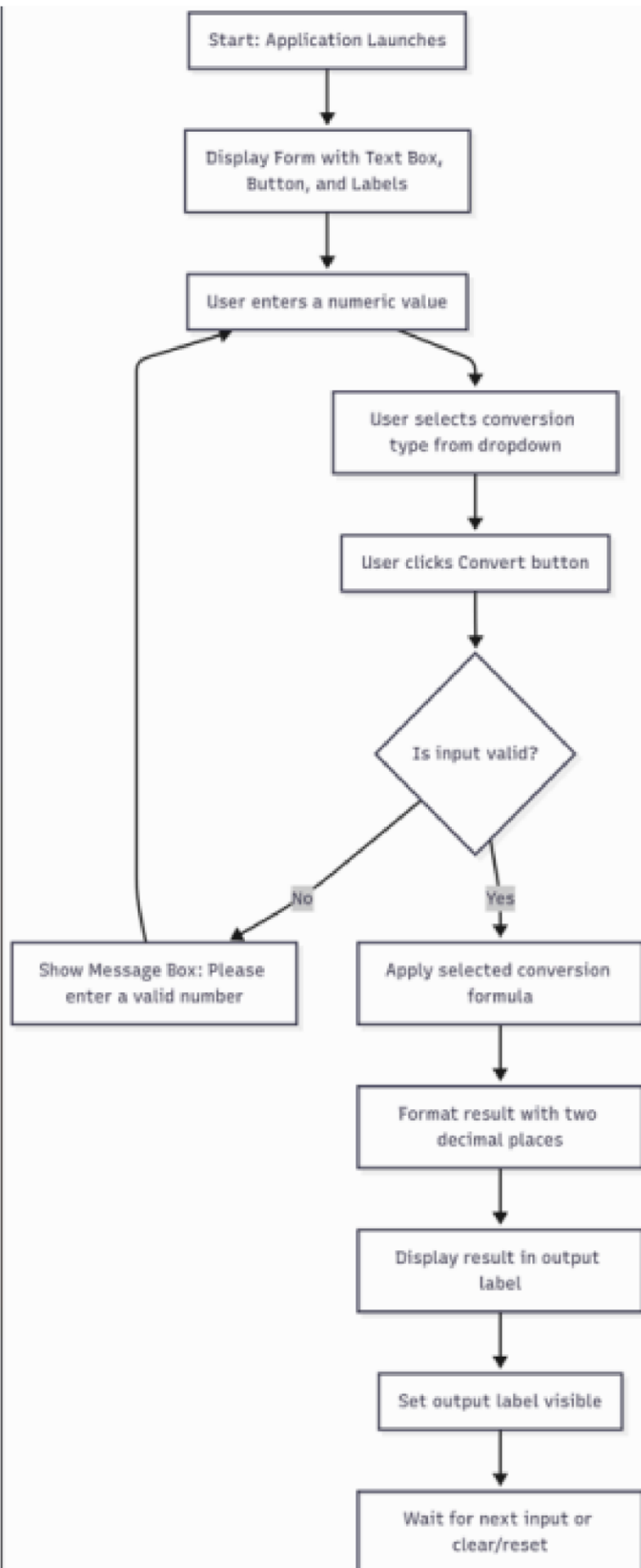
This screenshot presents the code file (Form1.cs) showing the logic executed when the button is clicked.

At the top, the required citation block is added, including my name, course, and a declaration of original work.

The button click method contains summary comments and functional inline comments that describe what each line of code does.

This ensures clarity and proper documentation for anyone reviewing the code. Naming conventions and best practices in C# are followed consistently throughout.

Part 3: Flowchart



Flowchart modeling the weight conversion process from Earth to Mars.

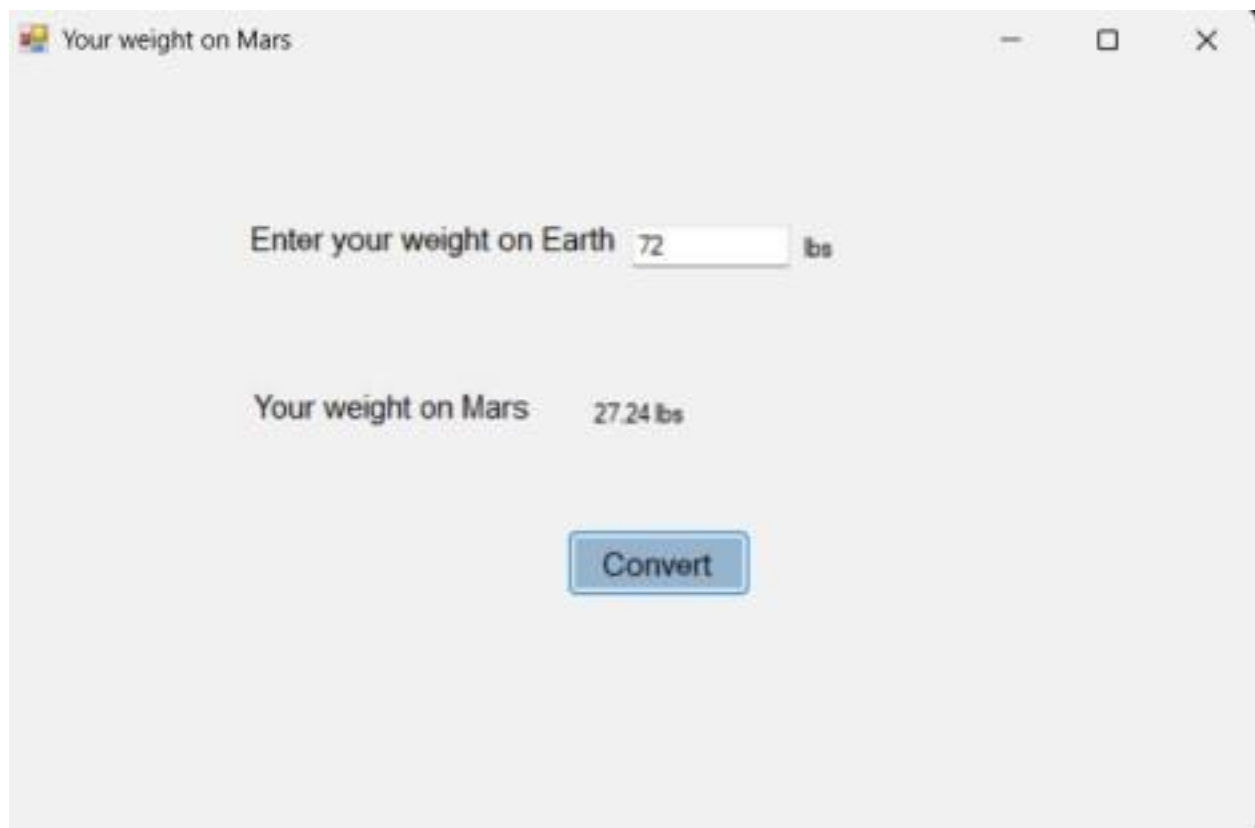
Explanation:

This flowchart outlines the procedural steps of the application.

The program starts by prompting the user to input their weight on Earth.

Once the user clicks the Convert button, the app calculates the mass on Earth using Earth's gravity and then calculates the Mars weight using Mars' gravitational acceleration.

The result is then displayed in a label formatted to two decimal places. This structure helps organize logic before coding.



Windows Forms Application running with converted weight result.

```

+ Name: Ctrl Button1
+ CST_100
+ Activity 1
+ 10/04/2020
+ Citation(s)
Hill, R. (2011, February 17). Microsoft offers Visual Studio 104 extension for .NET appraohel. Infoworld.Lee
Yan, R. (2009). Computer Software Programming based on C Language. 38th Asian Conference on Intelligent Technologies (ACIT), Intelligent Technologies (ACIT)
Yan, J., Chen, Y., & Jiao, A. (2021). Visual Studio Code in Introductory Computer Science Course: An Experience Report.
*/

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace CST_100_Activity_1
{
    /// <summary>
    /// public partial class Form1 : Form
    /// </summary>
    public partial class Form1 : Form
    {
        /// <summary>
        /// InitializeComponent()
        /// //Hide the weight on Mars Labels
        /// lblMars.Visible = false;
        /// lblMarsWeight.Visible = false;
        /// </summary>
        public Form1()
        {
            InitializeComponent();
            //Hide the weight on Mars Labels
            lblMars.Visible = false;
            lblMarsWeight.Visible = false;
        }

        /// <summary>
        /// Handles the click event for the Convert button.
        /// </summary>
        /// <remarks>This method is triggered when the Convert button is clicked. It processes the event
        /// and performs the associated action. Ensures that the sender is the expected control before performing any
        /// operations.</remarks>
        /// <param name="sender">The source of the event, typically the button that was clicked.</param>
        /// <param name="e">An <see cref="EventArgs"/> object containing event data.</param>
        /// </summary>
        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 Earth Weight
            earthWeight = Convert.ToDecimal(txtEarthWeight.Text);

            //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("{0:0.00} lbs", finalValue);

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

```

Code behind the Mars weight converter with citation and detailed comments.

Explanation:

The code handles input from the user, performs gravity-based conversion, and outputs the formatted result.

It declares decimal variables for gravitational constants and performs calculations using correct math expressions.

Inline comments explain each logic step within the `btnConvert_Click` method. This aligns with C# best practices and enhances readability.