



# VISUAL BASIC/ C# PROGRAMMING (330)

REGIONAL – 2019

**Production Portion:**

Program 1: Unit Conversion Utility \_\_\_\_\_ (400 points)

***TOTAL POINTS*** \_\_\_\_\_ (*400 points*)

**Graders: Please double check and verify all scores and answer keys!**

Property of Business Professionals of America.  
May be reproduced only for use in the Business Professionals of America  
*Workplace Skills Assessment Program* competition.



### Unit Conversion Utility

Responsive applications that do not require any sort of buttons to confirm operation are intuitive and easy to use. In this exercise, you will create a Visual Basic or C# Windows Form Application that handles the functionality of unit conversion to handle instant computation as soon as the input value is changed. For simplicity, only volume and weight conversions will be explored.

#### Requirements:

1. You must create a C#/VB Windows Form Application named CS\_330\_ContestantNumber, where ContestantNumber is your BPA assigned contestant number (including dashes). For example, CS\_330\_01\_2345\_6789. If you are using VB, use VB instead of CS in the application name.
2. Your name and contestant number must appear as a comment at the top of the main source code file.
3. Application on Exit button press must ask for confirmation to close, with the Title of the confirmation box being “Confirm” and the text being “Are you sure you wish to exit the Unit Conversion Utility?”
  - a. Exit button must be named “exitButton”
4. The main form must not have minimize, maximize, or close buttons in the Form Handle.
5. The main form must have text “Unit Conversion Utility” for its title.
6. The main form must have a tab control containing two tabs.
  - a. The tab control must be named “unitTabs”, with Dock set to Top.
  - b. The first tab must be named “volumePage” with text “Volume.” It must also have four labels and four textboxes, with label text “Liter:”, “Pint:”, “Quart:”, “Gallon:”, and names “literBox”, “pintBox”, “quartBox”, “gallonBox”.
  - c. The second tab must be named “weightPage” with text “Weight.” It must also have four labels and four textboxes, with label text “Pound:”, “Kilogram:”, “Ounce:”, “Ton:”, and names “poundBox”, “kgBox”, “ounceBox”, “tonBox”.
7. All textboxes must be tied to the same KeyPress event that blocks all non-digit or decimal input. Only one decimal is allowed in a textbox (i.e. 2.00 is valid, 2.00.0 is invalid).
8. On loading the main form, the application must load in the conversion factors from the file “convdata.txt”.
  - a. Volume conversion data is on the first line, ordered by liter, pint, quart, and gallon.
  - b. Weight conversion data is on the second line, ordered by pound, kilogram, ounce, and ton.
  - c. Conversion is scaled by the first unit, thus in the case of volume 1 liter is equivalent to 0.4732 pints.
9. All textboxes must be tied to TextChanged events that parses the current input, then assigns the correct converted value to all other textboxes, displayed to the fourth decimal place.
  - a. Hint: parse the current value, divide it by its conversion factor to normalize it, then for the other three boxes multiply by the corresponding conversion factor.
  - b. Hint: use the Focused attribute to avoid infinite loops.



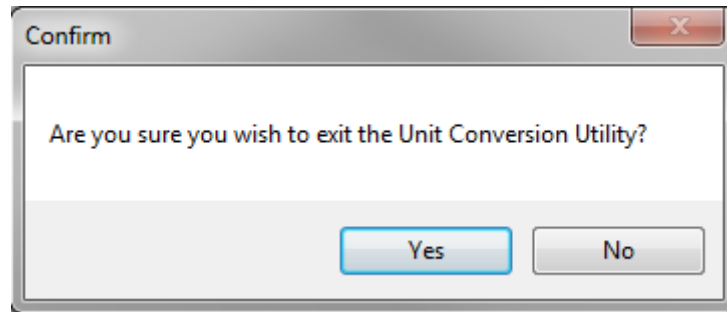
Sample Pictures:

The screenshot shows a window titled "Unit Conversion Utility" with two tabs: "Volume" and "Weight". The "Volume" tab is active. It contains four input fields with labels: "Liter:" (value: 1), "Pint:" (value: 0.4732), "Quart:" (value: 0.9464), and "Gallon:" (value: 3.7854). An "Exit" button is located at the bottom right.

**Figure 1:** The volume tab with an input of 1 in the liter field. Note that the pint, quart, and gallon fields output their conversion factor.

The screenshot shows the same "Unit Conversion Utility" window, but the "Weight" tab is active. It contains four input fields with labels: "Pound:" (value: 16.0000), "Kilogram:" (value: 7.2576), "Ounce:" (value: 1), and "Ton:" (value: 0.0080). An "Exit" button is located at the bottom right.

**Figure 2:** The volume tab with an input of 1 in the ounce field. Note that the pound, kilogram, and ton fields take the input from the ounce field, scaled by the ounce to pound conversion factor, and then scale it back to the corresponding to pound conversion factor.



**Figure 3:** The confirmation window. This should be opened when Exit is pressed. If “Yes” is clicked, the application should close, otherwise return to the main form.

You will have ninety (90) minutes to complete your work.

Your name and/or school name should *not* appear on any work you submit for grading.

Submit a copy your entire solution/project to the flash drive provided. You must submit your entire solution/project so that the graders may open your project to review the source code and/or build and execute your solution/project. **Submissions that do not contain source code will not be graded.**

#### Development Standards

- Standard name prefixes must be utilized for variables.
- All subroutines, functions, and methods must be documented with comments explaining the purpose of the method, the input parameters (if any), and the output (if any).



Your application will be graded on the following criteria:

**Solution and Project**

- The project is present on the flash drive \_\_\_\_\_ 10 pts
- The project is named according to the naming conventions \_\_\_\_\_ 10 pts

**Program Execution**

- Code copied to USB drive and program runs from USB \_\_\_\_\_ 20 pts

*If the program does not execute, then the remaining items in the program executive section receive a score of zero.*

- The program contains an Exit button that asks for user confirmation \_\_\_\_\_ 25 pts
- The exit button is named "exitButton" \_\_\_\_\_ 15 pts
- The main form does not have minimize, maximize, or close buttons  
in the Form Handle \_\_\_\_\_ 25 pts
- The main form has a tab control named "unitTabs" with Dock is set to Top \_\_\_\_\_ 25 pts
- "unitTabs" control has a "volumePage" and a "weightPage" with correct text \_\_\_\_\_ 25 pts
- Each tab page has the correct text boxes and labels \_\_\_\_\_ 25 pts
- All text boxes have proper input validation \_\_\_\_\_ 25 pts
- On form load, conversion data is loaded properly \_\_\_\_\_ 25 pts
- On input change, conversions are performed automatically \_\_\_\_\_ 20 pts
- Conversions are performed correctly \_\_\_\_\_ 30 pts

**Source Code Review**

- Code is commented at the top, for each function, and as needed \_\_\_\_\_ 10 pts
- Code uses reasonable and consistent variable naming conventions \_\_\_\_\_ 10 pts
- Code for reading convdata.txt is present \_\_\_\_\_ 10 pts
- Code for parsing the input file is present \_\_\_\_\_ 10 pts
- Code for volConv array is present \_\_\_\_\_ 20 pts
- Code for wgtConv array is present \_\_\_\_\_ 20 pts
- Code to limit character input to numbers and decimal points is present \_\_\_\_\_ 20 pts
- Code to update text fields is present \_\_\_\_\_ 20 pts

**Total Points: \_\_\_\_\_ / 400 pts**



## SOURCE

Form1.cs

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

namespace unitconverter
{
    public partial class Form1 : Form
    {
        double[] volConv = new double[4];
        double[] wgtConv = new double[4];

        void ReadConversionData()
        {
            FileStream fs = new FileStream("convdata.txt", FileMode.Open, FileAccess.Read);
            using (StreamReader sr = new StreamReader(fs))
            {
                string[] tokens = sr.ReadLine().Split(' ');
                for (int i = 0; i < tokens.Count(); i++)
                    volConv[i] = double.Parse(tokens[i]);
                tokens = sr.ReadLine().Split(' ');
                for (int i = 0; i < tokens.Count(); i++)
                    wgtConv[i] = double.Parse(tokens[i]);
            }
        }

        public Form1()
        {
            InitializeComponent();
            ReadConversionData();
        }

        private void literBox_TextChanged(object sender, EventArgs e)
        {
            if (literBox.Focused)
            {
                double tarVal = double.Parse(literBox.Text) / volConv[0];
                pintBox.Text = (tarVal * volConv[1]).ToString("N4");
                quartBox.Text = (tarVal * volConv[2]).ToString("N4");
                gallonBox.Text = (tarVal * volConv[3]).ToString("N4");
            }
        }

        private void pintBox_TextChanged(object sender, EventArgs e)
        {
            if (pintBox.Focused)
            {
                double tarVal = double.Parse(pintBox.Text) / volConv[1];
                literBox.Text = (tarVal * volConv[0]).ToString("N4");
                quartBox.Text = (tarVal * volConv[2]).ToString("N4");
                gallonBox.Text = (tarVal * volConv[3]).ToString("N4");
            }
        }
    }
}
```



```
    }  
  }  
  
  private void quartBox_TextChanged(object sender, EventArgs e)  
  {  
    if (quartBox.Focused)  
    {  
      double tarVal = double.Parse(quartBox.Text) / volConv[2];  
      pintBox.Text = (tarVal * volConv[1]).ToString("N4");  
      literBox.Text = (tarVal * volConv[0]).ToString("N4");  
      gallonBox.Text = (tarVal * volConv[3]).ToString("N4");  
    }  
  }  
  
  private void gallonBox_TextChanged(object sender, EventArgs e)  
  {  
    if (gallonBox.Focused)  
    {  
      double tarVal = double.Parse(gallonBox.Text) / volConv[3];  
      pintBox.Text = (tarVal * volConv[1]).ToString("N4");  
      quartBox.Text = (tarVal * volConv[2]).ToString("N4");  
      literBox.Text = (tarVal * volConv[0]).ToString("N4");  
    }  
  }  
  
  private void poundBox_TextChanged(object sender, EventArgs e)  
  {  
    if (poundBox.Focused)  
    {  
      double tarVal = double.Parse(poundBox.Text) / wgtConv[0];  
      kgBox.Text = (tarVal * wgtConv[1]).ToString("N4");  
      ounceBox.Text = (tarVal * wgtConv[2]).ToString("N4");  
      tonBox.Text = (tarVal * wgtConv[3]).ToString("N4");  
    }  
  }  
  
  private void kgBox_TextChanged(object sender, EventArgs e)  
  {  
    if (kgBox.Focused)  
    {  
      double tarVal = double.Parse(kgBox.Text) / wgtConv[1];  
      poundBox.Text = (tarVal * wgtConv[0]).ToString("N4");  
      ounceBox.Text = (tarVal * wgtConv[2]).ToString("N4");  
      tonBox.Text = (tarVal * wgtConv[3]).ToString("N4");  
    }  
  }  
  
  private void ounceBox_TextChanged(object sender, EventArgs e)  
  {  
    if (ounceBox.Focused)  
    {  
      double tarVal = double.Parse(ounceBox.Text) / wgtConv[2];  
      kgBox.Text = (tarVal * wgtConv[1]).ToString("N4");  
      poundBox.Text = (tarVal * wgtConv[0]).ToString("N4");  
      tonBox.Text = (tarVal * wgtConv[3]).ToString("N4");  
    }  
  }  
  
  private void tonBox_TextChanged(object sender, EventArgs e)
```



```
{
    if (tonBox.Focused)
    {
        double tarVal = double.Parse(tonBox.Text) / wgtConv[3];
        kgBox.Text = (tarVal * wgtConv[1]).ToString("N4");
        ounceBox.Text = (tarVal * wgtConv[2]).ToString("N4");
        poundBox.Text = (tarVal * wgtConv[0]).ToString("N4");
    }
}

private void checkKeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = (!char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar) &&
(e.KeyChar != '.')) // allow numbers
    || ((e.KeyChar == '.') && ((sender as TextBox).Text.IndexOf('.') > -1)); //
allow 1 decimal point
}

private void exitButton_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Are you sure you wish to exit the Unit Conversion
Utility?", "Confirm",
        MessageBoxButtons.YesNo) == System.Windows.Forms.DialogResult.Yes)
    {
        this.Close();
    }
}
}
```





## Visual Basic Source Code Example

```
Imports System
Imports System.Collections.Generic
Imports System.ComponentModel
Imports System.Data
Imports System.Drawing
Imports System.IO
Imports System.Linq
Imports System.Text
Imports System.Threading.Tasks
Imports System.Windows.Forms
```

```
Namespace unitconverter
```

```
    Public Partial Class Form1
```

```
        Inherits Form
```

```
        Private volConv As Double() = New Double(3) {}
```

```
        Private wgtConv As Double() = New Double(3) {}
```

```
        Private Sub ReadConversionData()
```

```
            Dim fs As FileStream = New FileStream("convdata.txt", FileMode.Open, FileAccess.Read)
```

```
            Using sr As StreamReader = New StreamReader(fs)
```

```
                Dim tokens As String() = sr.ReadLine().Split(" ")
```

```
                For i As Integer = 0 To tokens.Count() - 1
```

```
                    volConv(i) = Double.Parse(tokens(i))
```

```
                Next
```

```
                tokens = sr.ReadLine().Split(" ")
```

```
                For i As Integer = 0 To tokens.Count() - 1
```

```
                    wgtConv(i) = Double.Parse(tokens(i))
```

```
                Next
```

```
            End Using
```

```
        End Sub
```

```
        Public Sub New()
```

```
            InitializeComponent()
```

```
            ReadConversionData()
```

```
        End Sub
```

```
        Private Sub literBox_TextChanged(ByVal sender As Object, ByVal e As EventArgs)
```

```
            If literBox.Focused Then
```

```
                Dim tarVal As Double = Double.Parse(literBox.Text) / volConv(0)
```

```
                pintBox.Text = (tarVal * volConv(1)).ToString("N4")
```

```
                quartBox.Text = (tarVal * volConv(2)).ToString("N4")
```

```
                gallonBox.Text = (tarVal * volConv(3)).ToString("N4")
```

```
            End If
```

```
        End Sub
```



```
Private Sub pintBox_TextChanged(ByVal sender As Object, ByVal e As EventArgs)
    If pintBox.Focused Then
        Dim tarVal As Double = Double.Parse(pintBox.Text) / volConv(1)
        literBox.Text = (tarVal * volConv(0)).ToString("N4")
        quartBox.Text = (tarVal * volConv(2)).ToString("N4")
        gallonBox.Text = (tarVal * volConv(3)).ToString("N4")
    End If
End Sub
```

```
Private Sub quartBox_TextChanged(ByVal sender As Object, ByVal e As EventArgs)
    If quartBox.Focused Then
        Dim tarVal As Double = Double.Parse(quartBox.Text) / volConv(2)
        pintBox.Text = (tarVal * volConv(1)).ToString("N4")
        literBox.Text = (tarVal * volConv(0)).ToString("N4")
        gallonBox.Text = (tarVal * volConv(3)).ToString("N4")
    End If
End Sub
```

```
Private Sub gallonBox_TextChanged(ByVal sender As Object, ByVal e As EventArgs)
    If gallonBox.Focused Then
        Dim tarVal As Double = Double.Parse(gallonBox.Text) / volConv(3)
        pintBox.Text = (tarVal * volConv(1)).ToString("N4")
        quartBox.Text = (tarVal * volConv(2)).ToString("N4")
        literBox.Text = (tarVal * volConv(0)).ToString("N4")
    End If
End Sub
```

```
Private Sub poundBox_TextChanged(ByVal sender As Object, ByVal e As EventArgs)
    If poundBox.Focused Then
        Dim tarVal As Double = Double.Parse(poundBox.Text) / wgtConv(0)
        kgBox.Text = (tarVal * wgtConv(1)).ToString("N4")
        ounceBox.Text = (tarVal * wgtConv(2)).ToString("N4")
        tonBox.Text = (tarVal * wgtConv(3)).ToString("N4")
    End If
End Sub
```

```
Private Sub kgBox_TextChanged(ByVal sender As Object, ByVal e As EventArgs)
    If kgBox.Focused Then
        Dim tarVal As Double = Double.Parse(kgBox.Text) / wgtConv(1)
        poundBox.Text = (tarVal * wgtConv(0)).ToString("N4")
        ounceBox.Text = (tarVal * wgtConv(2)).ToString("N4")
        tonBox.Text = (tarVal * wgtConv(3)).ToString("N4")
    End If
End Sub
```

```
Private Sub ounceBox_TextChanged(ByVal sender As Object, ByVal e As EventArgs)
    If ounceBox.Focused Then
        Dim tarVal As Double = Double.Parse(ounceBox.Text) / wgtConv(2)
        kgBox.Text = (tarVal * wgtConv(1)).ToString("N4")
        poundBox.Text = (tarVal * wgtConv(0)).ToString("N4")
        tonBox.Text = (tarVal * wgtConv(3)).ToString("N4")
    End If
End Sub
```



```
Private Sub tonBox_TextChanged(ByVal sender As Object, ByVal e As EventArgs)
    If tonBox.Focused Then
        Dim tarVal As Double = Double.Parse(tonBox.Text) / wgtConv(3)
        kgBox.Text = (tarVal * wgtConv(1)).ToString("N4")
        ounceBox.Text = (tarVal * wgtConv(2)).ToString("N4")
        poundBox.Text = (tarVal * wgtConv(0)).ToString("N4")
    End If
End Sub

Private Sub checkKeyPress(ByVal sender As Object, ByVal e As KeyPressEventArgs)
    e.Handled = (Not Char.IsDigit(e.KeyChar) AndAlso Not Char.IsControl(e.KeyChar) AndAlso (e.KeyChar <>
"."c)) OrElse ((e.KeyChar = "."c) AndAlso ((TryCast(sender, TextBox)).Text.IndexOf("."c) > -1))
End Sub

Private Sub exitButton_Click(ByVal sender As Object, ByVal e As EventArgs)
    If MessageBox.Show("Are you sure you wish to exit the Unit Conversion Utility?", "Confirm",
MessageBoxButtons.YesNo) = System.Windows.Forms.DialogResult.Yes Then
        Me.Close()
    End If
End Sub
End Class
End Namespace
```