Contestant Number: _____

Time: _____

Rank: _____

VISUAL BASIC/ C# PROGRAMMING
(330)

REGIONAL – 2019

Production Portion:

Program 1: Unit Conversion Utility   _____ (400 points)

TOTAL POINTS   _____ *(400 points)*

Failure to adhere to any of the following rules will result in disqualification:
1. **Contestant must hand in this test booklet and all printouts. Failure to do so will result in disqualification.**
2. **No equipment, supplies, or materials other than those specified for this event are allowed in the testing area. No previous BPA tests and/or sample tests or facsimile (handwritten, photocopied, or keyed) are allowed in the testing area.**
3. **Electronic devices will be monitored according to ACT standards.**

No more than ten (10) minutes orientation
No more than ninety (90) minutes testing time
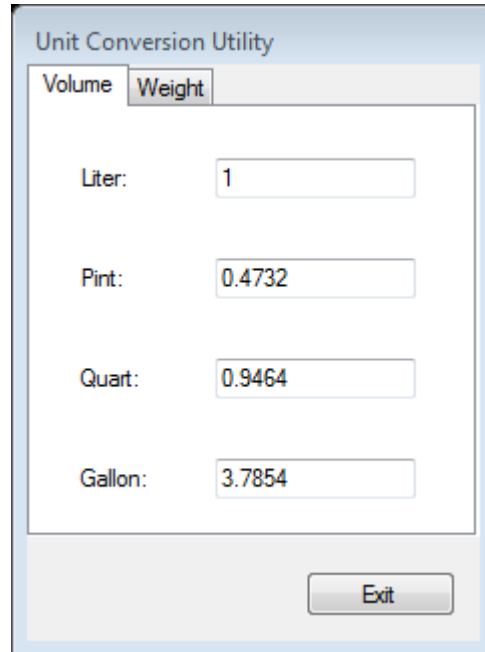No more than ten (10) minutes wrap-up

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.
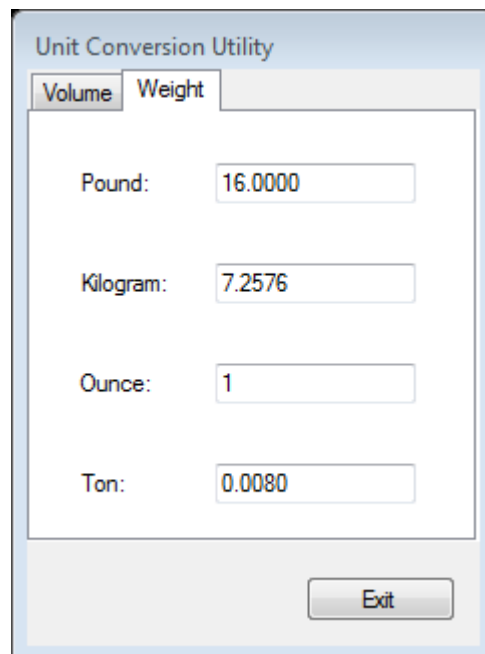
Requirements:

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:

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.

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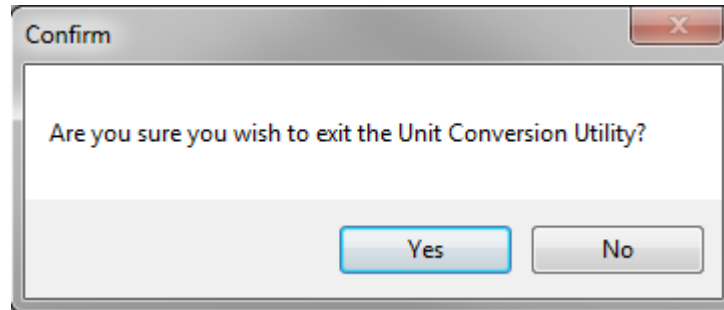
Figure 3: The confirmation window. This should be opened when Exit is pressed. If "Yes" is clicked, the application should close, otherwise return to the main form.

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