

CS 1400 Fundamentals of Programming

Programming Project 2: Wagon Wheel Odometer

Objective:

At the completion of this project, you will have created an application that

- makes use of the algorithm design worksheet to work out your design
- gets input from the user
- makes simple calculations
- formats output and displays it back to the user.
- uses comments to document your conversion algorithm.

The problem

The early pioneers came up with a system to measure how far they traveled each day using a system of cogs and wheels. Knowing the diameter of the wagon wheel, and measuring how many times the wheel turned, they were able to calculate how far they traveled each day. In this project you are to design a Graphical User Interface program that allows the user to enter in a wheel diameter in inches, and then displays the number of turns required for the wheel to travel one mile.

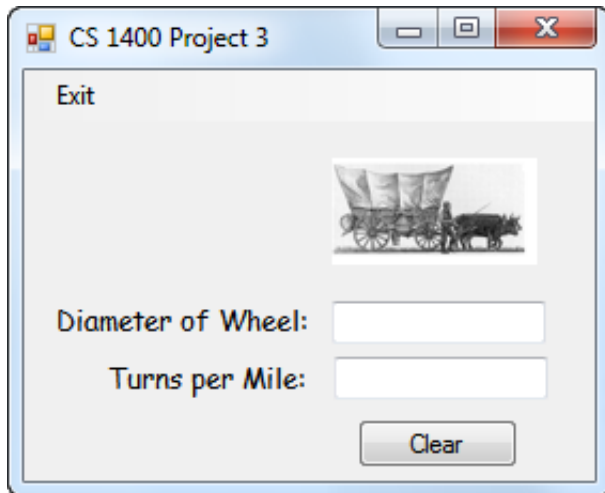


The Interface

You should design your own user interface. It does not need to look like the one shown here, but it should at least provide the following:

- A TextBox, appropriately labeled, where the user enters the diameter of a wagon wheel in inches. This number could contain a decimal part.
- A TextBox, appropriately labeled, where the program will display the

- number of turns that the wheel would take to travel one mile.
- A Button to clear the text boxes and position the cursor in the first TextBox.
- An Exit Menu item and an About menu item.



Writing The Program

Before you start writing the code for this program, you may have to do some research to figure out how to do the math conversions. Once you understand how to do this, write the pseudo-code for the algorithm required to do the conversions.

Once you have your pseudo-code written, start Visual Studio and create a new Windows Forms Application. Then lay out your user interface. I wrote my program so that the code to calculate the number of turns was triggered when the user tabbed out of the first TextBox. See this [hint](#) if you want to try doing it this way. After you complete your code make sure that your program is completely documented. You should have a complete file prologue, every event handler you write should have a complete method prologue, and you should include comments to documenting your algorithm.

Format and document your code in accordance with the course style guidelines. Include a file prologue identifying you as the author. Submit your project using the instructions outlined in the Course Syllabus, Programming Projects section.

File(s) to Submit:

Place your complete project folder into a zip file and name the zip file proj_o2_your-initials_V1.o.zip. For example, I would name my file proj_o2_RKD_V1.o.zip. Submit this assignment as Project #2 on Canvas.

Grading Criteria

	Points	Your
--	--------	------

Description	possible	points
Project meets grading guidelines: o Source code files contain a declaration that you did not copy any code o Project has been submitted to Canvas o Code meets style guidelines o Code is properly documented	5	
Your program contains the pseudo-code that you developed for converting the wheel diameter to rotations per mile.	5	
Your interface contains a working Exit menu item and a working About menu item.	5	
Your interface contains the required GUI components, and your interface is intuitive and easy to use.	5	
Your program correctly converts a wheel diameter to rotations per mile, and formats the output so that it contains 2 digits after the decimal point.	10	
Early bonus (+5 pts) or late penalty (20%/day)		
Total	35	

Sample Executable

You can get an executable that runs correctly [here!](http://debyro.tc.uvu.edu/1400/projects/project02/p02.html)