Inventory Bins

Basic Steps in Creating your Program

1. Plan your logic before working on the project and review the syntax you will use in the project. Think about how you want the lines of output to be displayed to the user. You will want to tell the user what the program does and what type of input you expect from the user (Introduction to the User). Select names of the variables you will use in the program.
2. Create a C++ project in Visual Studio 2012. And have the code window open to write the project code. Display Using Visual Studio to create your program in C++ and follow the instructions on creating the project.
    - Declare variables to hold information about tank capacity and mpg
    - Set up information about this program and what the user will enter and results expected
    - Ask the user to enter some information requested and save the response to a variable. Make sure the position for the user to enter information is on the same line as the request for information.
    - Use line spacing (**endl** or **\n**) to separate lines of output to make it easy for the user to read the information.
3. Documentation: All lines of code must be documented.
    - Use the syntax to designate lines in code as comments (**//**)
    - These comments can be at the end of a line of code or in a group definition at the beginning of a section of code. All methods and function need a group heading document.
4. Run the program using **Ctrl+F5.** Debug the code, making sure all calculations are correct and the line spacing is suitable for readability.

Your program Assignment

Write a program that simulates inventory bins in a warehouse. Each bin holds a number of the same type of parts. The program should use structure that keeps the following data:

1. Description of the part kept in bin
2. Number of parts in the bin

The program should have an array of 10 bins, initialized with the following data:

Part Description	Number of Parts in the Bin
Valve	10
Bearing	5
Bushing	15
Coupling	21
Flange	7
Gear	5
Gear Housing	5
Vacuum Gripper	25
Cable	18
Rod	12

The program should have the following functions:

AddParts: a function that increases a specific bin's part count by specified number.

RemoveParts: a function that decreases a specific bin's part count by a specified number.

When the program runs, it should repeat a loop that performs the following steps:

- The user should see a list of what each bin holds and how many parts are in each bin.
- The user can choose to either quit the program or select a bin.
- When a bin is selected, the user can either add parts to it or remove parts from it.
- The loop then repeats showing the updated bin data on the screen.

Input Validation: No bin can hold more than 30 parts, so don't let the user add more than a bin can hold. Also, don't accept negative values for the number of parts being added or removed.