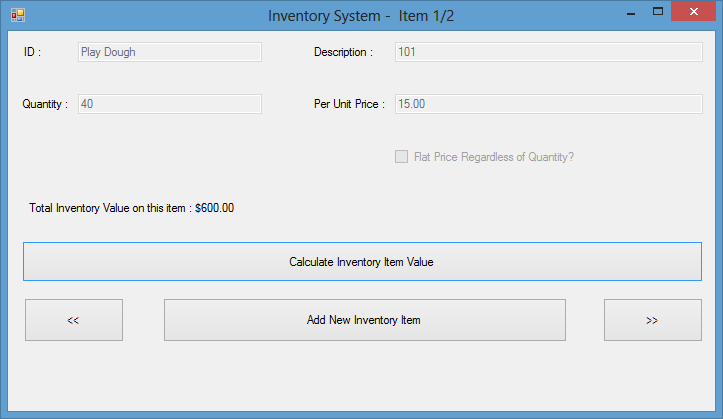
***CIS 311 Assignment 1***

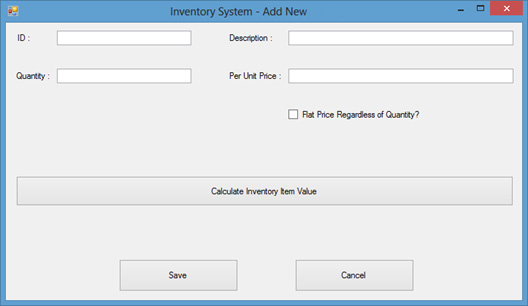
For your first assignment, you will be using pretty much all of the skills that you should have learned in CS 146. I want you to write a Windows Forms based Inventory Management System that allows the information stored in the system to be persisted out to a file between program executions. In addition, the system should allow a user to add new inventory items to it (you do not have to worry about allowing editing or deleting). Once an inventory item is added, it stays added.

Here’s what the initial form will look like with an inventory item being displayed:

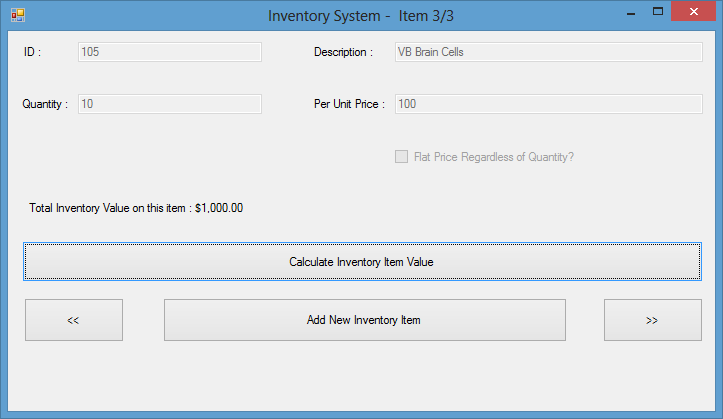


The top four data items should be pretty obvious as to what they are. The fifth item, the checkbox, indicates whether this is a per-unit costed item or a flat price regardless of quantity. These five items encompass what you will need to save out to a file. You will not store the calculated inventory item value – that needs to be done on the fly, whenever and wherever a user decides to press the Calculate button.

When you start up the system for the first time, everything will be empty, so you will want to start by presenting the Add a New Inventory Item form:

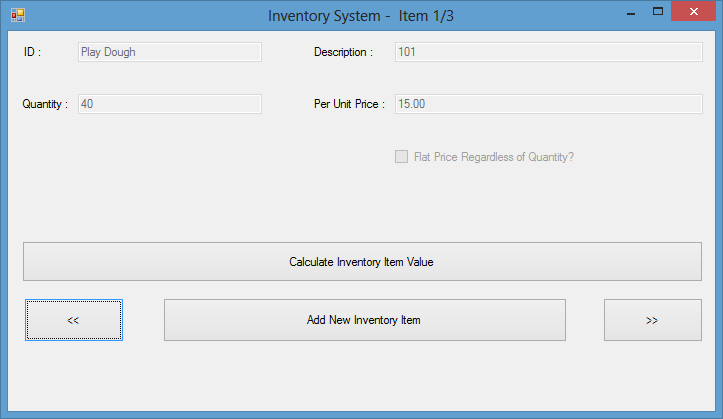


Notice that the title has changed on the form. The textboxes and checkbox are now all available and can be changed. The user may press the Calculate Inventory Item Value button at any time and see that value displayed in the label halfway down the screen (see screenshot on previous page for an example calculation). Once the user has typed in the necessary information, he or she may either Save or Cancel the data record addition. Also understand that this would be the same form that the user would see if he/she pressed the “Add” Button on the main form and that only Calculate Value, Save and Cancel buttons are available on the Add form. If the user decides to add the inventory item, we are taken back to the main screen:

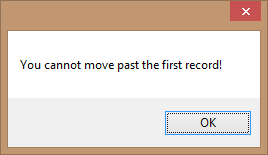


In my program’s execution, this was actually the third inventory item that I added… Notice that the textboxes and checkboxes are not editable other than when adding a new inventory item. At this point, we can navigate through the other records using the << and >> buttons. We can also calculate the current record’s total inventory item value as well by pressing the “Calculate Inventory Item Value” button. The “Add New Inventory Item” button is there whenever we need to add another new inventory item. Finally, we see that the title tells us which record we are viewing (current record number / count of records).

Remember that your system needs to persist the records out to a file (you pick what you want this called – but make sure that it lives in the same directory as your executable file, so there’s no pathing issues). If we examine the original form that we saw, you’ll notice that there are navigation buttons, << and >>, on the bottom of the form:



You need to add MessageBoxes if the user tries to navigate past either end of the records, a MessageBox if the inventory item addition is cancelled, a MessageBox containing an error message if the inventory total cannot be calculated for any reason (characters instead of numbers or missing data). In other words this should be a well behaved application… Here’s one of my error MessageBoxes:



The easiest way to implement this would probably be to use a backing memory store like a one-dimensional array of structures or strings. You could then put each inventory item into an element and simply read and write the array elements in/out to the file.

You will need to zip all of your project files (everything in the project folder and all folders/contents beneath it) and upload that to Canvas. You will also need to print and complete a program cover sheet, print your program’s source code and print screenshots of the program in execution – ***staple (buy a stapler if you don’t have one!!!!)***  those items together in that order and turn that in at the beginning of class on the due date. This will be standard practice for every assignment in this class, so make sure you do things correctly or I will deduct points.