Programming Assignment #2

Price Statistics

Due: 04/03/2019 at 11:59 PM

Introduction

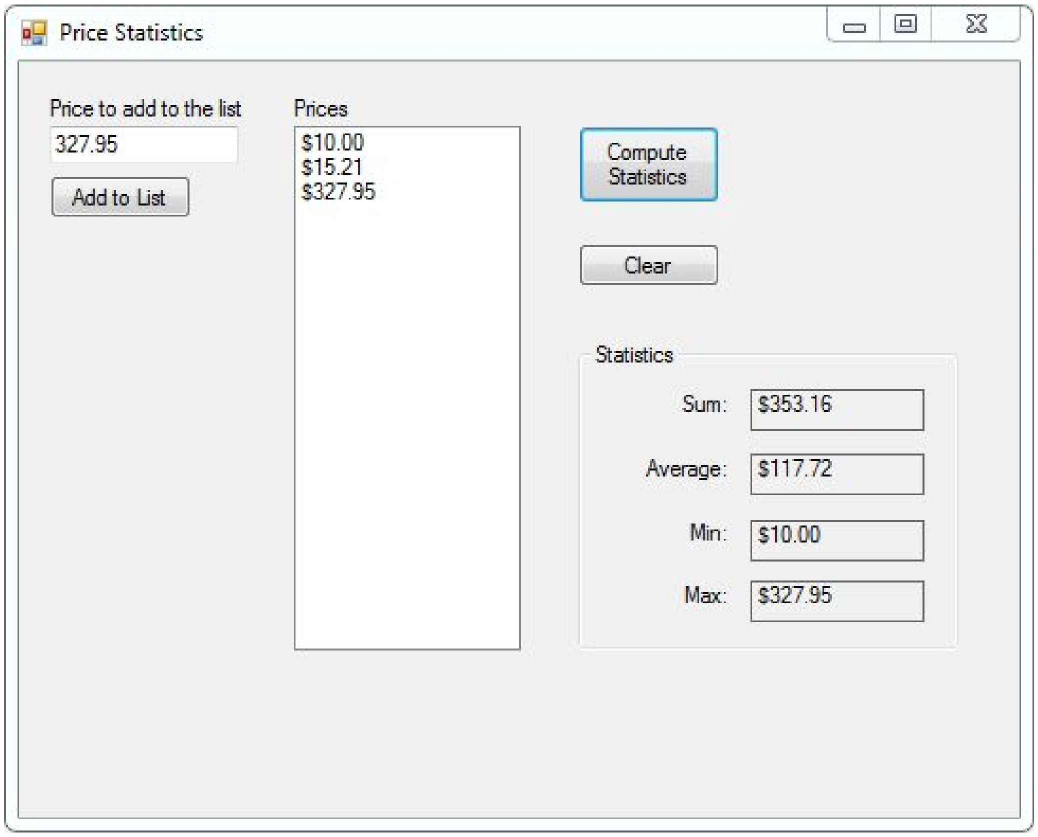
For this assignment, you are going to create a program that allows users to enter prices into a list, and then compute statistics about those prices.

Statistics your code should be able to compute:

- Sum of the prices
- Average of the prices
- Max price
- Min price

User Interface Design (20 pts)

You should try to make your user interface look similar to the following screenshot. You need to have labels and textboxes, as well as a "Calculate" button somewhere in the interface.



327.95

$10.00
$15.21
$327.95

Sum: $353.16
Average: $117.72
Min: $10.00
Max: $327.95

Instructions

The grading of this assignment has been partially automated by programs that will type in inputs into your program, and will record how your program reacts. Failure to follow directions may result in your program "breaking" the grading scripts, and will result in points being taken off.

There are 2 things you'll need to check to make sure the automated stuff will work correctly:

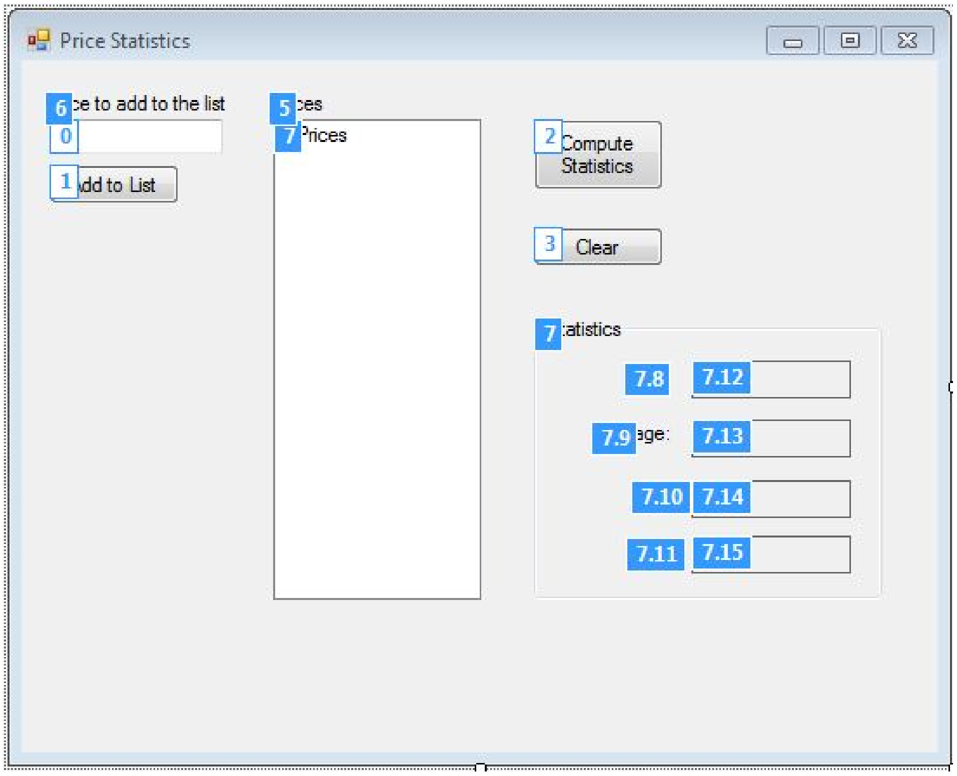
- Window Title
- Tab Order

Window Title

You need to make sure your application's window title is set to "**Price Statistics**".

Tab Order (10 pts)

Tab Order is the order that the focus will jump around in a program when the user hits the "Tab" key. Many programs use tab ordering to provide access to key features for disabled users, and for automation software.



(You can see this view of your application by clicking on a UI element, then going to "View" > "Tab Order". Clicking on a UI element will change its Tab Order in this view.)

Regardless of how these items are arranged visually, hitting the Tab key should cause the cursor to jump between interface elements in the following order:

1. Price Entry textbox
2. “Add to list” button
3. “Compute statistics” button
4. “Clear” button
5. List box of prices

Adding Items to the List

Your program needs to provide a way for users to add prices to the list. Use a **text box** and **button** to do this.

Clearing the List

Your program needs to provide a way for users to quickly clean up a list, and start a new one. Use a **button** for this. Ensure the focus lands back on the price entry textbox using its `.Focus()` method.

Computing the Sum (10 pts)

Your program will compute 4 things about the list of prices. One of these is the sum, or total, of all the prices in the list.

You should use a loop for this, as you don’t know in advance quite how many items a user might put in the list. Display your results with a **label**.

Computing the Average (10 pts)

The average of all the list prices can be computed with the following formula:

$$(\text{price}_1 + \text{price}_2 + \text{price}_3 + \ldots + \text{price}_N) / N$$

Where N *is the number of prices in the list.*

This formula can be implemented with a loop, and a little math afterwards. Display your results with a **label**.

Computing the Max (10 pts)

Finding the max value in a list can be done with a loop and a variable to keep track of the largest value seen so far. Display your results with a **label**.

Computing the Min (10 pts)

Finding the minimum value in a list can be done with a loop and a variable to keep track of the smallest value seen so far. Display your results with a **label**.

Handling Errors (10 pts)

We've seen in class how to handle bad inputs by the user. For this assignment, **you need to handle errors by displaying a MessageBox** to the user, as we've done before.

The message for bad inputs can be pretty generic; the important thing is that you inform the user about what went wrong in a useful and concise way.

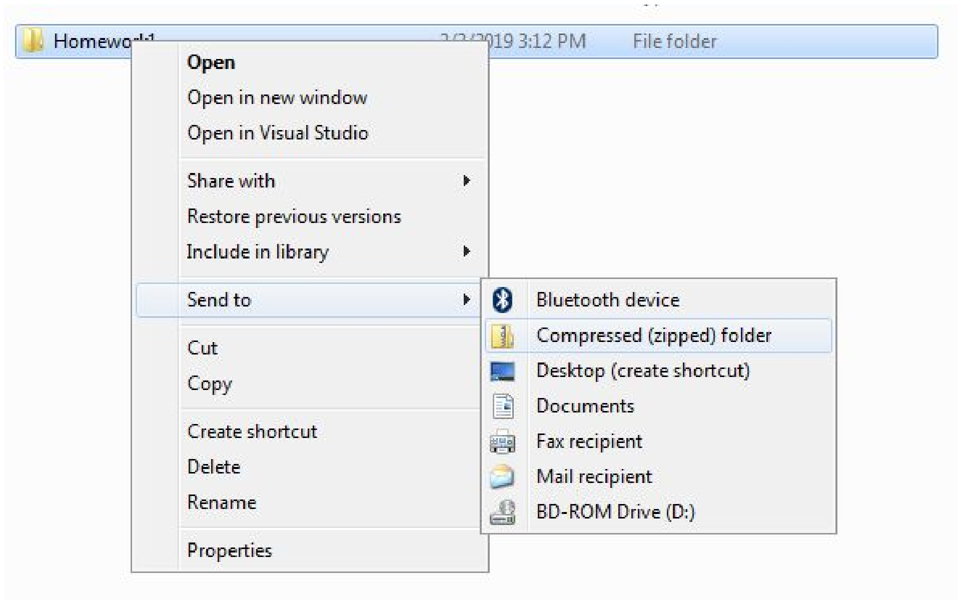
Coding Style (20 pts)

The Instructor/Grader will look at your source code for how you implement your grade calculations. We will be looking for the following things:

- **Your name in a comment** at the top of the file.
- Clear and concise comments, where appropriate.
- Clear/Informative variable names.
- Good layout of the code (sometimes a blank line in the right spot makes the code look better!)

How to turn in your assignment

1. Find the `C:\Users\<YOUR_USERNAME>\source\repos` folder.
2. Right click on the folder where your Visual Basic project lives.
3. Select "Send to" > "Compressed (zipped) folder" to create a ZIP file of your project. The menu should look something like the following picture:



4. Upload the ZIP file to Blackboard.

Grade Rubric

● UI Design/Layout	:: 20 pts
● Tab Order	:: 10 pts
● Code:	
○ Sum	:: 10 pts
○ Average	:: 10 pts
○ Min	:: 10 pts
○ Max	:: 10 pts
● Code Style	:: 20 pts
● Error-handling	:: 10 pts
Total	:: 100 pts