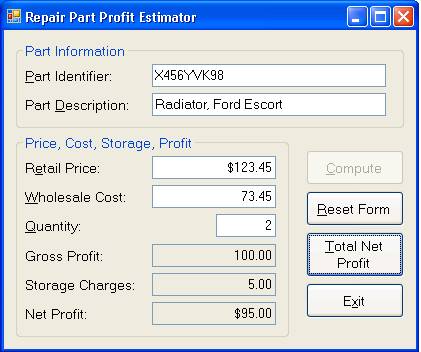
Lab 2b

Management at Bock Repair Parts Sales requires you to develop an application that can compute the profits associated with the sale of repair parts.

Design Requirements. Develop a form that is similar to the one shown below. Your form may have minor differences. The form will enable the display of information about repair parts and services that the company has on sale.



**Form Design.**

* Use GroupBox, Label, TextBox, and Button controls as shown in the figure.
* Left and right-align data as shown in the figure.
* The output controls in the Price, Cost, and Cost Information GroupBox (Gross Profit, Storage Charges, and Net Profit TextBox controls are set to be read-only with the TabStop = False).
* The tab order must be top to bottom, left to right within the GroupBox controls.
* The Compute, Total Net Profit, and Exit Button controls are enabled on startup. The Reset Form Button control is disabled on startup.
* The Label and Button controls must have hot keys as indicated in the figure.
* Map the Compute Button control to the keyboard Enter key; map the Reset Form Button control to the keyboard Esc key.
* Add a ToolTip control to the form. Develop your own tool tip advice to be displayed for the Retail Price, Wholesale Cost and Quantity TextBox controls.

**Coding Requirements.**

Add remarks to the program as required in earlier labs.

You must set OPTION STRICT ON.

Compute Button Coding. Data is entered into the five input TextBox controls within the GroupBox controls, and then the Compute Button is clicked.

Use a Try-Catch block to catch any exceptions that may occur including those associated with data entry. If there is a data entry error in the Retail Price, Wholesale Cost, or Quantity TextBox values, display a message box like the one shown below.



* Gross Profit is computed by multiplying the Quantity times the difference between the Price less the Cost of a part.
* Storage Charges are charged at 5% of the Gross Profit – declare the 5% value as a constant of data type Single.
* The Net Profit is the Gross Profit less the Storage Charges.
* Use the Parse, Convert, and Round methods as appropriate within the program.
* Format output display to TextBox controls as shown in the first figure.
* Accumulate the Total Net Profit for each part to a module-level accumulating variable. Accumulate the Total Quantity to a module-level accumulating variable. You can assume that each time the Compute button is clicked, the profit for a part is estimated.
* Disable the Compute button and enable the Reset Form button.
* Reset Form Button Coding.
* Clear all data input and output controls.
* Enable the Compute button and disable the Reset Form Button.
* Set focus to the Part Identifier TextBox control.

**Total Net Profit Button Coding.**

* Use the module-level variables accumulated when clicking the Compute Button to compute the average net profit, and display a message box like the one shown below that gives the total net profit, total quantity of parts, and average net profit for all profit estimates.
* Use a Try-Catch block with multiple Catch blocks to catch the exception that can occur if there have not been any parts estimates made yet, and to catch any general exceptions that may occur. The message box for a no parts estimates made yet exception is shown below. You should also design a message box to display in the event that a general exception occurs when clicking this button.

Exit Button Coding. Clicking this control should close the form.

Test the Lab. Test your lab to ensure that it works correctly.