Lab3.htm; January 12, 2012

**The Edwardsville Auto Store - Lab 3**

For this assignment, the auto store needs an application that will compute the value to be credited to a customer

**Getting Started.**

* Start Visual Studio and create a project as you have for previous projects. Name the project **Lab3-SectionTime-YourLastName-YourFirstName** as you did for previous projects. The form’s **File** **Name** property should be **Lab3.vb**.
* Save the project as you have for previous projects.

**Design Requirements.** Develop a form that is similar to the one shown below. Your form may have minor differences. Customers sometimes return items that have been purchased. The customers are refunded their purchase price less 5% for a restocking fee. The form enables the computation of the net $$$ value to be returned to the customer.



**Form Design.**

* Set the form’s Title Bar as shown but place your name in the Title Bar (not Doug Bock).
* 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 Purchase Price, Quantity, Net Returned GroupBox (Total Value, Restocking Fee, and New to Customer 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 – you pick the hot keys to be assigned.
* 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 Purchase Price and Quantity Returned TextBox controls.
* Map the form’s Compute Button control to the keyboard Enter key. Map the Reset Button control to the keyboard Esc key.

**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 input TextBox controls within the GroupBox controls, and then the Compute Button is clicked.
  + The Compute Button starts up enabled.
  + 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 Purchase Price or Quantity Returned TextBox values, display a message box like the one shown below.



* + The Total Value is computed by multiplying the Purchase Price times the Quantity Returned.
  + The Restocking Fee is charged at 5% of the Total Value – declare the 5% value as a constant of data type **Single**.
  + The Net to Customer is the Total Value less the Restocking Fee.
  + 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 to Customer to a module-level accumulating variable. Accumulate the Total Quantity Returned to another module-level accumulating variable. You can assume that each time the Compute button is clicked, these totals need to be accumulated.
  + Disable the Compute button and enable the Reset Form button.
* **Reset Form Button Coding.**
  + The Button starts up disabled.
  + Clear all data input and output controls.
  + Enable the Compute button and disable the Reset Form Button.
  + Set focus to the Item Identifier TextBox control.
* **Total Net Returns Button Coding.**
  + Use the module-level variables accumulated when clicking the Compute Button to compute the average net return, and display a message box like the one shown below that gives the total net returned to customers, total quantity of items returned, and average value of the net returns for all customer returns.
  + Use a Try-Catch block with multiple Catch blocks to catch the exception that can occur if there have not been any items returned yet, and to catch any general exceptions that may occur. The message box for a *no items returned 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. Use the assessment grading form shown below—it is the same form that will be used to grade your project.

**CAUTION: DO NOT run your project from a flash drive – this can cause the project to become corrupted and you will have to recreate the project – always copy the project to the drive C:\TEMP folder or to My Documents and test run your project.**

Use the test data given here for initial program testing. You may need to extend the testing with additional data sets.

* Ensure you test each Try-Catch block by entering erroneous data in addition to the test data shown below.

| ***Test Data***  **Item ID** | **Description** | **Purchase**  **Price** | | **Quantity** | | **Total Value** | **Restocking Fee** | **Net to Customer** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 121 | Muffler, 2004 Jaguar | $350.00 | | 2 | | 700.00 | 35.00 | $665.00 |
| 333 | Wiper Blades | $29.95 | | 4 | | 119.80 | 5.99 | $113.81 |
|  |  | **Data displayed when the Total Net Returns button is clicked after two items are turned.** | | | | | | |
| Total Net Returns: $778.81 | | | Total items: 6 | | Average Net Return: $129.80 | | | |
|  |  |  |  |  |  |  |  |  |

**What to Do When You’re Finished, How to Save and Submit the Lab.**

Ok, you’ve finished the lab. **Do NOT use the File menu, Save As option**. Instead, follow these steps.

* First, CLOSE Visual Studio—you cannot copy the project to another location if it is open.
* Locate the folder that contains the project. It should be located in the **My Documents** folder where you first saved the project.
* Copy the entire folder to a flash drive.
* Bring the flash drive to the University to either a computer lab or to the computer classroom. Copy the entire folder to drive Y: to the submission folder for your class. You can copy the folder to drive Y: at the beginning of the class period when the project is due.

**Visual Basic Project Assessment Plan – Lab 3 (25 points possible).**

**Before Startup.**

* Submitted late – see the course syllabus for a description of the late penalty.
* Submitted on time but some of the files necessary to run the project are missing – you must resubmit the project, see your instructor if you need assistance submitting the project – your resubmission will be considered a late submission.
* Project should be named correctly: **Lab2-SectionTime-YourLastName-YourFirstName**.
* Form’s File Name property is correct (Lab2.vb, NOT Form1.vb).
* Form’s Title bar displays correct information including your name.
* Form has a good appearance: controls aligned, control size appropriate, no misspelled words, not too much gray space around controls.
* Program code has the required remarks statements to identify the program, programmer name, and date programmed.
* Each sub procedure has remarks statements to identify what the sub procedure does.
* All form controls are named properly following the naming convention taught in the notes and in class.
* Delete empty sub procedures that have no executable code that were created accidentally.
* The form has only a single ToolTip control (only one ToolTip control is needed for each form).
* Each GroupBox control displays the appropriate Text value.
* Erroneously declared variables that were not used or declared too many variables as module-level where most variables should be local.
* Inspect the Compute Button Click Event.
* The restocking fee is declared as a **single** constant (use a Const statement).
* The Decimal.Round, Parse, and Convert methods were used correctly within the program (-1 for each error; maximum deduction -3).
* Accumulated values for total net to customer and total quantity of items returned with coding statements.
* Inspect the Totals Button Click Event – program code has Try-Catch block with multiple Catch blocks to catch zero items returned and to catch general exceptions (major error -2).
* Each GroupBox control displays the appropriate Text value.
* The form's Display Button control is mapped to the keyboard Enter key; the form's Reset Button control is mapped to the keyboard Esc key.
* Option Strict is On.

**Startup.**

* Project will not compile and run – if a good effort was made, stop grading now and assigned 10 out of 25 points – a program that will not run is not a passing grade. If a poor effort was made, stop grading now and assign 5 out of 25 points simply to indicate that the project was submitted. If the project is submitted with the form constructed, but there is no programming code assign a grade of zero points – a project with no programming code is not worth any points.
* Form starts up centered on the screen. Form is an appropriate size.
* The tab order for the form is correct
* Reset Form Button is disabled on startup (all other buttons are enabled).
* Press the Alt key – confirm that the form has hot keys as selected by the programmer.
* Tool Tips will display for the Purchase Price and Quantity Returned TextBox controls as described in the lab assignment.
* Click Enter key – since no data has been entered, if the Compute Button is mapped to the Enter key an error message should display that there is an error in the purchase price or quantity.
* Click the Totals Button – since no items have been returned yet, the “No items have been returned yet” message should display.

**Compute Button Click Event.**

* The data displayed in the Purchase Price, Quantity Returned, Total Value, Restocking Fee, and Net to Customer TextBoxes are right-aligned; the bottom three TextBox controls set to read-only with TabStop = False.
* Output is formatted with currency symbol for at least the Net to Customer TextBox control.
* Total Value is correct for all test cases (-2).
* Restocking Fee is correct for all test cases (-2).
* Net to Customer is correct for all test cases (-2).
* Disables the Compute Button and enables the Reset Form Button.
* Displays appropriate message box when Purchase Price and/or Quantity Returned TextBox values are invalid.

**Reset Form Button Click Event.**

* Clears all data input and output controls.
* Enables the Compute button and disables the Reset Form Button.
* Sets focus to the Item Identifier TextBox control.

**Total Net Returns Button Click Event.**

* Declared module-level variables to store average net value of an item returned.
* Displays the correct running total for net to customer, total number of items returned, and average net value of items returned (-3; -1 for each of the three output values that is incorrect).

**Exit Button Click Event.** Exits the program correctly.

**Other Errors.**

* During program execution, your computer program should compile and execute without generating any error messages – if the program will not compile or generates error messages during execution, you will lose up to 20 points depending on the severity of the error.

**End of Lab**