Lab 4: Car Inventory

Lab Assignment for NETD2202 – Net Development I

Prior to attempting this assignment you should…

1. Read Chapters 8 (Collections), 11 (Class & Objects)
2. Attend Class
3. Watch Classes and Objects Videos
4. Watch Collections Videos
5. Sample Code…
    - VB – Collections Demonstration
    - VB – Student Demo (Classes & Objects)
    - VB – Customer List (Classes and Objects)

IMPORTANT any submission submitted as anything other than a .zip file will be penalized 15%.

This lab is to be completed individually.

Analyze the problem, code and test a solution. Submit your solution, desk-check, and the GitHub URLs to the Lab 3. **IMPORTANT** do not create a repository for Lab 4, the repository for Lab 4 should be created during Lab 5 and the URL for Lab 4 will also be submitted with the Lab 5 submission.

Important: The data in this Lab will only exist for as long as the application is running we will not be storing the data permanently. No data storage required.

Demonstrates

- Visual Studio (Ch. 1)
- Constants and Variables (Ch. 4)
- Use of controls (Ch. 2)
    - Setting Properties Statically
    - Setting Properties Dynamically
- Adherence to the Style Guide
- Events (Ch. 6)
- If/ElseIf (Ch. 5)
- Methods (Functions and Subroutines) (Ch. 6)
- Loops (Ch. 5)
- Class (Ch. 11)
    - Creation
        - Constructor(s)
        - Private Fields/Variables (Instance and Shared/Static)
        - Properties (Get, Set, ReadOnly)
        - Method
- SortedList (Ch. 8)
    - Add (set key value)

- Item (get value with)
- Loop SortedList

Client Requirements

You are being asked to create a car inventory application for a new client. The good news is the tech lead had created a similar application for another client, so you can use the old application as the basis for the new application (CustomerList). The old application maintains a list of customers, while the new application will maintain the current inventory for a car lot.

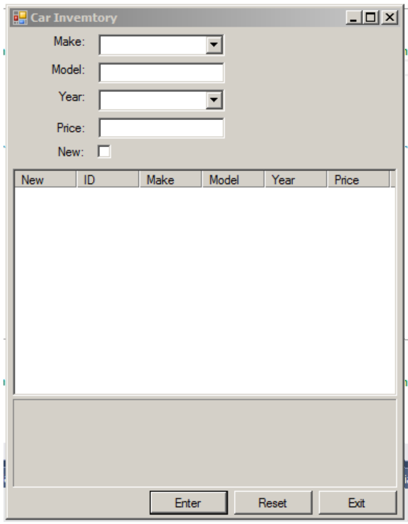
The car lot application will need to maintain the Make, Model, Year, Price and if the car is New or Used. The Make, is the manufacturer of the car, a list will be provided below. The user will be able select the Make from a combo box. The model, or that name, of the car will be entered in a text box. The Year, is the year the car was manufactured and the user will be able to select the year from a combo box, the years that the car lot allows will be provided in a list below. The Price is what the car is selling for and will be entered into a text box by the user. Finally, the status of the car is whether the car is new or used. The user will be able to check a check box to signify if a car is new if it is unchecked it is considered used..

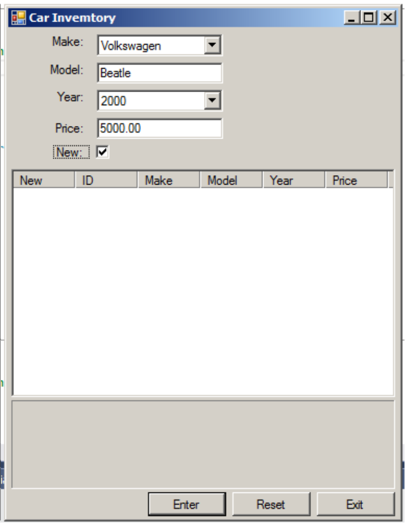
As well, the solution will reset the input controls when the user strikes the enter key, once the data is determined to be valid and entered into a list view control.

The solution should also allow the user to exit the program regardless of the state of the form, in other words. The user can exit at any time.

As part of the company audit process you will be expected to submit your code, a desk-check for the quality assurance department and a link to your GitHub account, specifically, to the new code.

Instructions from the UX (User Experience) Department





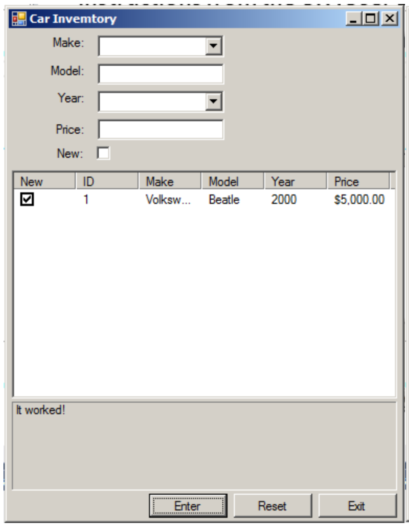
Make: Volkswagen

Model: Beatle

Year: 2000

Price: 5000.00

New: ☑



Make:

Model:

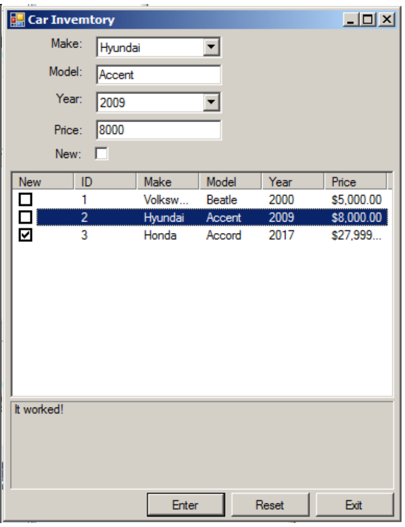
Year:

Price:

New: □

New	ID	Make	Model	Year	Price
☑	1	Volksw...	Beatle	2000	$5,000.00

It worked!



Make: Hyundai

Model: Accent

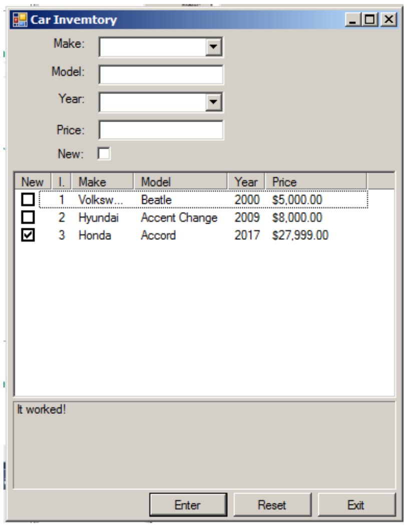
Year: 2009

Price: 8000

New: □

New	ID	Make	Model	Year	Price
□	1	Volksw...	Beatle	2000	$5,000.00
□	2	Hyundai	Accent	2009	$8,000.00
☑	3	Honda	Accord	2017	$27,999...

It worked!



Make:

Model:

Year:

Price:

New: □

New	I..	Make	Model	Year	Price
□	1	Volksw...	Beatle	2000	$5,000.00
□	2	Hyundai	Accent Change	2009	$8,000.00
☑	3	Honda	Accord	2017	$27,999.00

It worked!

1. The form should…
   - Not allow maximizing or minimizing of the screen.
   - Have its title bar set to something meaningful.
   - Ensure its Accept and Cancel buttons are set.
   - Open in the centre of the screen.
   - Be named as per our development style guide.

2. All buttons should…
   - Have their tab indexes set.
   - Have its text set appropriately using access keys (&) to ensure efficiency.
   - Have a tool tip to ensure the user understands the purpose of the key.
   - Be named as per our development style guide.

3. Result/Output Label should…
   - Have its tab index set.
   - Not be allowed to auto size

- Have its text set to an empty string.
- Have its text in the text box should be aligned TopLeft.
- Have a tool tip to ensure the user understand the purpose of the control.
- Have a 3D border style.
- Be named as per our development style guide.

4. Labels that are used strictly for the purpose describing the use of another control like Label to TextBox, and appear to the left of the control they are describing should…
   - Have their tab indexes set.
   - Not be allowed to auto size.
   - The text should be aligned MiddleRight.
   - Have its text set appropriately using access keys (&) to ensure efficiency.
   - Be named as per our development style guide.

5. Input text boxes should…
   - Have their tab indexes set.
   - Text should be empty.
   - Have a tool tip to ensure the user understands the purpose of the control.
   - Be named as per our development style guide.

6. ToolTip control should be named as per our development style guide

7. Combo Boxes should…
   - Have their tab indexes set.
   - Have the drop down style set to DropDownList
   - Have the items collection with appropriate data for the specific list.
   - Text should be empty.
   - Have a tool tip to ensure the user understands the purpose of the control.
   - Be named as per our development style guide.

8. Check Boxes should…
   - Have their tab indexes set.
   - Have the check align set to MiddleRight.
   - Checked default to False.
   - Have text align set to MiddleRight
   - Have its text set appropriately using access keys (&) to ensure efficiency.
   - Have a tool tip to ensure the user understands the purpose of the control.
   - Be named as per our development style guide.

9. ListView should…
   - Have its tab index set.
   - CheckBoxes should be set to True.
   - Have 6 columns

   a. each uniquely named (colNew, colID, colMake, etc.).
   b. each column should have their Text set, this represents the column header (New, ID, Make, etc.).
- Have full row select set to True.
- Have header style set to Nonclickable
- Have multi-select set to False.
- Have view set to Details. (This shows the columns side by side)
- Have a tool tip to ensure the user understands the purpose of the control.
- Be named as per our development style guide.

User Input Flow

1. The user should be allowed to exit the solution by either…
   - Clicking the Exit Button with the mouse on the screen.
   - Clicking the Control Box with the mouse on the screen.
   - Tabbing to the Exit Button and Tapping Enter on the keyboard.

2. The user should be allowed to reset the screen by…
   - Clicking the Reset Button with the mouse on the screen.
   - Tapping the Esc Key on the keyboard,
   - Tabbing to the Reset Button and Tapping Enter on the keyboard.

3. The user should be allowed to enter by…
   - Clicking the Enter Button with the mouse on the screen.
   - Tapping the Enter Key on the keyboard,
   - Tabbing to the Enter Button and Tapping Enter on the keyboard.

4. The user will, (Data Entry)…
   - Select a Make in a combo box representing a car manufacturer.
   - Enter a Model in a text box representing a car name.
   - Select a Year in a combo box representing the year the car was built.
   - Enter a Price in a text box representing the sale price of the car.
     - When display Price in the list view control, ensure that is formatted to 2 decimal places as a currency which a dollar-sign ($).
   - Check if the car is new in a check box, unchecked means the car is used.

5. Validation
   - The client does not want message boxes to be used. So, validation will need to display a list validation messages in the Result label.

6. New Data Entry
   - Once the car data has been entered, validated and the user Enters…
     - The Input controls should be cleared.

    - The new car data should be added to the list view control as a new row with any existing entries that the user might have entered previously.

7. Select and Edit a previously added entry
   - Selecting an existing car from the list view will
     - Populate the Input controls with the data selected
   - The user will now be able to modify the data in the input controls.
   - Once the car data has been modified, validated and the user Enters…
     - The Input controls should be cleared.
     - The modified car data should be update in the list view control.

8. Clicking Reset will …
   - Clear the Input text boxes.
   - Set the combo box list index to -1.
   - Set check boxes checked property to false.

Clicking Exit will …

- Close the Application.

Instructions from the Business Analyst

1. The client requires that the user must be able to
   - Select a car manufacturer (Make) from a list.
   - Enter a car name (Model).
   - Select a car year (Year) from a list.
   - Enter a car sale price (Price).
   - Set whether a car is new or used (New).

2. For a Make to be valid it must be…
   - Selected from the list, this is required. A Make must be selected

3. For a Model to be valid it must…
   - Model is required.
   - Not be empty. Since car names can be just about anything, something like X1 or 25 is valid.
   - It cannot be all blank spaces and ensure blank spaces are removed from the front and back of the entry (Trim).

4. For a Year to be valid it must be…
   - Selected from the list, this is required. A Year must be selected

5. For a Price to be valid it must be…
   - Price is required
   - A real number.
   - It cannot be less than zero.

6. For New to be valid…
    - There are no requirements for New or Used.

Instructions from the Tech Lead

1. All classes and the form must be set to `Option Strict On`
2. Reset…
    - Should have its own Method (subroutine) as it will not only be called from the btnReset click event, but it will also be called when an Entry has been validated and entered into the list view.
    - Please ensure that the reset clears/resets form level variables form level variables as well as Input controls.

3. Class. This project may expand and reusability may become an issue, so a Car class will need to be developed for future considerations…
    - Properties
        - Count – ReadOnly - (This property should refer to a Private Shared/Static variable).
        - IdentificationNumber – ReadOnly
        - Make
        - Model
        - Year
        - Price
        - NewStatus
    - Constructors (2)
        - Default Constructor(no parameters) should…
            - Increment the car count
            - Set the car identification number based on the new car count
        - Parameterized Constructor should…
            - Call the Default constructor.
            - Have parameters representing the make, model, year, price, and new status of the car. It should set the private variable to the values passed by the parameters.
    - Method(s)
        - GetCarData (Function)
            - Returns a string representing the car objects data

4. Enter should gather the user input, validate the input, process the input (if any), and display the input…
    - Validation should have its own Method (function) that will…
        - Return a Boolean
            - False, if any of the user input is invalid
            - True, if all of the user input is valid
        - Set a specific message for each invalid user input
        - Display all messages to the user in the Result label
        - Prevent processing if the return value of the validation method returns False.
    - Processing should…
        - Check to see if the car is a new entry or if the car is an existing car selected from the list.
        - If it is a new entry…
            - Instantiate a new car object using the parameterized constructor for Make, Model, etc.
            - Add the newly instantiate car object to the cars collection object, and ensure when adding to set the key to the car's identification object, so it will be easier to find when editing. (The cars collection should be declared with a form level scope.)
        - If it is an existing entry selected from the list…
            - Get the car object from the cars collection using the key set during the user selection.
            - Assign/Set the Make, Model, Price, NewStatus properties of the car object to the corresponding controls.
        - Repopulate the list view control
            - The list view items should be cleared.
            - The cars collection should be used to repopulate the list view control. (For or For Each Loop can be used for the repopulation, you choice)
        - Reset the Input controls once the data has been entered in to the list view.

5. List View consideration
    - Prevent the list view check box from allowing the user to click it on "On" or "Off", please see the previously created application to see how to prevent this.

6. Selecting an Item in List View should
    - Set a form level variable to the currently selected car identification number in the list view control. The reason this is form level is it will be used later when the user modifies and enters to update the car object data and list view control.
    - Get the car object from the cars collection using the identification number obtained from the list view control.
    - Set the Input entry controls based on the corresponding values in the car object.

7. Questions: Tech Lead is Alfred at alfred.massardo@durhamcollege.ca.
8. CustomerList will help guide you through most of the code.