

## The Edwardsville Auto Store - Lab 11

For this assignment, the auto store needs an application that will enable a store worker to select repair parts from a ComboBox control. Each repair part selected is added to a listing of parts for a shipment--these parts are listed in a ListBox control. You will demonstrate your skill with list type controls including new controls and methods learned in your recent studies.

### Getting Started.

- Start Visual Studio and create a project as you have for previous projects. Name the project **Lab11-SectionTime-YourLastName-YourFirstName** as you did for previous projects. The form's **File Name** property should be **Lab7.vb**.
- Save the project as you have for previous projects.

**Design Requirements.** Develop a form that is similar to the one shown below. You may have minor differences.

The screenshot shows a Windows application window titled "The Edwardsville Auto Store - Your Name Here". The window has a menu bar with "File", "Maintain Parts", and "Help". Below the menu is a section titled "Part and Order Information". It contains a "Description:" label followed by a ComboBox control showing "Wheel Rim". To the right of the ComboBox is a label "ComboBox". Below the Description is a "Sales Price:" label followed by a TextBox containing "\$225.50". To the right of the Sales Price is a "Quantity to be Shipped:" label followed by a TextBox containing "4". To the right of the Quantity TextBox is a label "TextBoxes (f". Below these fields is a ListBox containing two items: "Brake Pads - Price: \$29.95 - Qty: 2 - Line Total: \$59.90" and "Injector - Price: \$75.15 - Qty: 4 - Line Total: \$300.60". To the right of the ListBox is a label "ListBox". At the bottom right of the window is an "Order Total:" label followed by a TextBox containing "\$360.50". To the right of the Order Total TextBox is a label "TextBox (rea for output)".

### Build and Code the Project

#### 1. Design (5 points).

- Use appropriate names for the controls on the form.
- The Description ComboBox control – set DropDownStyle = DropDown. Add the following repair parts to the Items (collection) property at design time:
  1. Brake Pads
  2. Engine
  3. Injector
  4. Oil Filter

5. Tire
  6. Transmission
  7. Wheel Rim
- Sales Price and Quantity to be Shipped TextBox controls – set ReadOnly = False. These two controls are used for data input.
  - The ListBox Items (collection) property is empty on startup.
  - Order Total TextBox control – set ReadOnly = True and TabStop=False. This control will only display output.
  - All TextBox controls – set TextAlign = Right.
  - Initial focus should be on the Description ComboBox control. Tab order must be correct.
  - Organize the code by regions. You decide an appropriate organization of regions for the program.
  - Add comments throughout the program as appropriate sufficient to describe the processing. Also add general comments at the top of the program to identify the project, your name, and date.
  - You must set **OPTION STRICT ON**.
  - The program must startup with the form centered on the computer screen.
  - The form must be an appropriate size and controls must be large enough to display required data.

**2. Menu (2 points).** Add a menu to the form by using a MenuStrip control with the following menu items. Add the specified shortcut keys and separators. The function of each menu item is explained throughout the remaining requirements.

&File	&Maintain Parts	&Help
E&xit	Add Part to Co&mboBox      Ctrl+M &Delete Part from ComboBox      Ctrl+D ----- &Add Part to ListBox      Ctrl+A &Clear Part ListBox      Ctrl+C C&ount Parts in ListBox      Ctrl+O	&About

### Programming Events

**3. File – Exit menu item (1 point).** When clicked, ask the application user to confirm to exit prior to exiting the application.

**4. Parts – Add Part to ListBox menu item (12 points).** Application users will: (1) select an existing auto part from the Description ComboBox control, (2) enter the price and quantity of the part to be added to the order, (3) click the Parts menu, Add Part to ListBox sub menu.

The Click event for this menu item must call a function named **ValidData** that validates the following business rules. The data must not be processed if a business rule is violated.

- o Business rule #1: An auto part has to be selected from the Description ComboBox control—this means that just typing in an auto part description is not sufficient (testing for Text = String.Empty is not sufficient). The SelectedIndex property must have a value other than -1.
- o Business rule #2: The PriceTextBox control must contain a numeric value greater than zero.

- o Business rule #3: The QuantityTextBox control must contain a numeric value greater than zero.

The click event code must properly call the ValidData function. When the data is valid, complete these additional computing tasks.

- Compute the Line Total for each part added to the ListBox.
- Display (Add) a string consisting of the Description (from the ComboBox) + Price + Quantity + Line Total (The Line Total is the price multiplied by the quantity) as a string item to the ListBox control. The string of information must be formatted as illustrated in the figure of the form shown above. Note that the same auto part may be listed more than once in the ListBox (you do not need to write code to prevent adding duplicates to the ListBox).
- Accumulate the Order Total to a module-level variable and update the Order Total TextBox control with the accumulated Order Total value.
- Clear the text property for each of the Description ComboBox, Price TextBox, and Quantity TextBox controls. Unselect the Description ComboBox item. Set the focus to the Description ComboBox.
- Auto parts listed for shipment as part of the order in the ListBox control should NOT be sorted.

**5. Parts – Clear Part ListBox menu item (3 points).** In order to start a new listing of auto parts for a different order, the application user will need to have the contents of the ListBox cleared. When this menu is clicked:

- Display a message box to ask the system user to confirm that the ListBox control is to be cleared by using a message box with multiple buttons (Yes and No). The default button is No.

If the application user clicks Yes:

- Unselect the part that may be selected in the Description ComboBox control.
- Clear the Text property of the Description ComboBox control.
- Clear the Text property of all of the TextBox controls.
- Clear the contents of the Part ListBox control.
- Set the value of the Order Total module-level variable to zero (see #4 above for an explanation of this variable).
- Set focus to the Description ComboBox.

If the application user clicks No, do nothing (no action is necessary because the application user did not mean to clear the listing).

**6. Parts – Count Parts in ListBox menu item (2 points).** Display a message box with a message that tells the application user how many different auto part items are in the ListBox control (NOT the ComboBox control). Consider for example the figure shown above. The Count = 2 for this example (brake pads and an injector), NOT 6 because your code should count the line items, not the quantity total). Design an appropriate message and use an appropriate title bar entry and appropriate icon/button combinations for the message box.

**7. Parts – Add Part to ComboBox (9 points).** The application user may add a new auto part item (such as an air filter or head light bulb) that is not currently listed in the Description ComboBox. The application user should type the new auto part item into the Text portion of the Description ComboBox, and then click this menu. When the menu is clicked, add the new auto

part item to the Description ComboBox (collection property). Do not allow blank or duplicate auto part items. If the application user tries to add blank or duplicate auto part item, display an appropriate error message. Auto part items should display in sorted order. An example of a duplicate: Air Filter is a duplicate of AIR FILTER.

**8. Parts – Delete Part from ComboBox menu item (2 points).** Delete the currently selected Description ComboBox item (without asking the application user for confirmation to delete). If an auto part has not been selected, display an error message telling the application user to select an auto part to delete. Ensure that the contents of the Text property of the Description ComboBox control are cleared if an auto part is deleted.

**9. Help – About menu item (1 point).** Use a message box to display your name and the current date and time with an appropriate icon. The output should display your name on the first line and the current date/time on the second line.

**10. Context menus (3 points).** Add a context menu to the form that will function when you right-click anywhere on the form (except for the form's title bar – context menus that you add to a form do not apply to the form's title bar).

- The context menu should have two entries: **Add Part to ListBox** and **Exit** with a separator bar between the two entries.
- When the Add Part to ListBox context menu item is selected, use one of the techniques covered in your notes to call the existing sub procedure for the Add Part to ListBox menu item.
- When the Exit context menu item is selected, use one of the techniques covered in your notes to call the existing sub procedure for the Exit menu item.

**Other Requirements.** Ensure that you follow all standards for programming that you learned and used in earlier computer programming assignments including the use of comments, proper spacing of code, proper spelling, proper indentation of code, etc., as listed in the assessment section given below.

**Test the Lab.** Use the assessment guidelines provided below to test your lab work. This is the same set of assessment guidelines that will be used to grade your work.

**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 11 (40 points possible).**

**Before Startup – Form Design (5 points; -1 for each mistake, maximum of -5 plus any deduction for late submission).**

- 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 is named correctly: **Lab7-SectionTime-YourLastName-YourFirstName**.
- The form's File Name property is set Lab7.vb, NOT Form1.vb.
- Title bar of the form has the required information as shown in the figure.
- Form has a good appearance with all controls aligned and sized appropriately so that the appearance is pleasing and professional. Errors include misspelled words, having too much or not enough gray space around controls, controls not aligned or controls too small to display all required information.
- All TextBox controls have the correct TextAlign, ReadOnly, and TabStop settings.
- The Description ComboBox has DropDownStyle = DropDown and has the required initial items stored in the Items (collection) property.
- Initial focus is on the Description ComboBox.
- Program code has the required remarks that identify the program, programmer name, and date programmed.
- Each sub procedure has remarks statements to identify what the sub procedure does.
- Code is organized by regions (-1).
- All variables and controls are named properly following the naming convention taught in the notes and in class (maximum deduction -2).

**Startup (2 points; -1 for each mistake; maximum of -2).**

- Form starts up centered on the screen. Form is an appropriate size.
- Tab order is correct.
- Any missing menu item will result in a deduction for the required coding of that menu (if the menu item is missing, then the code cannot be executed even if you have written the code).

**File – Exit menu item (1 point).** Exits the application by asking for the application user's confirmation.

**Parts – Add Part to ListBox menu item (9 points).**

- Calls **ValidData** function. (-2 if the function is not called properly)
- Displays correct string to the ListBox control (-5 if fails to display any string to the ListBox; -1 for not displaying the correct selected repair part; -1 for not displaying quantity; -1 for not displaying price; -2 for not displaying Line Total cost).
- Correctly updates the Order Total TextBox control (-3 for wrong total).
- Clears the text property for each of the Description ComboBox, Price TextBox, and Quantity TextBox controls (-1 for each item not cleared properly).
- Unselects the Description ComboBox item.
- Set the focus to the Description ComboBox.

**ValidData Function (3 points).**

- Description ComboBox controls is validated correctly (-1).
- Price TextBox and Quantity TextBox controls are validated correctly (-1 each).

**Parts – Clear Part ListBox menu item (3 points).**

- Displays a message box to confirm clearing the ListBox and defaults the button to No.
- Unselects the item that may be selected in the Description ComboBox control and ensures the Text property is cleared.
- Clears the Text property of all of the TextBox controls.
- Clears the contents of the ListBox control.
- Sets the value of the Total Order Value memory variable to zero.
- Sets focus to the Description ComboBox.

**Parts – Count Parts in ListBox menu item (2 points).**

- Displays a message box with a message that includes a count of the items in the ListBox (not ComboBox).
- Message box has appropriate icon/button combination and title bar.

**Parts – Add Part to ComboBox (9 points).**

- Correctly adds a new auto part item to the ComboBox (-9 for doesn't work at all).
- Allows duplicate or blank items (-2).
- Displays appropriate message box when the user attempts a duplicate or blank add (-2 if not displayed).
- Handles duplicate entries regardless of the use of upper/lower case.

**Parts – Delete Part from ComboBox menu item (2 points).**

- Correctly deletes a selected auto part item from the ComboBox (-2 if fails).
- Displays appropriate error message if an item to be deleted has not been selected.
- Clears Text property of ComboBox if an item is deleted.

**Help – About menu item (1 point).**

- Displays required message box with your name and the current date and time on two separate lines; Includes an appropriate icon.

**Context menus (3 points).**

- Has the required context menu with two entries (-3 if context menu does not work).
- Context Menu Item Coding – uses one of the techniques covered in your notes to call the appropriate existing sub procedures (-2 if one or more are not called properly).

**Other Errors.**

- During program execution, your computer program should compile and execute without generating any error messages – if the program will not compile or if it generates error messages during execution, you will lose up to 30 points depending on the severity of the error.

End of Lab