

Objectives:

Use arrays/vectors, files, searching arrays, manipulating array contents, characters, and strings.

You are asked to implement a car ordering system for **Bobcats Auto Dealership**. This dealership is brand new and only sells one brand of cars with three different models. However, a buyer can add options if they choose.

Write a C++ program that allows the user to order a single car with different options. All the options available are stored in a file called "**options.txt**" along with their prices. The user should be able to select the car model, display options and prices, add options, remove options, or cancel the order. Allow the user to see all the available options and their prices. **The program should always display the car model, price, and the options ordered so far.** If the user has not selected a car model, then an error message should be displayed indicating that the order has not started yet (e.g. NO MODEL SELECTED). A user should not be able to add more than 6 options to the car.

The base prices of the three models: **E** (\$10,000.00), **L** (\$12,000.00), and **X** (\$18,000.00).

Implement the following menu options in separate functions:**1. Select a model (E, L, X)**

The user should enter either E, L, or X (either in lower or upper case). If the wrong character is entered, the user should be prompted repeatedly until the user enters a valid model. Update the order information after this selection. Update the model if a model had been selected already.

2. Display available options and prices

List all the options 3 per line. **They must be formatted neatly as shown below.**

3. Add an option

The user should enter an option such as "CarPlay", "10 Speakers", etc. The option entered must be one of the options available. If it's not, the user should be prompted **repeatedly** until the user enters a valid option or enter "**cancel**". After the selection, the order information should be updated. (see sample input/output below). Duplicate options should not be allowed.

4. Remove an option

Allow the user to remove one of the options from the list of options added earlier. Update the order information. If the option name is not in the list of options, then it should be ignored.

5. Cancel order

Start over. Update the order information.

6. Quit

Hints:

- Use an array/vector to store all the options and keep track of the number of options read.
- Use an array/vector to store all the options' prices.
- Use an array/vector of strings for the selected options and keep track of the number of options ordered.
- Write a function to convert a string to lowercase for comparison.
- Write a function that returns true if an option is valid
- Write a function that returns true if an option has been already added to the selected options array/vector

Sample input/output:

NO MODEL SELECTED

```
1. Select a model (E, L, X)
2. Display available options and prices
3. Add an option
4. Remove an option
5. Cancel order
6. Quit
Enter choice: 3
```

NO MODEL SELECTED

```
1. Select a model (E, L, X)
2. Display available options and prices
3. Add an option
4. Remove an option
5. Cancel order
6. Quit
Enter choice: 2
```

Prices for model E, L, & X: \$10000.00, \$12000.00, \$18000.00
Available Options

| | | |
|---------------------------|--------------------------|----------------------|
| Leather Seats(\$5000) | DVD System(\$1000) | 10 Speakers(\$800) |
| Navigation System(\$1400) | CarPlay(\$500) | Android Auto(\$500) |
| Lane Monitoring(\$2000) | 3/36 Warranty(\$800) | 6/72 Warranty(\$999) |
| Dual Climate(\$1500) | Body Side Molding(\$225) | Cargo Net(\$49) |
| Cargo Organizer(\$87) | 450W Audio(\$700) | Heated Seats(\$1000) |

NO MODEL SELECTED

1. Select a model (E, L, X)
2. Display available options and prices
3. Add an option
4. Remove an option
5. Cancel order
6. Quit

Enter choice: **1**

Enter the model (E, L, X): **1**

Model: L, \$12000.00, Options: None

1. Select a model (E, L, X)
2. Display available options and prices
3. Add an option
4. Remove an option
5. Cancel order
6. Quit

Enter choice: **3**

Enter option: **450w aUdIO**

Model: L, \$12700.00, Options: 450W Audio

1. Select a model (E, L, X)
2. Display available options and prices
3. Add an option
4. Remove an option
5. Cancel order
6. Quit

Enter choice: **3**

Enter option: **carplay**

Model: L, \$13200.00, Options: 450W Audio, CarPlay

1. Select a model (E, L, X)
2. Display available options and prices
3. Add an option
4. Remove an option
5. Cancel order
6. Quit

Enter choice: **4**

Enter option to remove: **carplay**

Model: L, \$12700.00, Options: 450W Audio

Grading:

Programs that contain syntax errors will earn zero points.

Programs that do not include functions other than main, will earn zero points.

Programs that use global variables other than constants, will earn zero points.

Programs that use libraries not discussed in class will earn zero points.

Your grade will be determined using the following criteria:

- **Correctness: the program works as requested above (55 points).**
 - Displaying menu and current cost of the car (5 points)
 - Selecting the car model (5 points)
 - Add an option (10 points)
 - Remove an option (10 points)
 - Cancel order (5 points)
 - Display available options and their prices (10 points)
 - Checking for valid option using lowercase or uppercase letters (5 points)
 - Checking for valid car model using lowercase or uppercase letters (5 points)

- **Documentation and Style (5 points)**

Follow the coding style outline on GitHub:

<https://github.com/nasseef/cs/blob/master/docs/coding-style.md>

Submit a link to your repository on Blackboard.