

# CMPSC 122

## Midterm Project

### Spring 2020

**DUE: Last Thursday Of Regular Classes (Drop Box will close)**

Write a program for a car dealership that has a database of information for different automobiles. Each automobile has one or more attributes associated with it (color, Looks sporty, etc.). The number of attributes that each automobile has is not known at this time (and can be different for each). The program needs to accept commands with the goal of providing a list of automobiles that fit your criteria.

Input:

1. The list of automobiles and attributes should be stored in a file called autodata.txt. This file is to be read at the beginning each time your program is run (if it exists). Your program will create this file. The format for this file shall be:  
NameCar1,Attribute1,Attribute2,...  
NameCar2,Attribute1,Attribute2,...  
This is just a comma separated list with the car name first followed by all of its attributes. I will also check your programs with a data file that I generate so the file **MUST** be in this format.
2. The input formats as well as the commands are in the table below. Also, the entire program shall not be case sensitive.

Output:

1. The responses to the user's commands.
2. Each time you execute the SHOW command, also place the results in a file called results.txt. This file can be any format, however, make sure it is labeled clearly. Also, this file should append these results to previous runs.
3. When new cars along with their attributes are added using the ADD command, this data must be persisted to the autodata.txt.

Commands:

Command	How it is processed
<b>ADD</b>	Adds a new automobile. Prompt user for car name and the list of attributes. Shall not work if in search mode
<b>Search</b>	Puts program in search mode. Prompt for name of this search (should probably be a heading in the results.txt file).
<b>hasFeature</b>	Prompt user to enter an attribute to look for. Then process your list of current cars that match this criterion. If there is only one car left, print out the name of this car and that you have found it (this case would terminate search mode). (Search Mode Only)
<b>CheckAuto</b>	Input the name of an automobile and print out the attributes about it if it exists in our search list. (Search Mode Only)
<b>Show</b>	Shows the current list of automobiles that match all of the features you have listed as criteria thus far. (Search Mode Only).
<b>Exit</b>	Exits Search mode and resets all attributes that were entered.
<b>Quit</b>	Save the autodata.txt file if automobiles have been added and exits the program.

Notes:

- This program must be written in C++ using classes with header files. Every function also needs to be properly documented.
- Main should only contain menu code. All other processing must be done in other classes.
- The number of attributes is not known, an array will not work here.
- The CheckAuto feature needs to be fast (a linked list will not work here). Use a binary search tree. This tree should be a copy of the original since it should only contain the vehicles that currently match all of the features that have been added for this search.
- The HasFeature does not have to be as efficient and can traverse the entire data structure to find matches and eliminate ones that don't (again, this should use the copy, why look at data that you have already eliminated).

Sample input (obviously a menu should be initially shown with the choices as well as again if an invalid input is entered. All the menu items typed are *italicized*):

*Add*

Name: Mazda 6

Attributes:   AWD  
                  V6  
                  Looks Good  
                  Great gas mileage

*Add*

Name: Honda Pilot

Attributes:   Large  
                  V6  
                  AWD  
                  Looks Like a Box

*Add*

Name: Corvette

Attributes:   Fast  
                  Great sound system  
                  Looks Good

*Add*

Name: DeLorean

Attributes:   V6  
                  Looks Good  
                  Stainless Steel body  
                  Has Flux Capacitor

*Search*

Name: Good Looking Car

*hasFeature*

feature: Looks Good

*hasFeature*

feature: V6

*CheckAuto*

Name: Corvette

Not Found

*CheckAuto*

Name: DeLorean

Attributes:

V6

Looks Good

Stainless Steel body

Has Flux Capacitor

*Show*

Mazda 6

DeLorean

*hasFeature*

AWD

Car Found!!!!

Mazda 6 is your car!

*Quit*

Deliverables:

- Your entire project directory zipped up (include your output files within this directory structure).
- A listing from the output of a test run. Name it output.txt and place it in your zipped folder as well (top directory).
- Code that you used to test your functions should be left in and named appropriately.
- Any “couts” that you used to debug your code should use the “#DEBUG” so that this can easily be turned on or off.
- A simple test plan document (see page 42-43 in your book) as plain text.