

## Programming Assignments #4 and 5

### CS 202 Programming Systems

#### For This Program

With both programs #4 and #5 you will be implementing your solutions using Java. Your goal must be to develop an object-oriented solution but this time implement it in Java. Make sure that your OO Design is not centered around your data structures – your data structures support the design but shouldn't be the primary emphasis of your design. You may use Eclipse or IntelliJ to develop your software. Your Java programs must follow these rules:

- **No public or friendly fields (data members) \*\*\* NONE!!!!**
- No friendly methods (member functions); this means the public, protected, or private keyword should be placed in front of each method.
- Limit your use of static methods – these should be restricted to just utility functions and main
- With your OO design, develop an inheritance hierarchy using “extends”; there must be a minimum of 5 classes with 3 of them in a hierarchy. *These should not be isolated to just your data structures.*
- The application that USES the hierarchy must be in a class or hierarchy of its own.
- Constructors:
  - Implement at least one constructor with arguments
  - Use the super keyword in invoking a base class' constructor. *This is what we use instead of an initialization list. \*\*\*Write about this in your*
- Support dynamic binding with this programming assignment
  - In your OO design, look for the self-similar behavior between classes – that is where dynamic binding applies
  - In your design write-up discuss the difference between the functions that are being overridden versus those being overloaded
  - Have at least one abstract base class
- Implement at least two functions using function overloading between classes and experiment with the way function overloading works in Java.
- And yes, you SHOULD use the string class!

You are required to turn in a paper on how this solution is object oriented (your design). There is ONLY ONE design writeup for the combined programs 4-5. But, with EACH program, you are required to write 400 words about the efficiency and IDE (rather than the debugger). For each of the above that you experiment with, write up information about it in your efficiency write-up.

## Program Requirements

Have you ever looked around and noticed so many of the cars look the same. They are tan, silver, white, gray or black. There are a few yellow or red cars, but they are few and far between. Wouldn't it be great if you could go online and put together all of the pieces of a vehicle that you really want – maybe a pink VW with ground lighting, and fender flairs? Or, more seriously, my husband would want a self driving electric vehicle with air ride suspension. I would want one that is off-road capable with a winch and lifted 8 inches!

Your job is to create a program in two phases. Think of it as something that would ultimately be an App to assist people in finding their ultimate vehicle. In Program #4, we will start by putting together all of the different options available for each kind of feature (interior, exterior, engine, suspension, wheels, etc.); you may limit the different types of features available to choose from to just three. Then, allow the user to select from the options available; this is the first step with Program #4. Then, once the options exist, with Program #5 we will have the program find the best match with a list of the cars for sale.

**Design Specifics:** In your design push up the common elements for the different kinds of features that are similar but yet different. Think about how dynamic binding can be used to put together a user's desired options. You should put a priority on each of the options in the event that the cars for sale in their area do not have all of the features exactly as they want. Maybe color isn't as important as the type of fuel (e.g., electric versus gas). Cost is also always an issue, so any feature that has an extra cost involved should include that information.

Your job is to come up with a design of an OO framework. The key is to make sure to solve this problem using Object Oriented methodologies **with dynamic binding** and function overloading. *The use of external data file(s) is necessary! **Don't HARD CODE the options available to select from** – as that would not be real-world! For example, to add lime green as a new paint color shouldn't require hiring a programmer to modify the code!*

## Data Structures

In these last two programs, you must implement two data structures:

1. Program 4: An array of features where each feature has a linear linked list of options available. With dynamic binding, this can be an array of base class references.
2. Program 5: A BST (binary search tree) of car lots (e.g., Carmax) organized by the name of the lot. *Each node will also have a root pointer to another BST of the vehicles with their linear linked list of features.*

The required data structures specified in the assignment must be your own implementation: as in BSTs and LLLs. Once you meet the basic requirements of the assignment, you are allowed to use libraries for any subsequent data structures.