***Directions:*** In this lab you will develop a large number of classes that inherit from one another. You will create constructors for each class and chain the appropriate constructors together. You will then sort objects of these classes in various ways by implementing the Comparable Interface.

***First start by designing an inheritance hierarchy for the following classes on paper*** (that means not on the computer). Then implement the methods shown below on these classes. Given careful thought to which methods should be in which classes. More general methods like getType() should be implemented higher up in the inheritance hierarchy. Methods particular to a subclass class like getGrossTonnage(), which only applies to WaterVehicles should be implemented there.

***You need to show your design to your instructor before proceeding with the coding.*** The design must be on paper. It should have a picture of the inheritance hierarchy between the classes as well as a list of which methods are defined in which classes.

Here are the classes you must implement:

Vehicle

Boat

Submarine

Car

Motorcycle

WaterVehicle

LandVehicle

***You must implement the following methods for the above classes.*** Note that not all the methods apply to all classes, above. You will also need to implement state variables to track information for each class.

**Accessor Methods Needed**

getType() - returns a String with the vehicles type. Some examples of type would be “Car”, “Boat”, “WaterVehicle”, etc. This method should be implemented at a high level.

getColor() - returns a String

getGrossTonnage() - returns a decimal. Only applies to vehicles on the water.

getWeight() - returns a decimal. Applies to all vehicles.

getNumberOfWheels() - this method is only appropriate for vehicles that travel on land.

getMaxSpeed() - returns an Integer, units are miles per hour.

getPriceWhenNew() - returns a decimal.

***You must also provide mutator methods that correspond to each of the methods listed above. So for example, the getColor() method should also have a setColor() method to go with it.***

***You must provide explicit constructors for each class.*** Look in the tester class for hints on which arguments the constructors should take.

***You must overload the toString() for each class, above***. Each toString() method should print all the information about the class on a single line. Do not rewrite each toString() method from scratch. Instead, call the parent toString() method and append any information specific to the current class.

Once you are done with all of the above, go back and implement the Comparable interface on the highest level (Vehicle) class. You demonstrate the following sorting capabilities to your instructor. For each case, print the ArrayList before you sort and again after you sort.

1. Sort the vehicles according to their max speed.
2. Sort the vehicles based on their price. (Rewrite the compareTo() to do this).
3. Sort the vehicles based on their color. But if the colors are identical, further sort the vehicles based on their type. (Again, rewrite compareTo()).

The tester class for this project is provided to you in the same directory as this document.