Homework 4 – Inheritance
Code Refinement of the Ordering System

Due next week class 5a at beginning of class

Orders for Truck & Car

Objectives:

This homework 4 will gain you experience with inheritance, adding to, and maintaining your previous code. It is to also clean up some of the input.

Maintenance of code:

This homwork builds on the existing code you wrote before. Do not write that code aga, use it, modify it. Any errors indicated in the previous homework should be corrected as the error(s) may affect your ability to complete this homework. Maintenance of code consists of fixing and enhancing existing code.

Enhancements:

Starting with your other homeworks Ordering System classes, the second and minor enhancement will be adding a Boat class. Think of all that new duplicate code you have to write! Not a fun prospect, is it? That is why it is the second item to do.

The first enhancement will be to simplify the Truck and Car classes by using inheritance. In Orders, the choice of Truck and Car need to grow with the addition of boat, so we want to make their use (Truck's, Car's, Boat's) more dynamic. This allows many more vehicle types to be added having minimal impact on the program.

`Inheritance`

Write a class, Vehicle, which contains the commonalities of Car, Truck and Boat. They are the **attributes**, **accessors** and **mutators** for Model, Color and Cost. You may add any other common methods to Vehicle as you see fit, such as a menu for these items.

`Options for a vehicle (a menu)`

Use String arrays (not Collections) for the choices in an option. A towing package for example would be:

```
final String [] TOWING = {"Towing package","No towing package"};
```

Change your application to allow for adding to the array, then recompiling; and not having any other changes in your code. This is for the options that print in the menu style. The options are placed in different String constants. To print the options, you use a loop in a separate menu-method. This common menu-method can be added to Vehicle and used by any class that inherits Vehicle. A loop would print the number choices 1. , 2. , etc. with the `TOWING[i]` text beside it, followed by the line "Choice:". See the output example at the end of this assignment. See menu-method description below.

Menu-method for entering menu choices

Write a method, showMenu, that accepts a String prompt and array of choices.

```
showMenu( String prompt, String [] choices )
```

The method then displays a menu, then returns the selected index (zero based) integer from the menu that this method displayed. Also handle any exceptions in this method. Only exit this method when you have a valid menu response. Your code can then be simplified by using this one method for all menus prompts in all classes using Vehicle. Truck, Car and Boat classes no longer have any keyboard input or printing. This has been moved to the Vehicle class. Once this conversion is finished, you will see that adding Boat or other classes are much faster than we programmed it before.

OOP

- Any access to instance variables (except constants) MUST use accessors or mutators. Even when the variables are accessed within the same class.
- UML required, showing the proper relationship between the classes. Indicate attributes, and major methods. Do not need to show accessors and mutators in UML.

Requirements summary:

- Vehicle:
  - Vehicle class, which Truck, Car, Boat inherit.
  - All common methods and attributes used by Truck, Car, Boat are only in Vehicle.
  - Vehicle contains a menu presentation method.
- Java Docs – modify where needed, add Java Docs to the new code.
- Inputs to the program are using keyboard input, MUST be in order and of the correct data type as shown (or 15 points deduction). That is, menu's must accept numeric entries for the choices. For example, you may set the Truck Engine Size to say any two size options you want, but they must be asked at the correct time and the number choices must be 1 and 2.
- All constants must be defined. Exceptions are Headings, and messages.
- Make corrections from previous homework.

Submitting your work:

- Submit all of your java and class files to the homework DropBox.
- Do not include your JavaDoc files to this homework DropBox.

Sample Output: (Bolded text was entered by the user)

```
C:\>java Orders
<Your Name>
Java for Programmers ISTE-200 <this year and semester eg: 2014-fall>
Homework # 4

Do you want to order a Truck (T/t), Car (C/c) or Boat (B/b)? c
Entering Car order:
      Model:  Ford
      Color:  Green
      Cost:   19875.95
What type of car is this?
      1. Sedan
      2. Coupe
      3. Wagon
      Choice:     3
Does this car have a towing package?
      1. Towing package
      2. No towing package
      Choice:     2

Do you want to order another vehicle? y
Do you want to order a Truck (T/t), Car (C/c) or Boat (B/b)? t
Entering Truck order:
      Model:  Dodge RamTruck
      Color:  red
      Cost:   25000.95
What size truck is this?
      1. Half-ton
      2. Full ton
      Choice:     1
What is the engine size of the truck?
      1. Really big
      2. Not so big
      Choice:     2

Do you want to order another vehicle? Y
Do you want to order a Truck (T/t), Car (C/c) or Boat (B/b)? B
Entering Boat order:
      Model:  SeaCraft
      Color:  white
      Cost:   9250.50
What type of boat is this?
      1. Pontoon
      2. PWC
      3. Sailboat
      Choice:     3
What is the boat's construction?
      1. Wood
      2. Fiberglass
      3. Steel
      Choice:     2

Do you want to order another vehicle? n

Car
      Model:  Ford
      Color:  Green
      Cost:   $19875.95
      Type:   Wagon
      Towing: No towing package
```

```
Truck
      Model:  Dodge RamTruck
      Color:  red
      Cost:   $25000.95
      Load:   Half-ton
      Engine: Not so big

Boat
      Model:  SeaCraft
      Color:  white
      Cost:   $9250.5
      Type:   Sailboat
      Made of:Fiberglass


Thank you for using Your Name's Ordering System.
C:\>
```

Improvements in the spacing of the output is encouraged.

ISTE-200 Homework 4 grade sheet

Item	Points available			Points earned		
Order.java (20%)						
Adds truck or car to a collection	5					
Prints the truck/car summary	5					
Javadoc comments	5					
Works as expected	5					
Vehicle.java (20%)						
Allows entry of common attributes,	5					
Accessors and mutators: Model, Color, Cost	5					
`showMenu`, method to produce a menu and returns menu choice	5					
toString() method of attributes. Method uses accessors to get data	5					
Boat.java, Truck.java, Car.java (60%)	**B**	**T**	**C**	**B**	**T**	**C**
Method to have Vehicle show menu for boat attributes	5	5	5			
Accessors and mutators	5	5	5			
toString() using accessors to get data	5	5	5			
Proper use of inheritance, works as expected	5	5	5			
Deductions: Violations of coding standards or other violations						
Total	100					