Lab 1

**NOTE**: For this lab, you will get 5 points for a valid attempt if turned in by the due date.

The main new component in this lab, inheritance, is in Chapter 2 and were reviewed in class.

**TIP**: Remember, don't be overwhelmed by all the details. Simply focus on one line at a time as the instructions walk you through the creation of the app.

1. **SUMMARY**
   1. Make a java application that creates vehicle objects using an inheritance class hierarchy and is able to start and stop them based on user requests.
   2. This lab will involve the following new features:
      1. Inheritance.
2. **DETAILS**
   1. Classes
      1. **MotorVehicle**
         1. Declare at least two instance variables in the MotorVehicle class that each child class (Motorcycle and Car) will inherit.
            1. One instance variable must be a boolean that is called "running".
            2. The other instance variable is up to you... "color", "make", etc.
      2. **Motorcycle** and **Car**
         1. Have them inherit from MotorVehicle.
         2. Declare one instance variable in each of these classes that is unique to the type.
            1. For example, you could put a "doorCount" int variable in the Car class and a "windshield" boolean variable into the Motorcycle class.
         3. In the Motorcycle and Car classes, make two methods in each: start and stop methods.
            1. The start and stop methods will first change the "running" variable.

In the start method, set the running variable to true.

In the stop method, set the running variable to false.

* + - * 1. The start and stop methods will then print to the screen:

The class's two inherited instance variable values.

The specific class's instance variable value (such as "doorCount" for the Car class).

What type of vehicle it is (Motorcycle or Car).

Whether it's been stopped or started.

* + 1. **VehicleManager**
       1. Two instance variables that are arrays, one of Car array type and one of Motorcycle array type.
       2. A "run" method that does the following:
          1. Instantiates three objects of Car type and three of Motorcycle type.
          2. Sets the "running" variable on each to false.
          3. Fills the other instance variables on each object directly in code with any value you wish.

So if you had "color" as a variable of String type in MotorVehicle, set one to "red", one to "green", etc.

* + - * 1. Put all the objects into the two arrays mentioned above.

(car objects in cars array, motorcycle objects in motorcycles array)

* + - * 1. Then displays a series of things to the user with the following flow:

First a menu for choosing which car to START:

Lists the cars with numbers and asks which one to start.

Use 1 to 3 as menu item numbers instead of 0 to 2.

When item # chosen, then the start method for that object is called.

The number they enter has to be adjusted for the correct array index obviously.

Second is the menu for which motorcycle to START:

Same as above except now for the motorcycle array.

Third is the menu for which car to STOP:

Same as the ones above except now the stop method is being called instead of start.

Fourth is the menu for which motorcycle to STOP.

Same as above, but now for motorcycle.

Fifth, display all 6 vehicles status on whether they're running or not.

* + 1. **Vehicle\_Test**
       1. Put the main method here.
          1. In the method, simply instantiate an instance of VehicleManager and call its “run” method.

1. **TURNING IN LAB:** 
   1. Turn your lab into Canvas where you got this lab doc.
   2. IMPORTANT: DO NOT ZIP your folders/files please.
   3. Turn in just the **five raw java files (.java files)** for this lab (don’t turn in any .class files).