M359 – AP Java Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Unit 3 Lab Teacher: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Per: \_\_\_\_

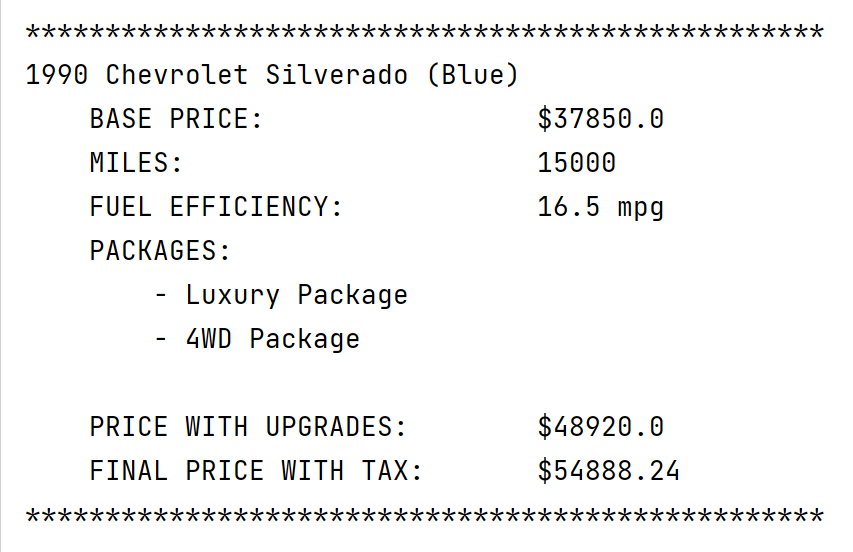
**Unit 3 Chevrolet Lab**

You’re just about to turn 16 and it’s time to get a ride! You visit your local Chevy dealer and overhear the general manager say “I wish we had a program that would allow us to track our new and used vehicles.” You walk up and offer your computer science services in exchange for a used car, and to your amazement she says yes! She outlines the requirements for the program, the details for each of their popular vehicles, as well as the upgrade options their dealership offers.

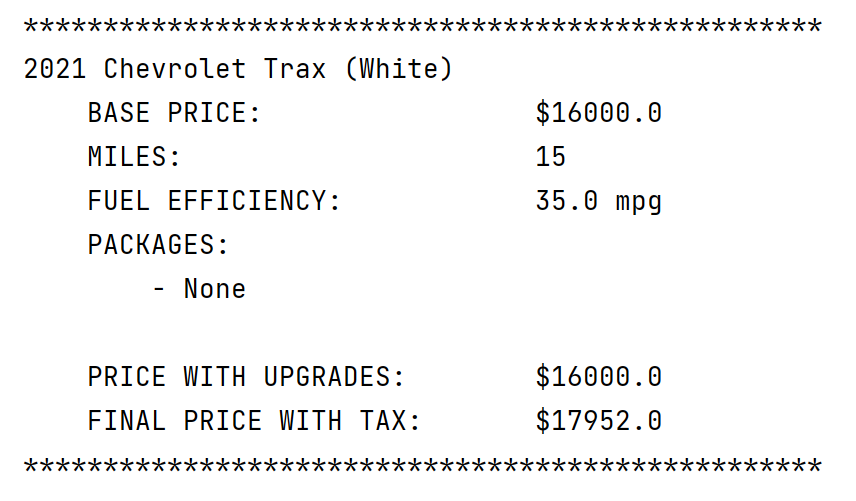
**Program Requirements:**

***Chevy Class***

* Vehicle Details: year, mileage, fuel efficiency, base price, price with upgrades, grand total, model, color, luxury package status, 4WD package status, and sports package status (constructor is in this same order!)
* Vehicle Constants: vehicle make of “Chevrolet”, tax rate of 12.2%, appropriate luxury package price increase, 4WD price increase, sports price increase, and fuel efficiency deduction (see calcPrice() method description)
* Constructor for each type of vehicle
  + Default Constructor
    - A default car is considered a White 2021 Trax with 0 miles that costs $16,000. The car gets 35 mpg, and has no additional packages. At the end of the constructor, all instance variables should have an initial value.
  + Full Constructor
    - The user is able to fully customize every attribute of a car. Make sure to check the driver program to see the required order of these variables! At the end of the constructor, all instance variables should have an initial value.
  + compareTo()
    - Compares one Chevy with another relative to the mileage of the vehicle
  + equals()
    - Two Chevy vehicles are considered “equal” if the following criteria are the same:
      * Vehicle’s model and color
      * Vehicle’s new/used status. A vehicle is considered new if it has under 200 miles, and used if it has 200 miles or more
  + Getters and Setters
    - Included for all instance variables (not needed for constants).
  + toString()
    - The returned String should be formatted identically like these examples:
      * Ex: Blue 1990 Chevrolet Silverado with 15,000 miles and a base cost of $37,850. The truck gets 16.5 mpg and has both a 4WD and luxury package.



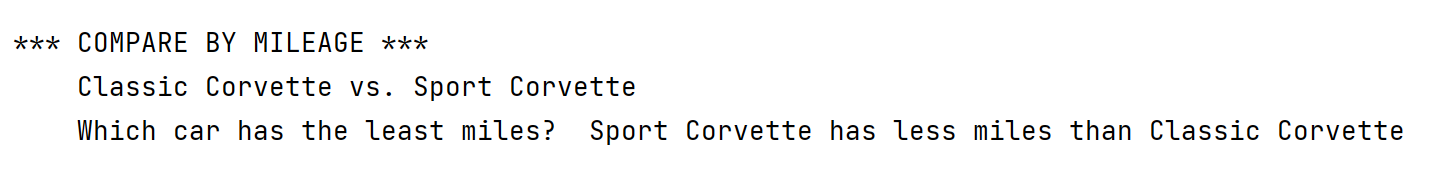
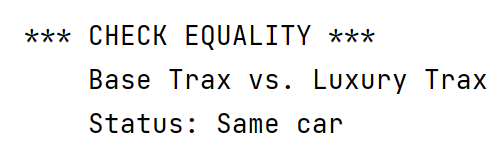
* + - Ex: White 2021 Chevrolet Trax with 15 miles and a base cost of $16,000. The car gets 35.0 mpg and has no packages.



* + calcPrice()
    - Checks a vehicle for the status of a Luxury Package, a 4WD Package, and a Sports Package. The added costs for these upgrades are as follows:
      * Luxury Package: 20% (of the base price) increase
      * 4WD Package: $3,500 price increase
      * Sports Package: 15% (of the base price) increase AND 20%   
         decrease of fuel efficiency
    - The grand total of the vehicle will include a 12.2% sales tax.
    - POSTCONDITION: Upon completion, the price with upgrades (pretax) and the grand total (includes tax) should reflect the extra costs. The base price instance variable should NOT be changed.
    - Keep in mind, that if any of the variables that effect cost are changed with set methods, these values should be recalculated.

***ChevyDriver Class***

This class is partially incomplete, and will require you to complete the following sections in their respective areas in the program:

* Update Chevy Objects
  + traxBase has its mileage set to 15 miles
  + traxLux has its mileage set to 175, luxury package added, and 4WD package added
* compareTo() Logic
  + Complete the empty decision statements so the correct message prints accordingly  
    
* equals() Logic
  + Complete the empty decision statement so the correct message prints accordingly  
    

**Additional Test Cases: Here are several Chevy definitions. You can test your toString and math calculations by adding in these lines to your driver and then printing the objects.**

**Chevy corvetteClassic = new Chevy(1963, 123500, 13.5, 61500, "Corvette (Classic)",**

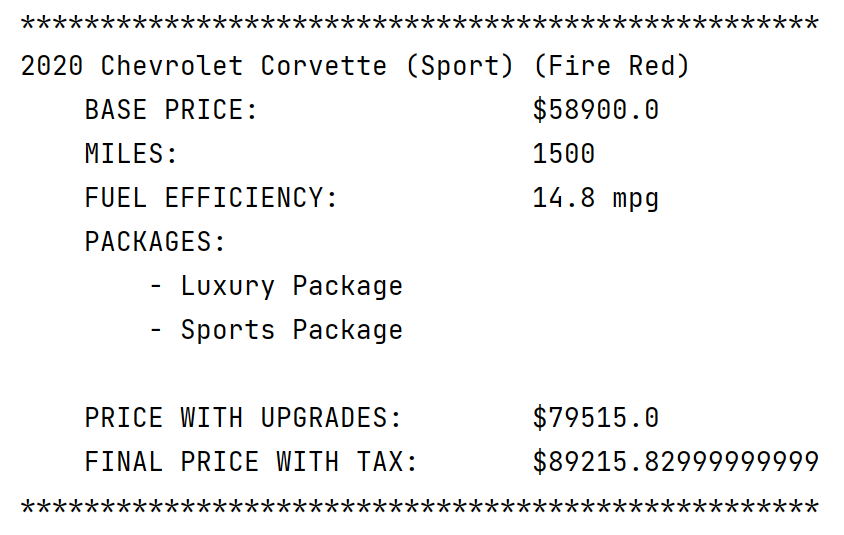
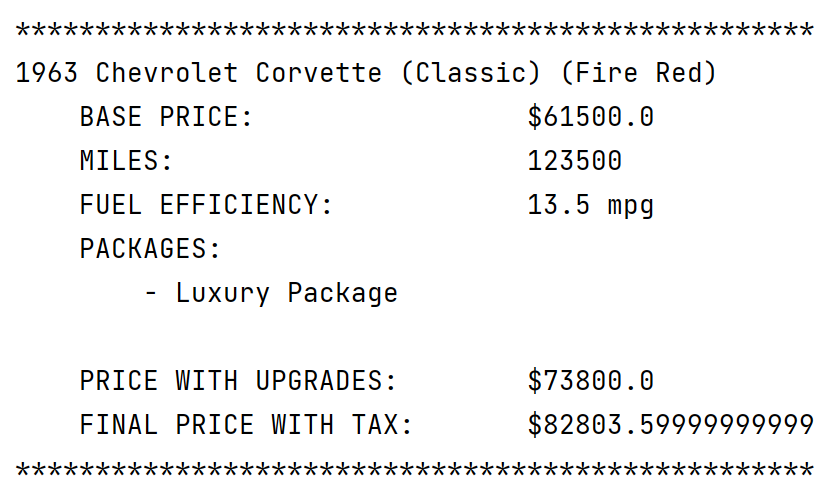
**"Fire Red", true, false, false);**

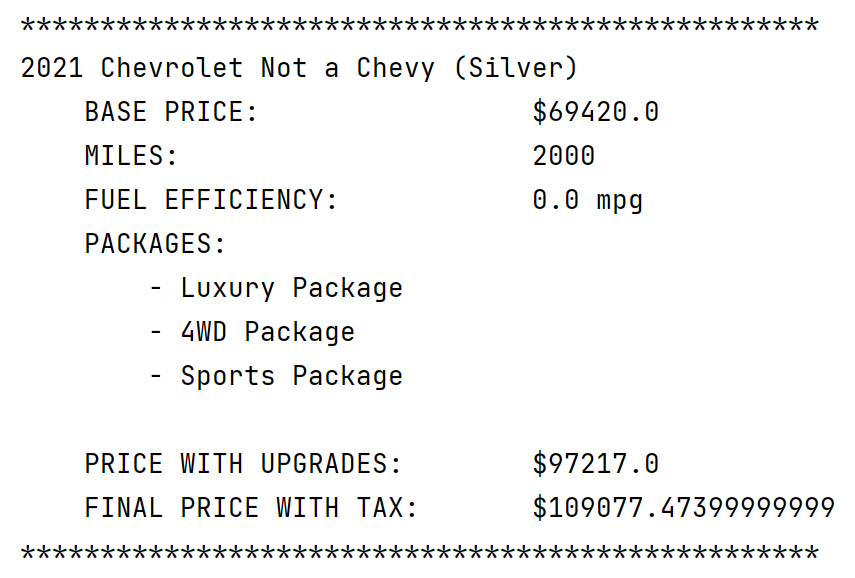
**Chevy corvetteSport = new Chevy(2020, 1500, 18.5, 58900, "Corvette (Sport)",**

**"Fire Red", true, false, true);**

**Chevy moonTesla = new Chevy(2021, 2000, 0, 69420, "Not a Chevy",**

**"Silver", true, true, true);**

****

****