Introduction to Computers and Engineering Problem Solving
Spring 2012
Problem Set 3: Buying a used car
Due: 12 noon, March 2, 2012

Problem statement

You want to buy a used car, but need help deciding, so you create a formula that will help you rate available used cars, given their characteristics and some user reviews that you find online:

Score = (26,000 – Price)/3000 – 0.2*(Years old) + 0.2*(MPG-25) + Driver rating

Assume that the Driver rating for a car is (G + (0.5*O) - B)/T, where G, O, B, and T are the number of Good, OK, Bad ratings, and Total number of ratings, respectively.

	Toyota	Honda	Chevrolet	BMW
Price	18,000	20,000	17,000	26,000
Years old	0.5	1	1	4
MPG	26	25	27	23

Driver rating	Toyota	Honda	Chevrolet	BMW
Bad	3	1	1	0
OK	2	2	4	1
Good	1	6	5	3

Program

Your assignment is to write a program containing classes and objects to decide which car to buy. You will need the following classes:
- A `UsedCarLot` class
- A `Car` class
- A `DriverRating` class
- A test class that only has a `main()` method to create and use the necessary objects.

The `UsedCarLot` class has space for four `Car` items. Each `Car` has a `DriverRating` object that contains the breakdown of the ratings given by reviewers. **You must have a `DriverRating` class; you may not put the ratings directly in the `Car` class.**

Your `main()` method does not need to accept any user inputs. All data can be hard-wired into the program. Add additional methods as appropriate in the classes listed above to enable your `main()` method to accomplish the following tasks.

1) Create the necessary`UsedCarLot`, `Car` and `DriverRating` objects. Create each object using the data in the tables from the *Problem Statement* section above.
2) Call a method in the appropriate class that will give the driver rating for each of the cars and output the result to the console. For this part, access the `Car` objects through the `UsedCarLot` object.
3) Call a method in the appropriate class that will compute the overall score for a given car. Call this for each `Car` and output the result. For this part, access the `Car` objects directly.
4) You read more reviews online, one Bad review for Chevrolet and three Good reviews for BMW. Call a method in the appropriate class that will update the appropriate data. Output the new overall scores for these two `Cars`.
5) Call a method that prints the make of the `Cars` in the `UsedCarLot` with an age of one year old or less, out of the set of four cars.
6) Some people will want to avoid "lemons". Call a method that returns the percentage of bad reviews for each car.

Your `UsedCarLot`, `Cars` and `DriverRating` classes must have the appropriate data members, constructors, `getXXX()` methods, `setXXX()` methods and other methods in order to complete the assignment. All data members must be private in all classes. You only need to write the `getXXX()`, `setXXX()` and other methods required to produce the specific outputs for this homework, as described above. You are welcome to use arrays if you are comfortable with them; they are not required for this assignment.

Example Output:

```
Driver ratings:
Toyota: -0.16666666666666666
Honda: 0.6666666666666666
Chevrolet: 0.6
BMW: 0.875

Over all scores:
Toyota: 2.6
Honda: 2.466666666666667
Chevrolet: 3.8
BMW: -0.325

With 1 additional bad review, the updated overall score for Chevrolet is: 3.654545454
With 3 additional good reviews, The updated overall score for BMW is: -0.2714285714

Cars in lot newer than one year:
Toyota
Honda
Chevrolet

Percent of bad ratings:
Toyota: 50.0%
Honda: 11.11111111111111%
Chevrolet: 18.181818181818183%
BMW: 0.0%
```

Turn In

1. Place a comment with your full name, section, TA name and assignment number at the beginning of the .java file for your solution.
2. Place all of the files in your solution in a single zip file.
    a. Do not turn in copies of compiled byte code or backup (.class or .java~ files)
    b. Do not turn in printed copies of your solution.
3. Submit the single zip file on the 1.00 Web site under the appropriate section. For directions see *How To: Submit Homework* on the 1.00 Web site.
4. Your uploaded files should have a timestamp of no later than noon on the due date.
5. After you submit your solution, please recheck that you submitted your .java file. If you submitted your .class file, you will receive **zero credit**.

Penalties

- 30 points off if you turn in your problem set after Friday noon but before noon on the following Monday. You have one no-penalty late submission per term for a turn-in after Friday noon and before Monday noon.
- No credit if you turn in your problem set after noon on the following Monday.

1.00 / 1.001 / 1.002 Introduction to Computers and Engineering Problem Solving
Spring 2012