UNK CSIT 150
Lab 2: Fundamentals of Classes

Objectives

- Practice defining a class
- Practice the use of object data members in a class

General Requirements

In this lab, you must write your code following the proper programming style. The bottom line is:

- Use indentation to show the logical structure of your code
- Use blank line to separate code blocks and give each code block a comment
- Give documentation comment to the class and each method

Programming Practice

Write an application that simulates driving cars.

1. Using the code attached with this program on blackboard, update the **Car** class:
   a. Finish the following methods:
   - **move** – Move the car one minute. (Update the distance traveled and time traveled variables.) Be sure this method is using floating point division, not integer division.
   - **accelerate**- Add 5 miles per hour to current speed. Remember, the current speed cannot exceed the maxSpeed of the car.
   - **brake** - Subtracts 5 miles per hour from current speed. The minimum speed for the car is 0.

   b. Add a **copy** constructor. (This is bonus)
   c. Add public mutator methods for the car owner, maxSpeed, distanceTraveled, and currentSpeed. (The move, accelerate, and brake methods also act as mutator methods for these variables.)
   d. Use JavaDoc documentation for each method.

2. Modify the class named **TestCar**. Using the comments, add statements to finish the program as specified.

Generate JavaDoc (Formal Technical Documents)

Follow the steps below to generate professional technical documents.

1. Write documentation within the documents for each class and method you have just created, if you have not done so.
2. Go to menu "Tools -> Generate Javadoc ..."
3. In the window that pops up, choose the java file for which you want to generate document.
4. Specify the output directory – usually, we name this “docs” and store this in the project folder.
5. Click the "ok" button.
6. Use the file explorer to check the document folder and you will see a file named index.html.
7. Use a browser to open the index.html file and see how the words you input in the documentation comments have been shown on this technical document.

Lab 2: Car Objects

Name(s): ____

Evaluation

Requirement	**Possible Points**	**Points Received**	**Comments**
move Method	2		
accelerate Method	2		
brake Method	2		
Mutator methods for owner, maxSpeed, distanceTraveled, tripTime and currentSpeed	2		
Testcar modified to simulate two other cars, with output of who drove the farthest, and the longest, formatted properly	2		

Programming style	**Possible Points**	**Points deducted**	
Inconsistent indentation	**-1**		Indent at least 3 spaces inside each brace
Poor use of white space	**-1**		Too many or too few lines between statements
Heading documentation, includes programmer names, date, algorithm, basic purpose	**-2**		
All sources not cited (Remember to cite all code used, even class demo code.)	**-1**		
JavaDocs not used on each method	**-1**		
Poor variable names	**-1**		Should not start with uppercase. No single letter variables.
Poor structure/logic issues	**-2**		
Program specifications			