Assignment 13: Car

Part 1

1. You are given the Car and CarExample examples. Run them and make sure you understand what the following words mean: variable, method, input variable, return value.
2. Add the following variables to the Car class:
   1. A String called nickName, to represent the nick name for the car
   2. A String called color, to represent the color of the car
   3. An int called yearsOwned, to represent the number of years the current owner has owned the car
   4. A double called miles, to represent the number of miles the car has driven
3. Change the constructor so that it includes input variables for all the new ones above.
4. Add getters and setters for each of the new variables. There is a shortcut for this: highlight the variables you want to make getters and setters for. Right click on them. Click ‘Insert Code,’ then ‘Getter and Setter’. Check the ones you want, then click generate.
5. Then, add variables and put them in the constructor in the CarExample class so that it doesn’t have an error.
6. Add all the new information to the print statement below. You can use new lines.
7. Run the program and make sure it works as expected.
8. Save the program in Git.

Part 2

1. You are given the CarTest file with unit tests on it.
2. Modify the constructor in the CarTest class so that it works correctly with all your new variables.
3. Add unit tests for all the new methods you added. There should be 14 tests when you are done.
   1. Note that for the milesGetter and milesSetter tests, assertEquals will work a little differently. For doubles, the function takes in the value, expected value, and a double that is the allowed error. It checks if the value is within the error’s distance of the expected value. Use the variable e as your error value.
      1. Ex. assertEquals(c.getMiles(), 67152.2, e); will check if c.getMiles is within e, which is a very small number, of 67152.2.
4. Run the program and make sure it works as expected.
5. Save the program in Git.