LAB EXERCISE

MPG

Background:

1. Professional programmers carefully design the classes they need before any coding is done. With well-designed classes, programming is much easier and the program has fewer bugs. Object-oriented design consists of deciding what classes are needed, what data they will hold, and how they will behave. All these decisions are documented (written up) and then examined. If something doesn't look right, it is fixed before any programming is done.

2. The specifications of a class that models the fuel efficiency of a car would be:

   Variables

```
int myStartMiles;        // Starting odometer reading
int myEndMiles;          // Ending odometer reading
double myGallonsUsed;    // Gallons of gas used between the readings
```

   Constructors

```
// Creates a new instance of a Car object with the starting
//   odometer readings.
Car(int odometerReading)
```

   Methods

```
// Simulates filling up the tank. Record the current odometer reading
//   and the number of gallons to fill the tank
void fillUp(int odometerReading, double gallons)

// Calculates and returns the miles per gallon for the car.
double calculateMPG()
```

Assignment:

1. Implement a `Car` class with the following properties.

   a. A `Car` keeps track of the start odometer reading, ending odometer reading, and the number of gallons used between readings.

   b. The initial odometer reading is specified in the constructor

   c. A method `calculateMPG` calculates and returns the mile per gallon for the car..

   d. A method `fillup` simulates filling up the tank at a gas station: `odometerReading` is the current odometer reading and `gallons` is the number of gallons that filled the tank. Save these values in instance variables.

   e. With this information, miles per gallon can be calculated. Write the method so that it updates the instance variables each time it is called (simulating another visit to the pumps). After each call, `calculateMPG` will calculate the latest miles per gallon.

2. Write a testing class with a `main` method that constructs a car and calls `fillUp` and `calculateMPG` a few times. Sample usage would be

```
Car auto = new Car(15);  // initial odometer reading of 15 miles
auto.fillUp(250, 10);    // odometer is at 250 miles
                         //   fillup with 10 gallons of gas
                         // repeat auto.fillup line for additional fillups

System.out.println(auto.calculateMPG()) // print miles per gallon
```

3. Write a testing class with a `main` method that constructs a car and calls `fillUp` and `calculateMPG` a few times. A sample run of the program would give (values in ***bold italics*** represent input from the user):

```
New car odometer reading: 15

Filling Station Visit
  odometer reading: 250
  gallons to fill tank: 10

Miles per gallon: 23.50

Filling Station Visit
  odometer reading: 455
  gallons to fill tank: 12.5

Miles per gallon: 16.40
```

4. Format the output as shown. Miles per gallon should be rounded to 2 decimal places.

5. It is recommended that the `Car` class and the testing class be in two source files. This will be the general approach in this curriculum guide. Call your instructor to your workstation for scoring.