Homework #3

In this assignment you will be using quicksort to sort an array of car objects by various criteria.

All code implemented in this assignment should be in a class called `Homework3`. You may use the data structures and algorithm code from the lecture notes.

Define a `class Car` as follows:

```
class Car
{
      public String make;
      public String model;
      public int mpg;       // Miles per gallon
}
```

a) **(2 points)** Implement a comparator called `compareCarsByMakeThenModel` that can be passed as an argument to the `quicksort` method from the lecture notes. `compareCarsByMakeThenModel` should return a value that will cause `quicksort` to sort an array of cars in ascending order (from smallest to largest) by `make` and, when two cars have the same `make`, in ascending order by `model`.

b) **(2 points)** Implement a comparator called `compareCarsByDescendingMPG` that can be passed as an argument to the `quicksort` method from the lecture notes. `compareCarsByDescendingMPG` should return a value that will cause `quicksort` to sort an array of cars in descending order (from largest to smallest) by `mpg`.

c) **(2 points)** Implement a comparator called `compareCarsByMakeThenDescendingMPG` that can be passed as an argument to the `quicksort` method from the lecture notes. `compareCarsByMakeThenDescendingMPG` should return a value that will cause `quicksort` to sort an array of cars in ascending order by `make` and, when two cars have the same `make`, in descending order by `mpg`.

d) **(3 points)** Write a `main` method that tests your methods from parts a-c with the following array of cars:

```
Car cars[] = {
     { "Toyota", "Camry", 33 },
     { "Ford", "Focus", 40 },
     { "Honda", "Accord", 34 },
     { "Ford", "Mustang", 31 },
     { "Honda", "Civic", 39 },
     { "Toyota", "Prius", 48 },
     { "Honda", "Fit", 35 },
     { "Toyota", "Corolla", 35 },
     { "Ford", "Taurus", 28 }
 }
```

Your test program should do the following:

1. Output (displaying make, model, and MPG) the cars in original unsorted order.
2. Output the cars sorted (using qksort from the book) by make then model.
3. Output the cars sorted (using qksort from the book) by descending MPG.
4. Output the cars sorted (using qksort from the book) by make then descending MPG.

e) **(1 point)** Make sure your source code is well-commented, consistently formatted, uses no magic numbers/values, and follows programming best-practices.

Turn in all source code, program output, diagrams, and answers to questions in a single PDF document.