Create a program that uses the Car class to fill a garage of cars. Then have a user “drive” one of the cars from the garage. If the car runs out of fuel while it is being driven it cannot be driven again and must be removed from the garage. You may use the Car class you created in the Car assignment.

The specifications for that Car class are as follows:

The Car class:

The Car class must be in a separate package from the any driver/tester program.

The Car class will contain, at a minimum, the following state fields as constants:

* fuel tank size
* make
* model
* year
* fuel economy
* these fields may be public

The Car class will also need at least 3 constructors:

* Car() – a no argument constructor that uses the default field values from the class to set the new Car object’s state variables.
* Car(String, String, String, year, double, double) – accepts arguments to initialize the new Car object with specific make, model, color, year, tank size, fuel economy.
* Car(Car) – accepts a Car argument to initialize the new Car object with the same state variables values as the argument’s.

The Car class must implement the attached interface.

Develop and use an algorithm that calculates the amount of fuel used in the drive() method. The algorithm must use a formula that gives proportionately poorer mileage when the Car is driven faster or slower than the optimal speed. You should set limits to the how poor of MPG your car will get.

You may add other methods and fields as needed.

When a new Car object is created, its mileage is set to a random number between 0.0 and 5.99.

When a new Car object is created with the default constructor, its fuel economy (MPG) is set to a random number between 15.0 and 65.99, and its fuel tank size is set to a random number between 8 and 35.5 gallons.

Hint: Use “helper” methods to generate the random numbers.

* Use Math.random( ) to generate your random numbers. Remember Math.random( ) generates a random double number from 0.0 to but not including 1.0. So, to get a random number between 0.0 and 99.99 you must multiply the result of Math.random( ) by 100.
* example: If Math.random( ) produced 0.5584, multiplying it by 100 would produce 55.84.

Since the new class Car inherits the .equals and .toString methods from the Java Object class, you will need to overload the .equals( ) method and override the .toString( ) method.

Once you have created your Car class, test it using the old testDriver program.

After you are comfortable with the Car class, create a garage using an array of Car types. You must use an TYPE[] not an ArrayList<TYPE> for the garage.

The rules for driving the cars from the garage are:

* The size of the garage is specified by the user.
* You may only use cars from the garage
* If a car is driven and runs out of fuel while it is being driven it is removed from the garage
* A car may only be refueled when it is in the garage
* The user may select to drive any car from the garage
* The user is the only one that may request that a car be refueled
* The user sets up the drive by entering in the average speed and the driving distance.
* The user drives the car by telling that car to drive.