```
/**
 * The Car class models a car with an odometer, a gas tank, and
 * that gets a given mileage when it drives.
 * @author Richard White
 * @version 2018-10-14
 */

public class Car
{
    // instance variables, to be initialized in constructors
    private double gas;
    private double odometer;
    private double milesPerGallon;

    /**
     * Constructor for objects of class Car
     */
    public Car()
    {
        // initialize instance variables
        gas = 0;
        odometer = 0;
        milesPerGallon = 20;
    }

    /**
     * Overloaded Constructor for objects of class Car
     * @param initialGas amount of gas in the car
     * @param initialOdometer the initial odometer reading
     * @param milesPerGallon the capacity of the car's gas tank
     */
    public Car(double initialGas, double initialOdom, double milesPerGallon)
    {
        // initialize instance variables
        gas = initialGas;
        odometer = initialOdometer;
        this.milesPerGallon = milesPerGallon;
    }

    /**
     * getGas method tells how much gas is left in the car
     * @return the amount of gas in the car's tank
     */
    public double getGas()
    {
        return gas;
    }

    /**
     * addGas method adds an amount of gas to the gas tank
     * @param gasAdded the amount of gas being added to the tank
     */
    public void addGas(double gasAdded)
    {
        gas = gas + gasAdded
    }

    /**
     * drive method drives the car a specified distance
     * @param distance the distance the car is driven
     */
    public void drive(double miles)
    {
        double gasNeeded = miles / milesPerGallon;
        odometer = odometer + miles;
        gas = gas - gasNeeded;
    }

    /**
     * getMiles method tells how many miles the car has traveled
     * @return the total miles the car has traveled ever (odometer reading)
     */
    public double getMiles()
    {
        return this.odometer;
    }
}
```

The Car class

This entire page is a class description, which describes the *constructors* for an object of the class Car, what *accessor* methods can access information about that object, and what *mutator* methods can do to alter the object.

Instance variables

These variables are always declared as private, meaning the user doesn't have direct access to them. Users will only by be able to interact with this information via the methods we write for them.

Constructors

One of these two will be used to construct a new Car when a program requests it. There are two constructors because there are two different ways to make a new Car: one default method, and one in which the initial characteristics of the car are specified.

Accessor methods

These methods are public so that a program can access them. They "return" values to a program that uses them, giving the user access to current information about the Car object.

Mutator methods

These methods are also public, and are used to alter the values of the private instance variables in our Car class.

```
/**
 * CarTester creates several objects of the class Car and tests them.
 *
 * @author Richard White
 * @version 2018-10-14
 */
public class CarTester
{

    public static void main(String[] args)
    {
        // Create two objects of the class Car
        Car myTruck = new Car();

        myTruck.addGas(10);
        System.out.println(myTruck.getGas());
        System.out.println("Expected: 10");
        myTruck.drive(150);
        System.out.println(myTruck.getMiles());
        System.out.println("Expected: 150");
        System.out.println(myTruck.getGas());
        System.out.println("Expected: 2.5");

        // 17 gal in tank, 10,000 miles on odometer, 10 mpg
        Car myHighlander = new Car(17, 10000, 10);

        System.out.println(myHighlander.getMiles());
        System.out.println("Expected: 10000");

    }
}
```

Car *Tester* or *Runner* main method

The CarTester class is the main program, usually in a separate file, that will be used to interact with the Car class that by creating car objects and calling their methods. In this example two Car objects are constructed, and then manipulated using their methods.

Often, we'll "test" the the class by comparing expected values with the actual values in our objects.

"Plumbing"

This line is used at the beginning of your main programs. The details of this syntax will be explained later. For now, just know that you need to include this line at the beginning of your main program or tester.

Calling a constructor

This instruction creates a new object of the class Car, an "instance" of the class with the variable name *myTruck*.

Mutator method call

Here we're calling the *addGas()* method for the *myTruck* object. Because it started out with 0 gallons of gas, we'd expect that it has 10 gallons now.

Accessor method call

Use the *getGas()* method to confirm that we have 10 gallons of gas, and output the results along with what we expected to find.

Calling another constructor

This instruction creates a new a second instance of the Car class, this one with the variable name *myHighlander*.