# Inheritance: The Car subclass

public class Car extends MotorVehicle {  
  
 private int numberWheels = 4;   
 private int numberDoors;  
   
   
 // constructors  
 public Car(String licensePlate, double maxSpeed,  
 String make, String model, int year, int numberOfPassengers,  
 int numberOfDoors) {  
 this(licensePlate, 0.0, maxSpeed, make, model, year, numberOfPassengers,   
 numberOfDoors);   
 }  
  
 public Car(String licensePlate, double speed, double maxSpeed,  
 String make, String model, int year, int numberOfPassengers) {  
 this(licensePlate, speed, maxSpeed, make, model, year,   
 numberOfPassengers, 4);   
 }  
  
 public Car(String licensePlate, double speed, double maxSpeed,  
 String make, String model, int year, int numberOfPassengers,  
 int numberOfDoors) {  
 super(licensePlate, speed, maxSpeed, make, model,  
 year, numberOfPassengers);  
 this.numberDoors = numberOfDoors;  
 }  
   
 public int getNumberOfWheels() {  
 return this.numberWheels;  
 }  
   
 public int getNumberOfDoors() {  
 return this.numberDoors;  
 }  
   
}

It may look like these classes aren't as complete as the earlier ones, but that's incorrect. Car and Motorcycle each inherit the members of their superclass, MotorVehicle. Since a MotorVehicle has a make, a model, a year, a speed, a maximum speed, a number of passengers, cars and motorcycles also have makes, models, years, speeds, maximum speeds, and numbers of passengers. They alse have all the public methods the superclass has. They do not have the same constructors, though they can invoke the superclass constructor through the super keyword, much as a constructor in the same class can be invoked with the this keyword.

[Previous](http://docs.google.com/07.html) | [Next](http://docs.google.com/09.html) | [Top](http://docs.google.com/index.html) | [Cafe au Lait](http://www.cafeaulait.org/)

Copyright 1997-1999, 2002, 2003 Elliotte Rusty Harold

[elharo@metalab.unc.edu](mailto:elharo@metalab.unc.edu)

Last Modified September 30, 2003