# Inheritance

Now suppose you also need a class for a motorcycle. A motorcycle also has a make, a model, a year, a speed, a maximum speed, a weight, a price, a number of passengers it can carry, two wheels and many other properties you can represent with fields. A class that represents a motorcycle might look like this:

public class Motorcycle {  
  
 private String licensePlate; // e.g. "New York A456 324"  
 private double speed; // kilometers per hour  
 private double maxSpeed; // kilometers per hour  
 private String make; // e.g. "Harley-Davidson"  
 private String model; // e.g. "panhead"  
 private int year; // e.g. 1997, 1998, 1999, 2000, 2001, etc.  
 private int numberPassengers; // e.g. 4  
 private int numberWheels = 2; // all motorcycles have two wheels  
   
   
 // constructors  
 public Motorcycle(String licensePlate, double maxSpeed,  
 String make, String model, int year, int numberOfPassengers) {  
  
 this(licensePlate, maxSpeed, make, model, year, numberOfPassengers);  
   
 }  
  
 public Motorcycle(String licensePlate, double speed, double maxSpeed,  
 String make, String model, int year, int numberOfPassengers) {  
  
 this(licensePlate, speed, maxSpeed, make, model, year,   
 numberOfPassengers);  
   
 }  
  
 public Motorcycle(String licensePlate, double speed, double maxSpeed,  
 String make, String model, int year, int numberOfPassengers) {  
  
 // I could add some more constraints like the  
 // number of doors being positive but I won't  
 // so that this example doesn't get too big.  
 this.licensePlate = licensePlate;   
 this.make = make;   
 this.model = model;   
 this.year = year;   
 this.numberPassengers = numberOfPassengers;   
  
 if (maxSpeed >= 0.0) {  
 this.maxSpeed = maxSpeed;  
 }  
 else {  
 maxSpeed = 0.0;  
 }  
   
 if (speed < 0.0) {  
 speed = 0.0;  
 }  
   
 if (speed <= maxSpeed) {  
 this.speed = speed;  
 }  
 else {  
 this.speed = maxSpeed;  
 }  
   
 }  
   
   
 // getter (accessor) methods  
 public String getLicensePlate() {  
 return this.licensePlate;  
 }  
  
 public String getMake() {  
 return this.make;  
 }  
  
 public String getModel() {  
 return this.model;  
 }  
  
 public int getYear() {  
 return this.year;  
 }  
   
 public int getNumberOfPassengers() {  
 return this.numberPassengers;  
 }  
   
 public int getNumberOfPassengers() {  
 return this.numberWheels;  
 }  
   
 public double getMaxSpeed() {  
 return this.speed;  
 }  
  
 public double getSpeed() {  
 return this.maxSpeed;  
 }  
  
 // setter method for the license plate property  
 public void setLicensePlate(String licensePlate) {  
 this.licensePlate = licensePlate;  
 }  
  
 // accelerate to maximum speed  
 // put the pedal to the metal  
 public void floorIt() {  
 this.speed = this.maxSpeed;   
 }  
   
 public void accelerate(double deltaV) {  
  
 this.speed = this.speed + deltaV;  
 if (this.speed > this.maxSpeed) {  
 this.speed = this.maxSpeed;   
 }  
 if (this.speed < 0.0) {  
 this.speed = 0.0;   
 }   
   
 }  
   
}

There's a lot of overlap between the class definitions for Car and Motorcycle. In fact the only things that are different are the constructors and a few of the fields. Inheritance takes advantage of the overlap.

[Previous](http://docs.google.com/04.html) | [Next](http://docs.google.com/06.html) | [Top](http://docs.google.com/index.html) | [Cafe au Lait](http://www.cafeaulait.org/)

Copyright 1997-1999 Elliotte Rusty Harold

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

Last Modified October 11, 1999