# Inheritance

Code reusability is claimed to be a key advantage of object-oriented languages over non-object-oriented languages. Inheritance is the mechanism by which this is achieved. An object can inherit the variables and methods of another object. It can keep those it wants, and replace those it doesn't want.

For example, let us also expand the Car class so that a car also has a make, a model, a year, a number of passengers it can carry, four wheels, either two or four doors. That class might look like this:

public class Car {  
  
 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. "Ford"  
 private String model; // e.g. "Taurus"  
 private int year; // e.g. 1997, 1998, 1999, 2000, 2001, etc.  
 private int numberPassengers; // e.g. 4  
 private int numberWheels = 4; // all cars have four wheels  
 private int numberDoors; // e.g. 4  
   
   
 // 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) {  
  
 // 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;   
 this.numberDoors = numberOfDoors;   
  
 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 getNumberOfWheels() {  
 return this.numberWheels;  
 }  
   
 public int getNumberOfDoors() {  
 return this.numberDoors;  
 }  
   
 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;   
 }   
   
 }  
   
}

Obviously this doesn't exhaust everything there is to say about a car. Which properties you choose to include in your class depends on your application.

[Previous](http://docs.google.com/03.html) | [Next](http://docs.google.com/05.html) | [Top](http://docs.google.com/index.html) | [Cafe au Lait](http://www.cafeaulait.org/)

Copyright 1997, 1998, 2000 Elliotte Rusty Harold

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

Last Modified April 10, 2000