# Using Constructors

The next program uses the constructor to initialize a car rather than setting the fields directly.

class CarTest7 {  
  
 public static void main(String args[]) {  
   
 Car c = new Car("New York A45 636", 123.45);  
   
 System.out.println(c.getLicensePlate() + " is moving at " + c.getSpeed() +   
 " kilometers per hour.");  
  
 for (int i = 0; i < 15; i++) {  
 c.accelerate(10.0);  
 System.out.println(c.getLicensePlate() + " is moving at " + c.getSpeed()   
 + " kilometers per hour.");  
 }  
  
 }  
   
}

You no longer need to know about the fields licensePlate, speed and maxSpeed. All you need to know is how to construct a new car and how to print it.

You may ask whether the setLicensePlate() method is still needed since it's now set in a constructor. The general answer to this question depends on the use to which the Car class is to be put. The specific question is whether a car's license plate may need to be changed after the Car object is created.

Some classes may not change after they're created; or, if they do change, they'll represent a different object. The most common such class is String. You cannot change a string's data. You can only create a new String object. Such objects are called immutable.

[Previous](http://docs.google.com/20.html) | [Next](http://docs.google.com/22.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 1, 1999