# Invoking Methods

class Car {  
  
 String licensePlate = ""; // e.g. "New York 543 A23"  
 double speed = 0.0; // in kilometers per hour  
 double maxSpeed = 120.0; // in kilometers per hour  
   
 // accelerate to maximum speed  
 // put the pedal to the metal  
 void floorIt() {  
 this.speed = this.maxSpeed;   
 }  
   
  
  
}

Outside the Car class, you call the floorIt() method just like you reference fields, using the name of the object you want to accelerate to maximum and the . separator as demonstrated below

class CarTest3 {  
  
 public static void main(String args[]) {  
   
 Car c = new Car();  
   
 c.licensePlate = "New York A45 636";  
 c.maxSpeed = 123.45;  
   
 System.out.println(c.licensePlate + " is moving at " + c.speed +   
 " kilometers per hour.");  
  
 c.floorIt();  
   
 System.out.println(c.licensePlate + " is moving at " + c.speed +   
 " kilometers per hour.");  
  
 }  
   
}

The output is:

New York A45 636 is moving at 0.0 kilometers per hour.  
New York A45 636 is moving at 123.45 kilometers per hour.

The floorIt() method is completely enclosed within the Car class. Every method in a Java program must belong to a class. Unlike C++ programs, Java programs cannot have a method hanging around in global space that does everything you forgot to do inside your classes.

[Previous](http://docs.google.com/08.html) | [Next](http://docs.google.com/10.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