# Passing Arguments to Methods

It's generally considered bad form to access fields directly. Instead it is considered good object oriented practice to access the fields only through methods. This allows you to change the implementation of a class without changing its interface. This also allows you to enforce constraints on the values of the fields.

To do this you need to be able to send information into the Car class. This is done by passing arguments. For example, to allow other objects to change the value of the speed field in a Car object, the Car class could provide an accelerate() method. This method does not allow the car to exceed its maximum speed, or to go slower than 0 kph.

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;   
 }   
   
 }

The first line of the method is called its signature. The signature

void accelerate(double deltaV)

indicates that accelerate() returns no value and takes a single argument, a double which will be referred to as deltaV inside the method.

deltaV is a purely formal argument.

Java passes method arguments by value, not by reference.

[Previous](http://docs.google.com/11.html) | [Next](http://docs.google.com/13.html) | [Top](http://docs.google.com/index.html) | [Cafe au Lait](http://www.cafeaulait.org/)

Copyright 1997, 1998 Elliotte Rusty Harold

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

Last Modified October 16, 1998