# Access Protection

Global variables are a classic cause of bugs in most programming languages. Some unknown function can change the value of a variable when the programmer isn't expecting it to change. This plays all sorts of havoc.

Most OOP languages including Java allow you to protect variables from external modification. This allows you to guarantee that your class remains consistent with what you think it is as long as the methods of the class themselves are bug-free. For example, in the Car class we'd like to make sure that no block of code in some other class is allowed to make the speed greater than the maximum speed. We want a way to make the following illegal:

Car c = new Car("New York A234 567", 100.0);  
 c.speed = 150.0;

This code violates the constraints we've placed on the class. We want to allow the compiler to enforce these constraints.

A class presents a picture of itself to the world. (This picture is sometimes called an interface, but the word  *interface* has a more specific meaning in Java.) This picture says that the class has certain methods and certain fields. Everything else about the class including the detailed workings of the class's methods is hidden. As long as the picture the class shows to the world doesn't change, the programmer can change how the class implements that picture. Among other advantages this allows the programmer to change and improve the algorithms a class uses without worrying that some piece of code depends in unforeseen ways on the details of the algorithm used. This is called encapsulation.

Another way to think about encapsulation is that a class signs a contract with all the other classes in the program. This contract says that a class has methods with unambiguous names which take particular types of arguments and return a particular type of value. The contract may also say that a class has fields with given names and of a given type. However the contract does not specify how the methods are implemented. Furthermore, it does not say that there aren't other private fields and methods which the class may use. A contract guarantees the presence of certain methods and fields. It does not exclude all other methods and fields. This contract is implemented through access protection. Every class, field and method in a Java program is defined as either public, private, protected or unspecified.

You're closer to your immediate family (your parents and your children) than you are to your cousins. You're closer to your cousins than to the general public at large, but there are some things you don't tell anybody. Furthermore, your family is not my family.

[Previous](http://docs.google.com/22.html) | [Next](http://docs.google.com/24.html) | [Top](http://docs.google.com/index.html) | [Cafe au Lait](http://www.cafeaulait.org/)

Copyright 1997, 1998, 2003 Elliotte Rusty Harold

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

Last Modified January 28, 2003