# Member Variables vs. Local Variables

class Car {  
  
 String licensePlate = ""; // member variable  
 double speed; = 0.0; // member variable  
 double maxSpeed; = 120.0; // member variable  
   
 boolean isSpeeding() {  
 double excess; // local variable   
 excess = this.maxSpeed - this.speed;   
 if (excess < 0) return true;  
 else return false;  
 }  
  
}

Until now all the programs you've seen quite simple in structure. Each had exactly one class. This class had a single method, main(), which contained all the program logic and variables. The variables in those classes were all local to the main() method. They could not be accessed by anything outside the main() method. These are called local variables.

This sort of program is the amoeba of Java. Everything the program needs to live is contained inside a single cell. It's quite an efficient arrangement for small organisms, but it breaks down when you want to design something bigger or more complex.

The licensePlate, speed and maxSpeed variables of the Car class, however, belong to a Car object, not to any individual method. They are defined outside of any methods but inside the class and are used in different methods. They are called member variables or fields.

*Member variable*, *instance variable*, and  *field* are different words that mean the same thing.  *Field* is the preferred term in Java. *Member variable* is the preferred term in C++.

A *member* is not the same as a member variable or field. Members include both fields and methods.

[Previous](http://docs.google.com/10.html) | [Next](http://docs.google.com/12.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