Learning object-oriented programming brings a whole slew of concepts and terminology that many ColdFusion developers are unprepared for. When we did the object-oriented and frameworks issue (Fusion Authority Quarterly Update Volume 1, Issue 2), I asked Hal Helms to write a lexicon as an easy reference. This was probably the hardest chapter in the whole book to write, and it caused great contention between members of our staff due to the differing philosophical opinions on various items.

Many of the definitions here are based on Java, the 500-pound gorilla of object-oriented programming (OOP). To apply them to ColdFusion, you may need to go beyond the definitions or limit their scope. This lexicon is meant to give you a basic understanding of the terminology, not to bring you the dogmatic “truth.”

**abstract class**: A superclass that is not meant to be instantiated. Abstract classes can have real methods and properties that will be used by their subclasses. Languages that implement the concept of abstract classes usually provide a mechanism for ensuring that they cannot be instantiated; however, ColdFusion does not provide such a mechanism.

**abstraction**: The process of removing details of something in order to reduce it to a set of essential characteristics. By simplifying the thing being modeled, the programmer reduces the original’s complexity (without sacrificing the original’s essential correctness, we hope). Abstraction is a concept used in many OO techniques, such as abstract classes, base classes, and interfaces.

**aggregation**: A design in which a class holds objects as instance variables. For example, a Department class might hold an array of Employee objects as well as a Manager object (see Figure 19-1). Aggregation often increases the flexibility of a design as it allows for polymorphic variations, which nonobject types do not. You could, for example, provide a subclass of Employee to the Department class with no ill effects (see Liskoff Substitution Principle).
API (application programming interface): Implements the principle of information-hiding by providing a public, well-defined, stable set of methods that expose the intended functionality of a class. APIs can be written on the small scale (e.g., a class’s public methods) and the large scale (e.g., a published API for an entire application composed of many classes).

argument: A value passed to a method.

base class: A class that is extended by other classes; also called a superclass. (See also inheritance.)

class: A type specification, or blueprint, used to instantiate objects. (Contrast with the abstract class, which cannot be used to instantiate objects.) Classes typically have methods and instance variables. ColdFusion refers to classes as components (CFCs).

class variable: A variable shared by all instances of a class. For every class instance, the value of the variable is the same. If one instance changes the value of this variable, the variable's value will be changed for all instances. For example, a class defining a 2010 Mercedes SL-550 might have a carsRegistered variable that tracks the number of these cars registered. All instances of this class will have the same value for this variable.

component: See class.

composition: A stronger form of aggregation in which the class cannot exist without its composite parts. A Triangle class would be composed of three individual Line objects (see Figure 19-2); remove any of these Line objects, and the Triangle object is incoherent. The compositing class often handles the instantiation of composite parts itself.