**Class Members**

* **fields** A field is a member variable used to hold a value. In OOP parlance, fields are sometimes referred to as the object's data. You can apply several modifiers to a field, depending on how you want the field used. These modifiers include *static*, *readonly*, and *const*. We'll get to what these modifiers signify and how to use them shortly.
* **methods** A method is the actual code that acts on the object's data (or fields).
* **properties** Properties are sometimes called *smart fields* because they are actually methods that look like fields to the class's clients. This allows a greater degree of abstraction for the client in that the client doesn't have to know whether it's accessing the field directly or an accessor method is being called.
* **constants** As the name suggests, a constant is a field with a value that cannot be changed. Later in this chapter, I'll discuss constants and compare them to something called *read-only* fields.
* **indexers** As a property is a smart field, an indexer is a smart array—that is, a member that lets an object be indexed through *get* and *set* accessor methods. An indexer enables you to easily index into an object for purposes of setting or retrieving values.
* **events** An event is something that causes some piece of code to run. Events are an integral part of Microsoft Windows programming. An event is fired, for example, when the mouse is moved or a window is clicked or resized
* **operators** C# gives you the ability, via operator overloading, to add the standard mathematical operators to a class so that you can write more intuitive code by using those operators.