

Dealing with Keywords

A *keyword* is a word that has a special meaning defined by the Java programming language. The program shown earlier in Listing 1-1 uses four keywords: `public`, `class`, `static`, and `void`. In all, Java has 53 keywords. They're listed in alphabetical order in Table 1-1.

<code>abstract</code>	<code>default</code>	<code>goto</code>	<code>package</code>	<code>synchronized</code>
<code>assert</code>	<code>do</code>	<code>if</code>	<code>private</code>	<code>this</code>
<code>boolean</code>	<code>double</code>	<code>implements</code>	<code>protected</code>	<code>throw</code>
<code>break</code>	<code>else</code>	<code>import</code>	<code>public</code>	<code>throws</code>
<code>byte</code>	<code>enum</code>	<code>instanceof</code>	<code>return</code>	<code>transient</code>
<code>case</code>	<code>extends</code>	<code>int</code>	<code>short</code>	<code>true</code>
<code>catch</code>	<code>false</code>	<code>interface</code>	<code>static</code>	<code>try</code>
<code>char</code>	<code>final</code>	<code>long</code>	<code>strictfp</code>	<code>void</code>
<code>class</code>	<code>finally</code>	<code>native</code>	<code>super</code>	<code>volatile</code>
<code>const</code>	<code>float</code>	<code>new</code>	<code>switch</code>	<code>while</code>
<code>continue</code>	<code>for</code>	<code>null</code>		



Strangely enough, three keywords listed in Table 1-1 — `true`, `false`, and `null` — aren't technically considered to be keywords. Instead, they're called *literals*. Still, they're reserved for use by the Java language in much the same way that keywords are, so I lumped them in with the keywords.

Stranger still, two keywords — `const` and `goto` — are reserved by Java but don't do anything. Both are carryovers from the C++ programming language. The `const` keyword defines a constant, which is handled in Java by the `final` keyword. As for `goto`, it's a C++ statement that is considered anathema to object-oriented programming purists, so it isn't used in Java. Java reserves it as a keyword solely for the purpose of scolding you if you attempt to use it.



Like everything else in Java, keywords are case-sensitive. Thus, if you type `If` instead of `if` or `For` instead of `for`, the compiler complains about your error. Because Visual Basic keywords begin with capital letters, you'll make this mistake frequently if you have programmed in Visual Basic.

Considering the Java community's disdain for Visual Basic, it's surprising that the error messages generated when you capitalize keywords aren't more insulting. Accidentally capitalizing a keyword in Visual Basic style can