

POINT OF INFORMATION STRICT AND STATIC METHODS

Each and every method that you use in Java is contained within a class in the same manner as a `main()` method. However, within a class, there can be two fundamentally different types of methods.

Methods that must be used with objects are referred to as **strict methods**. Two examples of strict methods are `print()` and `println()`. These methods are part of the `PrintStream` class. Strict methods are generally used with the syntax:

```
objectName.methodName( arguments );
```

An example of this syntax is `System.out.println("Hello World");`. In this example, the object is identified by both a class and object name. The object is the output stream named `out`, which is an object of the `System` class (see Information Box on page 24). Thus, it is fully specified by the full name `System.out`. Although strict methods are restricted to use with objects, they can be accessed from outside of the class they are written in if their header line includes the word `public`.

The second type of method is referred to as a **static method**. A static method is one that does not operate on an object but receives all of its data as arguments. As with a strict method, if a static method's header line includes the word `public`, the method can be called from outside of its own class. In such a case, the method is referred to as a **general-purpose method**. This implies that the method is constructed to perform a general-purpose task that can be useful in a number of places, such as constructing a dialog or computing the square root of a number. An example of such a method is `showMessageDialog()`, which uses its arguments to position and display a dialog box from within any method that uses it.

In using a general-purpose method outside of its class, however, you must indicate where the method is to be found. This is accomplished by listing its class name before the method's name using the syntax

```
ClassName.methodName( arguments );
```

Operationally, the syntax for both strict and general-purpose methods look the same in that both precede the method's name with a period and either an object and/or class name. For now, simply use each method as it is given in the text. In each case, whenever a new method is presented, we will indicate whether it is a static or strict method. After a number of general-purpose mathematical methods are presented in Chapter 3 and you learn to construct your own static methods in Chapter 6 and strict methods in Chapter 9, you will gain a much clearer understanding and appreciation of these two method types.

It should be noted that this statement, as with all Java statements, can be spread over multiple lines. The only requirement in doing so is that the method's name and any enclosed strings are not split between lines. Thus, the preceding statement can be written as

```
JOptionPane.showMessageDialog(null, "Hello World!",  
                             "Sample", JOptionPane.QUESTION_MESSAGE);
```