**Thinking in Java**

**Introduction**

You can’t look at Java as just a collection of features

**Prerequisites**

This book assumes that you have some programming familiarity: You understand that a program is a collection of statements, the idea of a subroutine/function/macro, control statements such as “if” and looping constructs such as “while,” etc. However, you might have learned this in many places, such as programming with a macro language or working with a tool like Perl. As long as you’ve programmed to the point where you feel comfortable with the basic ideas of programming, you’ll be able to work through this book. Of course, the book will be easier for C programmers and more so for C++ programmers, but don’t count yourself out if you’re not experienced with those languages—however, come willing to work hard. Also, the Thinking in C multimedia seminar that you can download from www.MindView.net will bring you up to speed in the fundamentals necessary to learn Java. However, I will be introducing the concepts of object-oriented programming (OOP) and Java’s basic control mechanisms

**Learning Java**

At about the same time that my first book, Using C++ (Osborne/McGraw-Hill, 1989), came out, I began teaching that language. Teaching programming ideas has become my profession; I’ve seen nodding heads, blank faces, and puzzled expressions in audiences all over the world since 1987.

**Goals**

Like my previous book, Thinking in C++, this book was designed with one thing in mind: the way people learn a language. When I think of a chapter in the book, I think in terms of what makes a good lesson during a seminar. Seminar audience feedback helped me understand the difficult parts that needed illumination

**An object has an interface**

Aristotle was probably the first to begin a careful study of the concept of type; he spoke of “the class of fishes and the class of birds.” The idea that all objects, while being unique, are also part of a class of objects that have characteristics and behaviors in common was used directly in the first object-oriented language, Simula-67, with its fundamental keyword class that introduces a new type into a program.

**Errors**

No matter how many tools a writer uses to detect errors, some always creep in and these often leap off the page for a fresh reader. If you discover anything you believe to be an error, please use the link you will find for this book at www.MindView.net to submit the error along with your suggested correction. Your help is appreciated.