
Denver JUG Review: Hardcore Java

A New Book by Robert Simmons, Jr.

Reviewed By Aaron Longwell

Table of Contents

1. Overview	1
2. I Didn't Know Java Could Do That!	1
3. A Collection of Tips on Data Collections	2
4. Thin on Pages, Thick on Content	2

1. Overview

Yes, yes. You don't need another Java book to clutter your shelves. The numerous books on EJB, Java2, UML, Web Services, XML, J2EE, Patterns, Web Design, HTML are already leaving a permanent divot in your floor, and their pages have consumed good portions of this nations forests. Well, forgive me, I have another Java book for you to buy. Good news though, this one is short, lightweight, easy to read, and endlessly valuable.

Hardcore Java, by Robert Simmons, Jr. is unique in the world of Java literature. It spends no time evangelizing some new coding movement, not a page explaining a complex API, no chapters evangelizing... well, anything. Instead, Simmons focuses on the techniques and secrets of the "Java Masters." These are tips, tricks, hints and suggestions. Everything from design to performance is covered in the most practical manner possible. Plenty of code examples complement chapters on data modeling, collections, exception handling, memory references and lesser-used features in the Java syntax.

2. I Didn't Know Java Could Do That!

Hardcore Java opens with two chapters on Java syntax, JDK and compiler features. You've probably used the **final** keyword before, but you can't imagine how the author fills an entire chapter on the importance of using **final** everywhere. He makes a convincing argument that this single keyword can prevent bugs, improve design, and generally take your Java coding to the next level. That's a lot from a little five-letter word, I know. Read the book, you'll understand.

These chapters also cover some features of the Java syntax you may not be familiar with. For example, it explains using a for loop to iterate over an Iterator object. Hint: it involves leaving the 3rd statement of the for loop's syntax blank. This method is more succinct, easier to read, and it locks the iterator and iterated item inside the for loop scope, reducing typo bugs.

In addition to its coverage of syntax, Hardcore Java also dives deep into the internals of Java itself. You'll learn about memory management, the many types of object references, immutable types, constant objects, reflection, and object proxies. These things may seem stuffy and academic, but I can almost guarantee that each could improve your current and future Java projects. Like this reviewer, you probably know these techniques are used extensively by the Java "gurus" who develop low-level libraries, but you doubt they're of any use to you. This book will show you how to leverage these techniques in any Java project, even RDBMS business applications.

Of particular value is the section on Java reflections. Java's reflection abilities are top-notch. If done right, reflection-based coding can save you a lot of development effort and make your code more flexible and powerful. Simmons properly addresses the performance issues inherent in reflection, and

provides good code examples for caching solutions to improve performance. As an aside, this reviewer also recommends taking a look at the Jakarta BeanUtils project for some production-ready reflection code in action. As you might know, this code forms the vital core of the Struts project's ActionForm design.

3. A Collection of Tips on Data Collections

As Robert Simmons points out, no feature of the Java language (or probably any language, for that matter) draws more questions than the use of data collections. The question of the most appropriate collection subclass for a given problem can be puzzling when you begin examining the decision's effects on performance, code reuse, and readability. Hardcore Java covers this topic quite well. It's a quick read, and it provides a good reference on collections best practices. Unfortunately, it's not as in-depth as the rest of the book. It's probably more appropriate as a collection intro for junior-level and intermediate Java developers. In that genre, Bruce Eckel's Thinking in Java is surely superior. If Hardcore Java has a weakness, it's here, in the collections chapter. Simmons' analysis and advice just isn't as plentiful.

What it lacks in data structure code recommendations, Hardcore Java makes up for in data modeling assistance and the design of data-centric applications. There is nothing new in these chapters, but they serve as the best review of fundamentals around. Simmons' walks through requirements gathering, natural language modeling, object relationships, and data constraints. Amidst the "stuff you already know", are nuggets of from-the-trenches advice.

4. Thin on Pages, Thick on Content

Given the wealth of books on Java, it is hard to justify another general Java book on your shelf. Hardcore Java is certainly an exception. You'll likely read this book in a single weekend, and re-read it multiple times. You'll need a notebook to keep track of all the little things that turn Java developers into "Java experts." Happy reading!