Extracted from:

Functional Programming in Java

Harnessing the Power of Java 8 Lambda Expressions

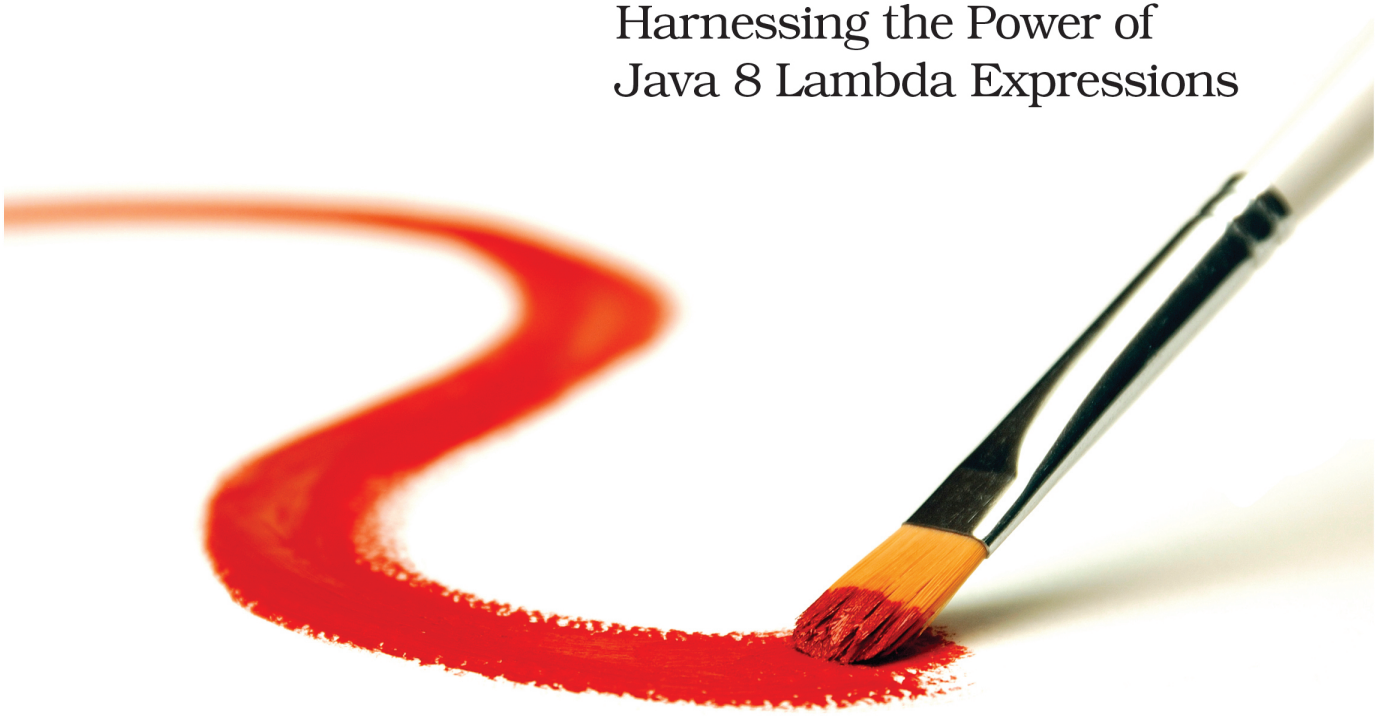
The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina

The Pragmatic Programmers

Functional Programming in Java

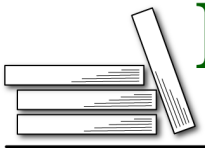
Harnessing the Power of Java 8 Lambda Expressions



Venkat Subramaniam

Foreword by Brian Goetz

Edited by Jacquelyn Carter



Our Pragmatic courses, workshops, and other products can help you and your team create better software and have more fun. For more information, as well as the latest Pragmatic titles, please visit us at *http://pragprog.com*.

Sir Charles Antony Richard Hoare's quote is used by permission of the ACM.
Abelson and Sussman's quote is used under Creative Commons license.

The team that produced this book includes:

Jacquelyn Carter (editor)
Potomac Indexing, LLC (indexer)
Candace Cunningham (copyeditor)
David J Kelly (typesetter)
Janet Furlow (producer)
Ellie Callahan (support)

For international rights, please contact *rights@pragprog.com*.

Printed in the United States of America.
ISBN-13: 978-1-937785-46-8
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—February 2014

To the loving memory of my grandmothers,
Kuppammal and Jayalakshmi. I cherish my
wonder years under your care.

In any field, find the strangest thing and then explore it.

➤ *John Archibald Wheeler*

Preface

You're in for a treat. One of the most prominent and widely used languages in the world has evolved. Until now Java gave us one set of tools—the object-oriented paradigm—and we did the best we could with it. Now there's another, more elegant way to solve the common problems we encounter when developing applications. We can now do quite effectively in Java what was previously possible only on the JVM using other languages—this means more power to Java programmers.

I'm thankful to have had the privilege over the past few decades to program with multiple languages: C, C++, Java, C#, F#, Ruby, Groovy, Scala, Clojure, Erlang, JavaScript... When asked which one's my favorite, my resounding answer has been that it's not the language that excites me, but the way we program.

The science and engineering in programming drew me in, but the art in programming keeps me. Coding has a lot in common with writing—there's more than one way to express our ideas. Java helped us write code using objects. Now we have an additional way to implement our designs and ideas.

This is a new way in Java, one that will make our code more expressive, easier to write, less error prone, and easier to parallelize than has been the case with Java until now. This way has been around for decades and widely used in languages like Lisp, Clojure, Erlang, Smalltalk, Scala, Groovy, and Ruby. It's not only a new way in Java, but a better way.

Since coding is like writing, we can learn a few things from that field. In *On Writing Well [Zin01]*, William Zinsser recommends simplicity, clarity, and brevity. To create better applications, we can start by making the code simpler, clear, and concise. The new style of programming in Java lets us do exactly that, as we will explore throughout this book.

Who's This Book For

This book is for programmers well versed in object-oriented programming in Java and keen to learn and apply the new facilities of lambda expressions. You'll need good experience programming in previous versions of Java, especially Java 5, to make the best use of this book.

Programmers mostly interested in JVM languages like Scala, Groovy, JRuby, and Clojure can benefit from the examples in this book and can relate back to the facilities offered in those languages. They can also use the examples to help fellow Java programmers on their teams.

Programmers experienced with the functional style of programming in other languages and who are now involved in Java projects can use this book, as well. They can learn how what they know translates to the specifics of the lambda expressions' usage in Java.

Programmers who are familiar with lambda expressions in Java can use this book to help coach and train their team members who are getting up to speed in this area.

What's in This Book

This book will help you get up to speed with Java 8 lambda expressions, to think in the elegant style, and to benefit from the additions to the Java Development Kit (JDK) library. We'll take an example-driven approach to exploring the concepts. Rather than discuss the theory of functional programming, we'll dive into specific day-to-day tasks to apply the elegant style. This approach will quickly get these concepts under our belts so we can put them to real use on projects right away.

On the first read, take the time to go over the chapters sequentially as we build upon previously discussed concepts and examples. Each chapter closes with a quick summary to recap what was covered. Later, when working on applications, take a quick glance at any relevant example or section in the book. There's also a syntax appendix for quick reference.

Here's how the rest of the book is organized:

We discuss the functional style of programming, its benefits, and how it differs from the prevalent imperative style in Chapter 1, *Hello, Lambda Expressions!*, on page ?. We also look into how Java supports lambda expressions in this chapter.

The JDK collections have received some special treatment in Java 8, with new interfaces, classes, and methods that support functional-style operations. We will explore these in Chapter 2, *Using Collections*, on page ?.

In Chapter 3, *Strings, Comparators, and Filters*, on page ?, we exploit functional-style and lambda expressions to work with strings, implement the `Comparator` interface, and use filters for file selection.

In addition to using the functional-style facilities in the JDK, we can benefit from applying the elegant style in the design of methods and classes we create. We'll cover functional-style design techniques in Chapter 4, *Designing with Lambda Expressions*, on page ?.

The lambda expressions facilitate a code structure that helps delineate operations to manage object lifetimes and resource cleanup, as we'll discuss in Chapter 5, *Working with Resources*, on page ?.

We'll see lambda expressions shine in Chapter 6, *Being Lazy*, on page ?; they provide us the ability to postpone instance creation and method evaluations as well as create infinite lazy collections, and thereby improve the code's performance.

In Chapter 7, *Optimizing Recursions*, on page ?, we will use lambda expressions to optimize recursions and achieve stellar performance using memoization techniques.

We'll put the techniques we cover in the book to some real use in Chapter 8, *Composing with Lambda Expressions*, on page ?, where we'll transform objects, implement MapReduce, and safely parallelize a program with little effort.

In Chapter 9, *Bringing It All Together*, on page ?, we'll go over the key concepts and the practices needed to adopt those techniques.

In Appendix 1, *Starter Set of Functional Interfaces*, on page ?, we'll take a glance at some of the most popular functional interfaces.

A quick overview of the Java 8 syntax for functional interfaces, lambda expressions, and method/constructor references is in Appendix 2, *Syntax Overview*, on page ?.

The URLs mentioned throughout this book are gathered together for your convenience in Appendix 3, *Web Resources*, on page ?.

Java Version Used in This Book

To run the examples in this book you need Java 8 with support for lambda expressions. Using automated scripts, the examples in this book have been tried out with the following version of Java:

```
java version "1.8.0"
Java(TM) SE Runtime Environment (build 1.8.0-b128)
Java HotSpot(TM) 64-Bit Server VM (build 25.0-b69, mixed mode)
```

Take a few minutes to download the appropriate version of Java for your system. This will help you follow along with the examples in this book.

How to Read the Code Examples

When writing code in Java, we place classes in packages, and executable statements and expressions in methods. To reduce clutter, we'll skip the package names and imports in the code listings. All code in this book belongs to a package:

```
package fpij;
```

Any executable code not listed within a method is part of an undisplayed main() method. When going through the code listings, if you have an urge to look at the full source code, remember it's only a click away at the website for this book.

Online Resources

A number of web resources referenced throughout the book are collected in Appendix 3, *Web Resources*, on page ?. Here are a few that will help you get started with this book:

The Oracle website for downloading the version of Java used in this book is https://jdk8.java.net/download.html. The JDK documentation is available at http://download.java.net/jdk8/docs/api.

This book's page at the Pragmatic Bookshelf website is http://www.pragprog.com/titles/vsjava8. From there you can download all the example source code for the book. You can also provide feedback by submitting errata entries or posting your comments and questions in the forum. If you're reading the book in PDF form, you can click on the link above a code listing to view or download the specific examples.

Now for some fun with lambda expressions…

Venkat Subramaniam

February 2014