

Welcome to the Java Workshop

By Trevor Miller

Table of contents

1 About The Book.....	2
1.1 Introduction.....	2
1.2 An "Open Source Book"?......	2
1.3 Who The Book Is For.....	2
2 Overview.....	3
2.1 Topics Presented In The Book.....	3
2.1.1 Installing The Java Development Kit.....	3
2.1.2 Using The Eclipse Integrated Development Environment.....	3
2.1.3 Creating and Using Java Packages.....	4
2.1.4 Using the Javadoc Tool to Produce an Application Programmer Interface.....	4
2.1.5 Using the Jar Tool to Create Libraries and Executable Java Archive Files.....	4
2.1.6 Using Apache Ant to Simplify the Build Process.....	4
2.1.7 Using JUnit Framework to Perform Unit Testing.....	4
2.1.8 Using Apache Log4J to Perform Logging.....	4
2.1.9 Processing XML with Apache Xerces.....	4
3 Software and Versions Used.....	5
4 Conventions Used.....	5
5 Reach Out.....	5
6 Acknowledgements.....	6

The programs in this book have been included for instructional value, the author does not offer any warranties or representations in respect of their fitness for a particular purpose, nor does the publisher accept any liability for any loss or damage arising from their use.

1. About The Book

1.1. Introduction

The original idea behind the Java Workshop was to introduce some of the practicalities faced when developing software using the Java Platform. As such, the original book focused on developing software using the Java Platform and not so much on the tools or the practical side of things. In revising the text, I decided to shift the focus somewhat to a new direction, the tools that a Java Developer should have in his / her toolbox.

It has often been said that the right tools for the job can make the job a lot easier. This is even more so when one considers the practice of developing software. When one uses the right tools, suddenly the task at hand becomes easier and quicker to accomplish creating a marked improvement in productivity. This book therefore focuses on the practical aspects of developing in Java specifically looking at common tasks and at the tools that make accomplishing these tasks easier and faster.

This is not a book that teaches Java. Instead it is a book that showcases typical tools found in a Java developer's toolbox. If you are looking for a book that teaches Java you should look elsewhere and once you've found such a book, you should couple it with this one, I guarantee the result will be worth it.

1.2. An "Open Source Book"?

The idea behind this is far from original. Credit must be given to Bruce Eckel's **Thinking in Java**, which is freely available on the web and also available in printed format. While I used a similar concept, the Java Workshop differs fundamentally in that it may be redistributed and modified, much like open source software. For those that are a little finicky about details, the license is available here: [License](#) (licence.html) .

1.3. Who The Book Is For

This book is for anyone interested in Java. Whether that interest is purely in an academic sense or truly a passion for the language, I hope that everyone can gain something from this book.

The target audience is specifically students who are learning Java for the first time. I

Welcome to the Java Workshop

remember when I first learnt Java, many of the trivialities were far from trivial and even the simple task of setting an environment variable was no small feat. I hope that if anything, this book will help students to accomplish these and many more trivial tasks and that they will not seem like small miracles.

The goal of this book is to introduce students to the practicalities of the Java language and provide them with enough references to pursue the topic further. In this regard, each topic is presented in a simplistic manner and a topic does not cover all there is, especially those topics that are extremely broad like XML. Rather a brief introduction is given and then a list of suitable references for further information.

2. Overview

Ironically my vision for the first edition was to showcase a number of tools for Java developers, however the book also included a large amount of theoretical content which was contrary to the idea of an introduction to the practicalities of programming with the Java platform; this second edition is much closer to a manifestation of that original vision than the first edition was.

I focused each chapter around a certain piece of software that I have found exceedingly useful, a tool that I consider to be an essential item in each developer's toolbox. This means that each chapter focuses around performing a specific task.

2.1. Topics Presented In The Book

2.1.1. Installing The Java Development Kit

Installing the Java Development Kit (JDK) is not difficult but it can be tricky for people that have never done it before. One of the most important learning experiences novice programmers can have is to install the JDK and set the PATH variable. Not only is it an enriching experience but also the satisfaction gained from accomplishing this task is monumental. I hope that the steps outlined in this theme will guide many to successfully install Java on their computers.

2.1.2. Using The Eclipse Integrated Development Environment

With all the IDEs available, many students may wish to become familiar with one of them and use it to write their programs with. The choice of which IDE to use is not an easy one, yet Eclipse has become the industry standard IDE surpassing many of its competitors. This chapter will look at installing and using this invaluable tool that every Java Developer should have.

2.1.3. Creating and Using Java Packages

Understanding Java Packages is key when one wishes to organize one's source code and this chapter will look at how to create and use Java Packages.

2.1.4. Using the Javadoc Tool to Produce an Application Programmer Interface

This chapter will focus on the importance of comments and the correct way to write Javadoc comments and will show how to use the Javadoc tool to produce an API.

2.1.5. Using the Jar Tool to Create Libraries and Executable Java Archive Files

Java Archives or jar files have become the most popular form of packaging and distributing Java software. It is thus imperative for students to understand what a jar file actually is, how to create one, view the contents and extract the contents as well as package software in an executable jar file.

2.1.6. Using Apache Ant to Simplify the Build Process

Small projects can quite easily be compiled from the command-line. Medium to large projects can become tedious to compile and package and therefore a build tool of some sort can make the process easier. In this regard, Apache Ant is simply the best build tool available. This chapter will discuss Ant and how to use it.

2.1.7. Using JUnit Framework to Perform Unit Testing

Unit testing is perhaps one of the most important aspects of software development. This chapter looks at how unit testing can be performed using the JUnit package.

2.1.8. Using Apache Log4J to Perform Logging

Logging is a very important way of debugging code as it provides a mechanism to see what a program is doing at every step of the way. This chapter looks at logging using the Apache Log4J package.

2.1.9. Processing XML with Apache Xerces

XML is used in a number of places and it is useful to know how to process it. This chapter will look at how to process XML using the Xerces parser as well as perform transformations using Apache Xalan.

3. Software and Versions Used

Keeping up to date with rapidly changing technologies is difficult at best. Below is the list of software and versions used during the writing of this book, by the time you read this there will no doubt be newer versions of some if not all the software listed below.

Tool	URL	Chapter	Description
JDK 1.5.0	java.sun.com (http://java.sun.com)	1, 3-5	The Java 2 Standard Edition SDK
Eclipse 3.0	eclipse.org (http://www.eclipse.org)	2	The Eclipse IDE
Apache Ant 1.6.5	ant.apache.org (http://ant.apache.org)	6	The Apache Ant build tool
JUnit 3.2	junit.sourceforge.net (http://junit.sourceforge.net)	7	JUnit unit testing framework
Apache Log4J 1.2.11	logging.apache.org (http://logging.apache.org)	8	Apache Log4J logging framework
Apache Xerces 2.7.1	xml.apache.org (http://xml.apache.org)	9	Apache Xerces XML parser
Apache Xalan 2.7.0	xml.apache.org (http://xml.apache.org)	9	Apache Xalan XML Transformation tool

4. Conventions Used

Italic is used for emphasis.

Constant width is used for all code fragments, paths and filenames.

Constant width italic is used for general placeholders that indicate some item is replaced for an actual value.

Constant width bold is used for emphasis in a code fragment.

5. Reach Out

I would like your feedback. After you have had a chance to use this book please take a moment to e-mail me with your evaluation. I'd like to know what you liked, what was useful and more importantly what wasn't of use. Although I will read all mails I receive, I cannot guarantee a reply.

tcmiller@users.sourceforge.net

Despite numerous efforts to ensure this book is error free, some errors have most likely slipped through. Even still, some of the material in this book will change or no longer be applicable over time. Please let me know of any errors or omissions in this text.

Trevor C. Miller

Pretoria, South Africa, January 5, 2006

6. Acknowledgements

This book would not have been possible if it were not for the aid and assistance of a number of people. First I would like to thank the people of Sun Microsystems, the Apache Software Foundation, IBM, and of course the Fedora Linux Community. All the software used to write this book came from these organizations that have all produced open source software of such a high standard that it often exceeds that of commercial software. The book was written on Windows using custom software that I wrote to produce the XML content. This was then transformed into static HTML using Apache Forrest. The book was also tested on Linux using Fedora Core.

I wish to thank all those wonderful people that I've had the pleasure tutoring and teaching, whether it was Java, C, C++ or Delphi. These people truly inspired me by persevering, and in those times they even challenged me by asking awkward questions that I at times didn't have an answer for. While I was teaching programming to these people, they in return taught me much; about teaching, about people and most importantly, about myself. To them I am ever grateful.

Lastly, writing this book wouldn't have been possible if it weren't for the music of Leftfield, Michelle Branch and 30 Seconds To Mars.