

Cambridge University Press

0521821134 - JavaTech: An Introduction to Scientific and Technical Computing with Java

Clark S. Lindsey, Johnny S. Tolliver and Thomas Lindblad

Frontmatter

[More information](#)

JavaTech

An Introduction to Scientific and Technical Computing with Java

JavaTech is a practical introduction to the Java programming language with an emphasis on the features that benefit technical computing, such as platform independence, extensive graphics capabilities, multi-threading, and tools to develop network and distributed computing software and embedded processor applications.

The book is divided into three parts. The first presents the basics of object-oriented programming in Java and then examines topics such as graphical interfaces, thread processes, I/O, and image processing. The second part begins with a review of network programming and develops Web client-server examples for tasks such as monitoring of remote devices. The focus then shifts to distributed computing with RMI, which allows programs on different platforms to exchange objects and call each other's methods. CORBA is also discussed and a survey of web services is presented. The final part examines how Java programs can access the local platform and interact with hardware. Topics include combining native code with Java, communication via serial lines, and programming embedded processors.

JavaTech demonstrates the ease with which Java can be used to create powerful network applications and distributed computing applications. It can be used as a textbook for introductory or intermediate level programming courses, and for more advanced students and researchers who need to learn Java for a particular task. *JavaTech* is up to date with Java 5.0.

CLARK S. LINDSEY received his Ph.D. in physics from the University of California at Riverside and has held research positions at Iowa State University, Fermilab, and the Royal Institute of Technology, Sweden. This book grew out of a course in Java programming he developed with Professor Lindblad. He now runs his own company that develops Java applications, Web publications, and educational tools and materials.

JOHNNY S. TOLLIVER holds a Ph.D. in Computational Plasma Physics and has worked in fusion energy research, computer security, and trusted operating systems. He is a Sun Certified Java Programmer and has been actively using Java since 1997. He is currently at Oak Ridge National Laboratory, developing Web services software and a GPS vehicle tracking application using GPS-enabled wireless phone handsets and other GPS devices.

THOMAS LINDBLAD received his Ph.D. in physics at the University of Stockholm in 1972 and became associate professor two years later. He is currently a professor in the Department of Physics at the Royal Institute of Technology, Stockholm, and also serves part time as Director of Undergraduate Studies. His research currently concentrates on techniques in image and data analysis in high data rate systems.

Cambridge University Press

0521821134 - JavaTech: An Introduction to Scientific and Technical Computing with Java

Clark S. Lindsey, Johnny S. Tolliver and Thomas Lindblad

Frontmatter

[More information](#)

JavaTech

*An Introduction to Scientific and Technical
Computing with Java*

**Clark S. Lindsey, Johnny S. Tolliver
and Thomas Lindblad**

Cambridge University Press

0521821134 - JavaTech: An Introduction to Scientific and Technical Computing with Java

Clark S. Lindsey, Johnny S. Tolliver and Thomas Lindblad

Frontmatter

[More information](#)

CAMBRIDGE UNIVERSITY PRESS

Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, São Paulo

Cambridge University Press

The Edinburgh Building, Cambridge CB2 2RU, UK

Published in the United States of America by Cambridge University Press, New York

www.cambridge.org

Information on this title: www.cambridge.org/9780521821131

© C. S. Lindsey, J. S. Tolliver and T. Lindblad 2005

This publication is in copyright. Subject to statutory exception and to the provisions of relevant collective licensing agreements, no reproduction of any part may take place without the written permission of Cambridge University Press.

First published 2005

Printed in the United Kingdom at the University Press, Cambridge

A catalog record for this publication is available from the British Library

Library of Congress Cataloging in Publication data

ISBN-13 978-0-521-82113-1 hardback

ISBN-10 0-521-82113-4 hardback

Cambridge University Press has no responsibility for the persistence or accuracy of URLs for external or third-party internet websites referred to in this publication, and does not guarantee that any content on such websites is, or will remain, accurate or appropriate.

Cambridge University Press

0521821134 - JavaTech: An Introduction to Scientific and Technical Computing with Java

Clark S. Lindsey, Johnny S. Tolliver and Thomas Lindblad

Frontmatter

[More information](#)

Contents

Preface	<i>page</i> xiii
Acknowledgements	xix
Part I Introduction to Java	
1 Introduction	3
1.1 What is Java?	3
1.2 History of Java	4
1.3 Versions of Java	5
1.4 Java – open or closed?	8
1.5 Java features and benefits	8
1.6 Real-world Java applications in science and engineering	12
1.7 The Java programming procedure	14
1.8 Getting started	17
1.9 Changes in Java 2 Standard Edition 5.0	23
1.10 Web Course materials	27
References	27
Resources	28
2 Language basics	29
2.1 Introduction	29
2.2 Language elements and structures	29
2.3 A simple application	31
2.4 Comments	32
2.5 Data types and Java primitives	33
2.6 Strings	35
2.7 Expressions	35
2.8 Operators	36
2.9 Statements	39
2.10 Casts and mixing	44
2.11 Floating-point	47
2.12 Programming	50
2.13 Basic math in Java	53
2.14 Web Course materials	55
Resources	55

3	Classes and objects in Java	57
3.1	Introduction	57
3.2	Custom data types	57
3.3	Class definition	58
3.4	Class instantiation	65
3.5	Static (or class) members	67
3.6	More about primitive and reference variables	69
3.7	Wrappers	73
3.8	Arrays	79
3.9	Exceptions	80
3.10	OOP in engineering and science	84
3.11	Web Course materials	90
	Resources	90
4	More about objects in Java	91
4.1	Introduction	91
4.2	Class inheritance	91
4.3	More about constructors	100
4.4	Abstract methods and classes	105
4.5	Interfaces	109
4.6	More about classes	115
4.7	More about arrays	120
4.8	Improved complex number class	125
4.9	Random number generation	126
4.10	Improved histogram class	128
4.11	Understanding OOP	130
4.12	Web Course materials	130
	References	131
	Resources	131
5	Organizing Java files and other practicalities	132
5.1	Introduction	132
5.2	Class definition files	132
5.3	Packages	133
5.4	The <code>final</code> modifier and constants	140
5.5	Static import in J2SE 5.0	141
5.6	JAR files	143
5.7	Distributing Java code	146
5.8	Applet directories	148
5.9	Javadoc	149
5.10	Coding conventions	150
5.11	Formatting numbers	153

Cambridge University Press

0521821134 - JavaTech: An Introduction to Scientific and Technical Computing with Java

Clark S. Lindsey, Johnny S. Tolliver and Thomas Lindblad

Frontmatter

[More information](#)

5.12 Web Course materials	158
References	159
Resources	159
6 Java graphics	160
6.1 Introduction	160
6.2 AWT	161
6.3 Swing: lightweight beats heavyweight	162
6.4 Swing class hierarchy	163
6.5 Containers	164
6.6 Drawing	174
6.7 Drawing with the <code>Graphics</code> class	178
6.8 Drawing in the Java 2D API	183
6.9 Images	190
6.10 Java and tech graphics	192
6.11 Histogram graphics	192
6.12 Web Course materials	199
References	200
7 Graphical User Interfaces	201
7.1 Introduction	201
7.2 Events	201
7.3 More user interface components	212
7.4 Layout managers	223
7.5 Convenience classes	237
7.6 Frames and menus	242
7.7 User interface with histogram display	247
7.8 Web Course materials	251
References	252
8 Threads	253
8.1 Introduction	253
8.2 Introduction to threads	253
8.3 Stopping threads	258
8.4 Multiprocessing issues	259
8.5 Using multiple threads	262
8.6 Animations	276
8.7 Timers	281
8.8 Concurrency utilities in J2SE 5.0	285
8.9 Web Course materials	285
References	286

Cambridge University Press

0521821134 - JavaTech: An Introduction to Scientific and Technical Computing with Java

Clark S. Lindsey, Johnny S. Tolliver and Thomas Lindblad

Frontmatter

[More information](#)

viii

Contents

9 Java input/output	287
9.1 Introduction	287
9.2 Streams	289
9.3 Stream wrappers	290
9.4 Console I/O	291
9.5 The <code>File</code> class	299
9.6 File I/O	301
9.7 Character encoding	312
9.8 Object I/O	313
9.9 Choosing a stream class	315
9.10 Primitive types to bytes and back	315
9.11 Sources, destinations, and filters	317
9.12 The <code>JFileChooser</code> dialog	318
9.13 Histogram I/O	320
9.14 More Java I/O	325
9.15 Web Course materials	326
References	326
10 Java utilities	327
10.1 Introduction	327
10.2 The <code>java.util</code> package	327
10.3 <code>Vector</code> and <code>Enumeration</code>	328
10.4 <code>Hashtable</code> , <code>Properties</code> , and <code>HashMap</code>	329
10.5 Preferences	331
10.6 The Collections Framework	335
10.7 Generics in J2SE 5.0	338
10.8 Concurrency utilities in J2SE 5.0	341
10.9 Enumerated types in J2SE 5.0	343
10.10 The <code>Arrays</code> class	345
10.11 Tools for strings	349
10.12 <code>Calendar</code> , <code>Date</code> , and <code>Time</code>	353
10.13 Arbitrary precision numbers	356
10.14 Bit handling	360
10.15 Other utilities	362
10.16 Web Course materials	363
References	363
11 Image handling and processing	365
11.1 Introduction	365
11.2 The <code>Image</code> and <code>BufferedImage</code> classes	365
11.3 Image loading	367
11.4 Image display	371
11.5 Creating images	372
11.6 Saving images	373

Cambridge University Press

0521821134 - JavaTech: An Introduction to Scientific and Technical Computing with Java

Clark S. Lindsey, Johnny S. Tolliver and Thomas Lindblad

Frontmatter

[More information](#)

11.7	Image processing	373
11.8	Pixel handling	374
11.9	Filtering	380
11.10	Web Course materials	389
	References	389
	Resource	389
12	More techniques and tips	390
12.1	Introduction	390
12.2	Printing	390
12.3	Cursor icons	392
12.4	Mouse buttons	394
12.5	Popup menu	396
12.6	Handling keystrokes	399
12.7	Audio	402
12.8	Performance and timing	404
12.9	Lifelong Java learning	409
12.10	Web Course materials	410
	References	410
Part II Java and the network		
13	Java networking basics	413
13.1	Introduction	413
13.2	Internet basics	413
13.3	Ports	415
13.4	Java networking	416
13.5	The URL class	417
13.6	InetAddress	423
13.7	Sockets	426
13.8	The client/server model	428
13.9	Web Course materials	430
	References	430
	Resources	430
14	A Java web server	431
14.1	Introduction	431
14.2	Designing a web server	432
14.3	Hypertext Transfer Protocol (HTTP)	435
14.4	Running the server	438
14.5	A more secure server	439
14.6	A client application	443
14.7	Server applications	445
14.8	Servers, servlets and JSP	446

14.9 Web Course materials	447
References	447
15 Client/server with sockets	448
15.1 Introduction	448
15.2 The client/server design	448
15.3 The client/server interaction	449
15.4 The <code>DataServer</code>	450
15.5 The <code>DataWorker</code>	453
15.6 The <code>DataClient</code>	459
15.7 The <code>DataClientWorker</code>	464
15.8 Benefits and shortcomings of sockets	469
15.9 Web Course materials	469
References	470
16 Distributed computing	471
16.1 Introduction	471
16.2 Distributed computing for scientific applications	471
16.3 Minimalist UML	473
16.4 A conceptual model for a simple distributed application	475
16.5 Collaboration diagram for a simple distributed application	477
16.6 Server details	481
16.7 Web Course materials	490
References	491
17 Distributed computing – the client	492
17.1 Introduction	492
17.2 Multithreaded client	492
17.3 Model-View-Controller for the client	493
17.4 More client details	496
17.5 Improved client class diagram	498
17.6 Web Course materials	500
Resources	500
18 Java Remote Method Invocation (RMI)	501
18.1 Introduction	501
18.2 How distributed computing works	501
18.3 RMI overview	503
18.4 The RMI client	509
18.5 RMI security issues	511
18.6 Finally, a working example	515
18.7 How to run on two machines	527
18.8 Conclusion	528

Cambridge University Press

0521821134 - JavaTech: An Introduction to Scientific and Technical Computing with Java

Clark S. Lindsey, Johnny S. Tolliver and Thomas Lindblad

Frontmatter

[More information](#)

18.9 Web Course materials	529
References	529
19 CORBA	530
19.1 Introduction	530
19.2 CORBA IDL	531
19.3 Compiling the IDL file	535
19.4 Creating the server implementation	537
19.5 Client implementation	544
19.6 Running the example	545
19.7 Running the CORBA example on two machines	547
19.8 Conclusion	548
19.9 Web Course materials	548
Resources	548
20 Distributed computing – putting it all together	549
20.1 Introduction	549
20.2 The sample application	549
20.3 Server interfaces	551
20.4 Server factory implementation	553
20.5 Server implementation	556
20.6 Client implementation	568
20.7 Enhanced client using the histogram class	569
20.8 Conclusion	571
20.9 Web Course materials	572
References	572
21 Introduction to web services and XML	573
21.1 Introduction	573
21.2 Introducing web services for distributed computing	573
21.3 XML	574
21.4 Java web services	578
21.5 Other web services technologies	580
21.6 Conclusion	581
21.7 Web Course materials	581
References	581
Part III Out of the sandbox	
22 The Java Native Interface (JNI)	585
22.1 Introduction	585
22.2 What is JNI?	586
22.3 Hello World in JNI	587
22.4 Deeper into JNI	597

22.5	Java <code>String</code> objects	600
22.6	Java primitive arrays	604
22.7	Java object arrays and multidimensional primitive arrays	606
22.8	Java objects on the C side	607
22.9	Calling Java methods from native code	615
22.10	Exceptions in JNI	618
22.11	Local and global references	622
22.12	Threads and synchronization in JNI	623
22.13	Conclusion	624
22.14	Web Course materials	624
	References	624
23	Accessing the platform	625
23.1	Escaping the sandbox	625
23.2	Accessing system properties	625
23.3	Running external programs	629
23.4	Port communications	631
23.5	Web Course materials	650
	References	650
	Resources	650
24	Embedded Java	651
24.1	Introduction	651
24.2	Embedded Java for science and engineering	652
24.3	J2ME – Java 2 Micro Edition	652
24.4	Real-time Java	654
24.5	Java real machines	657
24.6	Benefits of hardware processors	658
24.7	Java processors	658
24.8	Java boards	661
24.9	Programming the Javelin Stamp	665
24.10	An embedded web server	671
24.11	Java processor performance	680
24.12	Web Course materials	680
	References	680
	Appendix 1: Language elements	683
	Appendix 2: Operators	685
	Appendix 3: Java floating-point	693
	Index	697

Preface

Java is a serious language suitable for demanding applications in science and engineering. Really, we promise! Java offers a lot more than just those little applets in your Web browser.

In *JavaTech* we focus on how Java can perform useful tasks in technical computing. These tasks might involve an animated simulation to demonstrate a scientific principle, a graphical user interface for an existing C or C++ computational engine, a distributed computing project, controlling and monitoring an experiment remotely via the Internet, or programming an embedded Java hardware processor in a device such as a remote sensor. While other Java books intended for the science and engineering audience concentrate primarily on numerical programming, we take a much broader approach and examine ways that Java can benefit programmers working on many different types of technical applications.

This project grew out of a course given by two of us (C.S.L. and Th.L.) at the Royal Institute of Technology in Stockholm, Sweden in which students of diverse backgrounds followed the class via the Internet. For this type of *distance learning* situation, we developed hypertext instructional material for delivery via the Web browser that allows for a high degree of self-study. This approach works especially well with Java since many of the demonstration programs run as applets within the browser.

This book provides a handy print companion to this hypertext course, which is available online at www.javatechbook.com. The book includes additional material that deals with distributed computing techniques based on work done by one of us (J.T.) at Oak Ridge National Laboratory in the USA. Throughout the book we refer to the hypertext materials as the *Web Course*.

Who should use this book

JavaTech targets primarily those who want to learn the Java programming language so as to apply it to practical applications in science and engineering. From the freshman science major to the experienced programmer in a technical field, we believe this book and the Web Course will be helpful.

For those unfamiliar with the language and with object-oriented programming, we begin with a compact introduction to Java. Since Java has grown into a very big field we only touch on the essential elements needed to begin doing useful

programming. We include examples of how Java can apply to technical tasks such as histogramming of data and image analysis.

While familiarity with C and C++ will hasten a reader's understanding of Java programming, we do not assume the reader knows these languages.

After the Java introduction we discuss network programming, which we consider to be one of Java's strongest features. We focus particularly on how to build client/server systems for distributed computing applications. If you have a network application, such as the need to monitor remote devices or to give distant users access to a complex simulation running on a central server, the survey here should help you get started. Our aim is to show that you can create powerful network software with Java without needing first to become an authority on all the arcane intricacies of network systems. Java's networking tools and platform portability allow you to focus more on your application than on the underlying mechanisms.

The final part of the book looks at how Java can interact with the local platform, with code in other languages, and with embedded processors. For example, perhaps you have a legacy program in C that represents many years of development and tuning, but it lacks a graphical interface to make it interactive and flexible. We discuss the Java Native Interface (JNI) that allows you to connect your program to Java and to take advantage of the extensive graphical tools available in Java to build an interface around your computational engine in C (or in Fortran via intermediate C code as discussed in the Web Course). You can also add the networking capabilities of Java discussed above. For example, remote clients could connect with your legacy program that runs on a central server.

Hardware microprocessors designed especially to run Java are now widely available. Those who work on embedded processor applications will be interested in our survey of the field of Java processors. In a demonstration program, for example, we show how to connect via a serial port to a microcontroller that is programmed with Java and used to read a sensor.

We look at compact, low-cost platforms that contain Java processors, Ethernet connectors, analog-to-digital inputs, digital-to-analog outputs and other useful features. With such systems you can run servers that allow remote clients to monitor, control, and diagnose an instrument of some kind. This offers the opportunity to those who work with large complex installations, such as an elaborate scientific apparatus or a power plant, to access and control a system at a fine-grained level. We provide a demonstration of a server on such a Java processor platform in which the server responds to a Web browser with an HTML file containing a voltage reading.

Organization and topics

We attempted with this book and Web Course combination to create an innovative and highly flexible approach that allows readers with a diverse range of interests

Cambridge University Press

0521821134 - JavaTech: An Introduction to Scientific and Technical Computing with Java

Clark S. Lindsey, Johnny S. Tolliver and Thomas Lindblad

Frontmatter

[More information](#)

and backgrounds to find and use effectively the materials for their particular needs. The Web Course includes hypertext tutorial materials, many demonstration programs, and exercises. The book compliments the Web Course with more extensive discussions on a range of topics and with tables and diagrams for quick reference.

We follow an example-based teaching approach, using lots of applets and application programs to demonstrate the concepts and techniques described. In addition, we supply a large selection of *starter* programs that provide templates with which readers can quickly begin to develop their own programs.

The chapters in the book correspond directly to those in the Web Course. Note that while one of Java's strongest features is its extensive graphics capability, we do not discuss graphics programming in the first five chapters. Instead we focus on the components and structure of the language. We demonstrate techniques with stand-alone programs (referred to in Java as *applications*) that print to the console and applets that send output to the web browser's Java console window.

The book and Web Course are divided into three parts plus appendices.

Part I Introduction to Java

The 12 chapters in Part I provide an introduction to the Java language. These chapters focus on the Java language but also discuss various topics relevant to applying Java to technical areas. The Web Course expands the introductory material into three tracks:

The *Java Track* provides an introduction to Java programming. The reader can follow this track alone for a quick course in the basics of Java programming. *Supplements* provide additional information on both basic and advanced topics.

The *Tech Track* focuses on topics relevant to general math, science, and engineering applications of Java such as floating-point numbers, random number generators, and image processing.

The *Physics Track* provides an example of how to apply Java to a particular technical subject. The track corresponds to a short course for undergraduate students on the use of numerical computing, simulations, and data analysis in experimental physics.

Part II Java and the network

This part focuses on the application of Java to network programming and distributed computing. It begins with an introduction to TCP/IP programming and then looks at several topics including socket based client/server demonstration programs and distributed computing with RMI, CORBA, and other techniques. An introduction is given to Unified Modeling Language (UML), which leads to better object oriented code design and analysis. A brief overview of web services and XML is also provided.

Part III Out of the sandbox

This part deals with how Java programs can access information and resources on the underlying platforms on which the Java Virtual Machine (JVM) is installed and how the JVM can interact with its local environment. It also reviews implementations of Java in hardware rather than in a virtual machine. Topics include interfacing Java programs to C/C++ and Fortran codes with the Java Native Interface (JNI), communicating with devices via serial/parallel ports, and working with embedded Java processors.

Appendices

Appendices 1 and 2 provide tables of Java language elements and operators, respectively. Appendix 3 gives additional information about floating-point numbers in Java.

Topics not discussed

Java has grown into an enormous industry since it first appeared in the mid-1990s. No single book could possibly do justice to all of the Java classes, packages, tools, techniques, and applications of the language. In fact, there exist many books devoted to individual topics such as Java I/O, graphics, and multithreading. The Java industry expands further every day.

For this book we have chosen what we consider to be an important subset of Java topics relevant to technical applications. Some important topics not treated include:

- Java Enterprise techniques, such as database access and Java application servers
- Security topics such as the Java Cryptography Extension (JCE)
- Java 3D graphics

We do provide in the Web Course a large set of links to references and resources for these and other Java subjects. We also believe that this book provides the reader with a solid base of understanding on which to pursue further learning. All Java programmers must deal with the need to continually learn new classes and APIs (Application Program Interfaces). As we go to press, Sun is about to release Java 2 Standard Edition version 5.0, which contains significant additions to the language. We discuss the most important of these but some are beyond the scope of this book.

We emphasize the use of the web for access to language specifications, online tutorials, and other resources needed to tackle new Java techniques. We include references and web links in each chapter and in the Web Course. You can also find many online resources at <http://java.sun.com>, java.net, and www.ibm.com/developerworks/java/.

As mentioned in the introduction, we do not delve into numerical programming with Java. We only touch on this subject here while the Web Course *Tech* and

Physics tracks contain several introductory level sections. See the reference list at the end of Chapter 1 for a list of several books that deal extensively with numerical programming in Java.

How to use this book and Web Course

We designed the book and Web Course in a way that lets readers follow individualized paths through the materials. Part I, in particular, allows for a variety of different approaches. You could, for example, study only the Java sections of each chapter and get a fast introduction to the basics of Java programming. You could also study the sections with particular relevance to technical applications (the Web Course expands on these in its *Tech Track*) or, alternatively, you could skip these tech topics in a first pass and return to them later. Those already familiar with Java basics could focus just on the tech-related topics.

You can proceed through the book and Web Course at your own pace and experiment with the many applets and application demonstration programs. There is an emphasis on coding by the reader since ultimately you can only learn Java or any other language by writing lots of programs yourself.

Part II and Part III deal with specialized topics. If you are already familiar with the basics of Java programming, you could proceed directly to the chapter or sub-section of interest in those parts.

One of the most important features of Java is its extensive network programming capability. So we designed the course around the assumption that the reader has easy access to the Internet. Most of the Web Course pages include links to reference and resource materials, especially the tutorials and language specifications on the <http://java.sun.com> website. Rather than reinvent the wheel we try to incorporate resources such as the Sun tutorials in a way that takes best advantage of what is already available.

The Web Course hypertext materials and demonstration codes, along with updates and corrections to the book, are available at the website www.JavaTechBook.com. (A mirror site is available at www.particle.kth.se/~lindsey/JavaCourse/Book/.)

Note that if we included in the book the source codes for all the demonstration programs, it would be a very long book indeed. Since the source codes are easily available from the Web Course, we often print only “code snippets” rather than entire classes or programs.

Conventions

Fixed width style indicates:

- code samples such as: `for (i=0; i < 4; i++) j++;`
- Java class names, variable names, and other code-related terms
- console commands such as: `c:\> java HelloWorld`
- web addresses such as `http://java.sun.com`

Cambridge University Press

0521821134 - JavaTech: An Introduction to Scientific and Technical Computing with Java

Clark S. Lindsey, Johnny S. Tolliver and Thomas Lindblad

Frontmatter

[More information](#)

In code listings, italicized *fix width* indicates that the text is not actually in the code but included to emphasize some aspect of the code or to summarize code that was skipped. We also put the class name in **bold** in the code listings. (Coding style conventions are discussed in Section 5.9.) When discussing a method in the text we may often ignore the argument list for the sake of brevity. So `aMethod(int x, float y, double z)` is abbreviated as `aMethod()`.

In the main text, new terms of particular importance are italicized. The book name and Web Course sections are also italicized.

In Chapter 22 on the Java Native Interface, we use the notation `Xxx` and `xxx` as placeholders to represent the many possible names that can replace the `Xxx` or `xxx`. For example, JNI has a `GetIntField()` method. It also has `GetFloatField()`, `GetDoubleField()`, etc. methods. We refer to these as a group with the `GetXxxField()` notation. Similarly, the `xxx` in `jxxxArray` can be replaced with `int`, `float`, `double`, etc. to produce `jintArray`, `jfloatArray`, `jdoubleArray`, etc.

Java version

The code in *JavaTech* primarily follows that of Java version 1.4 released in 2002, but we discuss the significant enhancements available in the Java 5.0 release where relevant. (This release was under development for at least two years and became available in beta form near the end of the writing of this book.) Since many web browsers currently in use only run Java 1.1 applets and also since some small platforms (e.g. embedded processors) with limited resources only run Java 1.1, we also include in the Web Course some discussion of programming techniques for this version and provide sample codes.

The programs do not usually assume a particular platform and should run on MS Windows, Mac OS X, Linux, as well as Solaris and most Unix platforms.

Acknowledgements

We would like to thank our editors Simon Capelin and Vince Higgs for their help and patience. We thank Roger Sundman and Carl Wedlin of Imsys Technologies for their review of the Java hardware discussion and helpful suggestions. Thanks also to Michele Cianciulli for his comments on the manuscript.

We thank the many students who took our Web Course over the years and gave us a great amount of useful feedback. We especially want to thank one of our first students, Conny Carlberg, who encouraged us by quickly applying Java to his research. Several students at the Royal Institute of Technology (KTH) have used parts of the manuscript of this book, and many of them have come with interesting and useful comments, especially Bruno Janvier and Jaakko Pajunen. At a very early stage, when the Java course was introduced, and when Java was not generally too well known, we received encouraging support from many professors at KTH, the University of Stockholm, and the Manne Siegbahn Institute of Physics.

One of us (C. S. L.) would like to dedicate this book to his wife Kerima who provided great support and encouragement.

One of us (J. S. T.) would like to thank his wife Janey and children Kevin and Chelsea for their enduring patience with a too-often absent or preoccupied husband and father during many months on a project that grew to be longer and more difficult than anyone expected. Thank you.

One of us (Th. L.) makes a dedication to whoever said “do not write any more books, it is a much bigger undertaking than you recall from writing the previous one.”