

David Parsons

Foundational Java

Key Elements and Practical Programming

David Parsons

Foundational Java

Key Elements and Practical
Programming



David Parsons
Inst. of Information and Math. Sciences
Massey University
17 State Highway
Albany Campus
Albany, Auckland
New Zealand

ISBN 978-1-4471-2478-8 e-ISBN 978-1-4471-2479-5
DOI 10.1007/978-1-4471-2479-5
Springer London Dordrecht Heidelberg New York

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2012931443

© Springer-Verlag London Limited 2012

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licenses issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.

The use of registered names, trademarks, etc., in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

*To my mother, Angela (1931–2011), always
there for me*

Preface

In the Spring of 1996, I attended the Object Technology conference at Christ Church College, Oxford. The excited buzz around the “new” Java language at this conference got me curious, and a few months spent getting up to speed with the syntax convinced me that it was worth trying out as a teaching language. After overcoming some hurdles, including getting a teaching lab upgraded from Windows 3.1 to Windows 95 so we could actually run Java, I taught a class of student volunteers at what is now Southampton Solent University the basics of Java in 1997. From this experience I wrote the first edition of *Introductory Java* which was published by Letts Educational in 1998. In 1999, I left academia to spend some years in industry, training and consulting in Java. This gave me a whole new perspective on the language as a professional tool. The second edition of *Introductory Java* was published in 2003, but another update was long overdue.

This book was originally intended to be the 3rd Edition of *Introductory Java*. However, such were the changes that had taken place over the years, both in Java and in my own experiences of it, that this became a whole new book, with a new title and focus. It has grown out of 15 years of teaching Java both to students and to professionals across the world, and reflects a wealth of experience and insight. I have had a great journey with Java, and I am grateful for the opportunity to share my Java story with readers of this book.

Overview and Goals

Foundational Java: Key Elements and Practical Programming guides the reader through all the core features of Java, and some more recent innovations, in a way that builds skills and confidence through tried and trusted stages, supported by exercises that reinforce the key learning points. Using this book, the reader is introduced to all the most useful and commonly applied Java syntax and libraries, and is provided with many example programs that can provide the basis for more substantial applications.

Integral to the book is the use of the Eclipse Integrated Development Environment (IDE) and the JUnit testing framework. This ensures maximum productivity and code quality when learning Java. However, the fundamentals of using the Java compiler and run time are also explained to ensure that skills are not confined to one environment. In addition, coverage of the Ant tool ensures that the reader is equipped

to automatically build, test, and deploy their applications, including simple web applications, independent of an IDE.

Organization and Features

The book is organized into 20 chapters that cover various levels of Java (see “Suggested Uses” for further information.) Each chapter covers a discrete topic and includes scaffolded exercises that build skills in a step by step fashion. Earlier versions of the book used an “objects first” approach; however experience has shown me that it is better to cover the language fundamentals before addressing object oriented concepts, so this book follows an “objects later” philosophy.

The key features of this book are that it

- Meets the needs of both students and professionals
- Provides both introductory and intermediate coverage
- Is completely up to date, including Java 7
- Makes unit testing one of its key themes, introducing the JUnit 4 testing framework to emphasize the importance of unit testing in modern software development
- Uses the Eclipse IDE, the most popular open source Java IDE, but also explains how Java can be run from the command line
- Includes coverage of the Ant build tool
- Comes with code examples and exercises throughout
- Is accompanied by a full set of PowerPoint presentation slides that have been road tested with classes
- Builds on two previous editions and a set of classroom training materials that have been refined and developed as Java has continued to evolve
- Includes some important illustrations in color

Target Audiences

This is primarily an undergraduate textbook. It can be used for basic introductory courses or for intermediate classes. From that perspective, it has been structured as a teaching text that breaks into weekly topics that build upon one another. It is also a book suitable for professional software developers who need to pick up Java from previous experience in other tools or languages. The materials have been tried and tested in commercial training courses for professional software developers over the last 7 years. The choice of intermediate topics has been driven by customer requirements. All of these topics have been requested by clients in various courses.

Suggested Uses

The book has been structured in such a way that it breaks easily into weekly topics. There is a core set of chapters that can be used as an introductory course, in a single semester, and a further set of chapters that can be used for intermediate study, for

follow-on, longer or double-weighted courses. It contains exercises throughout, designed to reinforce learning about the topics covered in each part of the chapter. The final exercises at the end of each chapter draw together the key aspects that have been covered, which are also reiterated in chapter summaries.

Foundational Java can be used for courses of different lengths and levels by using it in three different ways. The first 12 chapters, listed below, cover the core knowledge of Java, and provide a solid basis for an introductory course on object oriented programming with Java. These fit easily into a 12 week semester, providing 11 weeks of teaching material (Chap. 1 is just an introduction) and opportunities for revision and reflection.

1. The Java Story
2. Compiling and Running Java Programs
3. Data Types, Arithmetic, and Arrays
4. Control Structures
5. Creating Objects
6. Creating Domain Classes
7. Objects Working Together: Association, Aggregation, and Composition
8. Inheritance, Polymorphism, and Interfaces
9. Exception Handling
10. Unit Testing with JUnit
11. Exploring the Java Libraries
12. The Collections Framework and Generics

The following four chapters, listed below, are more intermediate and provide more specialized coverage of Java; interaction with external connections to files, databases and build tools, as well as providing an introduction to multithreading. These are useful in longer semesters, or courses that require intermediate level study.

13. Input and Output Streams
14. Automatic Building and Testing with Ant
15. Java and the Database (JDBC)
16. Multithreading

The final four chapters are specific to building applications with a graphical user interface, and launching applications from a web server, and provide optional coverage for courses that have requirements for this type of programming. They provide additional resources and flexibility for longer or broader courses.

17. Building GUIs with the JFC Swing Library
18. Event-Driven Programming
19. Dialogs and Menus, Models and Views
20. Java Web Start and Applets

Supplemental Resources

A number of supplemental resources are available from the book's website at <http://www.introjava.com>

Resources on the website for students include

- Downloadable source code for all the examples in the book
- Downloadable source code for solutions to selected exercises
- Self-test questions

Additional resources for instructors include

- A complete set of PowerPoint slides
- Downloadable source code for solutions to all exercises

A note about the code

Source code in the text appears in a Courier font to mark it out clearly from the surrounding text

Java source code appears in this font

Due to the page width, it has often proved necessary to break lines of code in places where the original source code (which can be downloaded from the website) would not have a line break.

In the majority of cases the line breaks have been inserted so that they do not affect compilation. For example, this code statement appears in Chap. 3.

```
double mean =  
    ((double)intArray[0] + intArray[1] + intArray[2])  
    / intArray.length;
```

This is, in fact, a single statement, even though it is broken across three lines of text. A single statement in Java is terminated by a semicolon, and line feeds do not, in most cases, affect the way the code works. However in a very small number of cases it has not been possible to break lines within the margin constraints of the book in such a way that their workings are unaffected. If in doubt, or if you are having problems with compiling or running code, please refer to the original source code files.

Acknowledgements

It is difficult to acknowledge all the individuals who have contributed to this book, because my experience of Java has been so long and broad. I am grateful to the many authors of courseware from my various past employers: The Object People, BEA Systems, Valtech, IBM and Software Education Associates. The experience of teaching from material authored by others, however good or bad it is, provides new perspectives and understanding that goes way beyond what is possible when only teaching from your own perspective.

I am grateful to my colleague Hokyoung Ryu, for giving me the impetus to (re) write this book and test the material on his students.

I am also grateful to Wayne Wheeler and Simon Rees at Springer-Verlag London Ltd., who gave me the opportunity to bring this book to publication.

Contents

1	The Java Story.....	1
1.1	A Brief History of Java	1
1.2	Characteristics of Java.....	2
1.2.1	Simple	2
1.2.2	Object-Oriented.....	3
1.2.3	Distributed.....	3
1.2.4	Robust	3
1.2.5	Secure.....	4
1.2.6	Architecture-Neutral	4
1.2.7	Portable	5
1.2.8	High-Performing	5
1.2.9	Multithreaded	6
1.2.10	Dynamic	6
1.3	The JDK and the JRE.....	6
1.3.1	Java Versions	7
1.4	Java APIs.....	8
1.5	Summary	8
2	Compiling and Running Java Programs.....	9
2.1	Java from the Command Line	9
2.1.1	Setting the Path to the JDK	9
2.2	A First Java Program.....	10
2.2.1	The MyJavaProgram Class	11
2.2.2	The “main” Method	12
2.2.3	Output with the System Class	12
2.3	Compiling Java	13
2.4	Running Java	14
2.5	The Java Classpath.....	15
2.5.1	Packages.....	16
2.5.2	Compiling into a Package Folder.....	17
2.6	Comment Syntax.....	19

2.7	Using the Eclipse Integrated Development Environment (IDE).....	20
2.7.1	Creating a New Project	20
2.7.2	Editing Code in Eclipse.....	24
2.7.3	Compiling and Running Code in Eclipse.....	27
2.8	Summary	28
3	Data Types, Arithmetic, and Arrays	29
3.1	Java Primitive Data Types.....	29
3.1.1	Literals and Variables.....	30
3.1.2	Literals and the Compiler.....	32
3.1.3	Boolean Variables	33
3.1.4	Unicode Characters and Escape Sequences	33
3.1.5	Variable Scope.....	34
3.2	Arithmetic and Other Operators.....	36
3.2.1	Arithmetic Statements.....	37
3.2.2	Increment and Decrement Operators	38
3.2.3	Prefix and Postfix Operators	38
3.2.4	Assignment Operators.....	39
3.2.5	Promotion and Typecasting.....	40
3.2.6	String Concatenation.....	40
3.2.7	Bitwise Operators.....	41
3.3	Arrays.....	42
3.3.1	Declaring Arrays.....	43
3.3.2	Initializing Arrays	44
3.3.3	Accessing Array Data	44
3.3.4	The Array “length” Field	45
3.3.5	Multidimensional Arrays	45
3.4	Precedence, Casting, and Promotion	45
3.5	Summary	48
4	Control Structures	49
4.1	Making Selections.....	49
4.1.1	“if” Statements	49
4.1.2	Expressing Conditions with Relational and Boolean Operators	51
4.1.3	Using Selection: The CoinExample Class	52
4.1.4	The DieExample Class.....	53
4.1.5	Writing Test Code	55
4.1.6	“switch” Statements	56
4.1.7	The Ternary Operator.....	59
4.2	Iteration.....	59
4.2.1	“while” and “do...while” Loops.....	60
4.2.2	“for” Loops	61
4.3	Summary	70

5	Creating Objects	71
5.1	Classes and Objects.....	71
5.1.1	Object Creation: Constructors.....	72
5.2	Classes and Methods for String Data.....	73
5.2.1	The String Class	73
5.2.2	StringBuilders	76
5.2.3	StringBuffers	78
5.2.4	The “toString” Method.....	78
5.3	References and Memory	79
5.3.1	Garbage Collection	81
5.3.2	Object Equality	82
5.3.3	String Concatenation and Memory	83
5.4	Java Library Packages.....	84
5.4.1	Importing Classes.....	85
5.4.2	Wild Cards and Multiple Imports	86
5.4.3	Packages and Sub-packages.....	87
5.4.4	Managing Imports with Eclipse	88
5.5	Using Javadoc	90
5.5.1	Viewing Javadoc in Eclipse.....	90
5.6	Summary	92
6	Creating Domain Classes.....	95
6.1	Object Orientation and Domain Objects.....	95
6.1.1	Clock Objects in the Real World.....	96
6.1.2	Encapsulation and Information Hiding.....	97
6.1.3	Object Identity.....	97
6.2	Creating New Classes	98
6.2.1	Adding Fields to a Class	99
6.2.2	Adding Methods to a Class	100
6.3	Constructors	104
6.3.1	Overloaded Constructors.....	105
6.3.2	Chaining Constructors.....	106
6.3.3	Internal Encapsulation in Constructors	106
6.4	Access Modifiers.....	107
6.5	Javadoc and Code Comments	108
6.5.1	Creating Javadoc in Eclipse	111
6.5.2	Running Javadoc from the Command Line	113
6.6	Types of Method	114
6.7	Object Reuse	115
6.8	Static Fields and Methods.....	117
6.8.1	Static Methods.....	118
6.8.2	Invoking Static Methods	119
6.8.3	Static Final Fields.....	119
6.8.4	Static Methods in Java Library Classes	120

6.9	Objects as JavaBean Components.....	121
6.9.1	Reflection	122
6.10	Summary	124
7	Objects Working Together: Association, Aggregation, and Composition	125
7.1	Aggregations as Collections.....	126
7.2	Message Passing	126
7.2.1	Object Association	127
7.2.2	Association Direction.....	128
7.2.3	Implementing Associations.....	129
7.3	Associating Objects: The Snakes and Ladders Game.....	130
7.3.1	The Snakes and Ladders Game Source Code	136
7.3.2	When to Create Objects	144
7.4	Association, Aggregation, or Composition?.....	145
7.4.1	Aggregation Example	146
7.5	Composition.....	149
7.5.1	Electronic Gates	149
7.5.2	Half Adder Components	153
7.6	Summary	157
8	Inheritance, Polymorphism, and Interfaces	159
8.1	Abstraction.....	159
8.1.1	Inheritance.....	159
8.1.2	Polymorphism	160
8.1.3	The Relationship Between Inheritance and Polymorphism	160
8.2	Inheriting from the Object Class.....	163
8.2.1	Overriding the “toString” Method	165
8.2.2	Overriding the “equals” Method	166
8.2.3	Overriding the “hashCode” Method	168
8.2.4	The @Override Annotation.....	169
8.3	An Inheritance Hierarchy Using Abstraction	169
8.3.1	Abstract Classes	171
8.3.2	Protected Visibility and Inheritance.....	172
8.3.3	Creating Subclasses.....	174
8.3.4	Calling Superclass Constructors	175
8.3.5	Adding Further Subclasses.....	177
8.4	Dynamic Binding	179
8.4.1	Using Polymorphic Methods	186
8.5	Interfaces.....	189
8.5.1	Naming Interfaces.....	191
8.5.2	Implementing an Interface	192
8.6	Summary	195

9	Exception Handling	197
9.1	Java Exceptions.....	197
9.1.1	The Exception Handling Hierarchy	197
9.1.2	Checked and Unchecked Exceptions	199
9.1.3	Exception Handling Keywords	199
9.2	Handling Checked Exceptions.....	199
9.2.1	Catching Exceptions: “try” and “catch”.....	200
9.3	Handling Unchecked Exceptions.....	202
9.3.1	Exiting.....	204
9.4	Catching Multiple Exceptions.....	205
9.4.1	Ordering Multiple “catch” Blocks	206
9.5	Throwing an Exception with “throw”	207
9.5.1	Delegating Responsibility	209
9.6	Writing and Using Custom Exceptions.....	211
9.6.1	Throwing a Custom Exception.....	213
9.6.2	“finally” Blocks.....	214
9.6.3	Re-throwing Custom Exceptions	215
9.7	Summary	218
10	Unit Testing with JUnit.....	219
10.1	The JUnit Test Framework.....	220
10.1.1	Using JUnit with Eclipse.....	220
10.2	Test Cases and Units Under Test.....	222
10.2.1	Creating a JUnit Test Case	224
10.2.2	Required Imports.....	226
10.2.3	Writing Test Methods.....	227
10.2.4	Using TestRunners	228
10.2.5	The Eclipse TestRunner	229
10.3	Types of Assertions.....	232
10.3.1	Asserting Floating Point Equality	232
10.3.2	Object Equality	233
10.3.3	Other Assertions.....	233
10.4	Exceptions, Timeouts, and Failures	235
10.4.1	Forcing Failures	236
10.5	Arranging a Test with @Before and @After Methods	237
10.5.1	The 3A Pattern	239
10.5.2	@BeforeClass and @AfterClass Methods.....	242
10.6	Writing a Test Suite	243
10.7	Summary	244
11	Exploring the Java Libraries	245
11.1	Frequently Used Classes in the Java.lang Package.....	245
11.1.1	The Java.lang.Object Class	246
11.1.2	The Java.lang.Math Class	248
11.1.3	The Java.lang.System Class	249
11.1.4	Wrapper Classes.....	251

11.2	Classes in the Java.util Package	252
11.2.1	The Date Class	252
11.2.2	The Calendar Class and Factory Methods	253
11.3	Formatter Classes in the Java.text Package.....	255
11.3.1	Formatting Dates.....	255
11.3.2	Parsing Dates.....	257
11.3.3	Formatting and Parsing Numbers.....	258
11.3.4	Formatting and Parsing Currency	260
11.4	Summary	261
12	The Collections Framework and Generics	263
12.1	Objects That Contain Objects	263
12.1.1	Associations and Collections	264
12.2	The Core Collection Interfaces	264
12.2.1	Partial Implementations of the Core Interfaces.....	266
12.2.2	Concrete Implementations of the Core Interfaces.....	266
12.2.3	Legacy Classes	267
12.3	Typesafe Collections with Generics.....	267
12.4	A Concrete Collection Class: The ArrayList	269
12.4.1	Adding and Retrieving Objects	269
12.4.2	Wrapping and Autoboxing.....	270
12.4.3	Iterators	271
12.4.4	ListIterator.....	272
12.4.5	Enhanced “for” Loop	272
12.4.6	An ArrayList of Module Objects	273
12.5	Maps.....	275
12.5.1	Map Views	276
12.5.2	Using a TreeMap.....	277
12.6	Utility Classes	280
12.6.1	Comparing Objects with the Comparable Interface.....	280
12.6.2	The Collections Class.....	287
12.6.3	Creating a Comparator	290
12.7	Generics and Inheritance.....	293
12.7.1	Specifying Bounds	294
12.8	Summary	296
13	Input and Output Streams	297
13.1	Java Stream Classes	297
13.1.1	Different Types of Stream	298
13.2	Byte Streams	299
13.2.1	Streaming Bytes To and From Files.....	299
13.3	Filter Streams	303
13.4	Readers and Writers.....	306
13.4.1	Buffered Readers and Writers.....	306
13.5	PrintStreams and PrintWriters	307

13.6	The File Class	308
13.7	Streaming Objects	309
13.7.1	Serializing Course Objects	310
13.8	The New IO Library	314
13.8.1	Buffer Classes	314
13.8.2	File Channels	316
13.8.3	View Buffers	318
13.8.4	Charsets and String Data	318
13.9	Summary	320
14	Automatic Building and Testing with Ant	321
14.1	Installing and Configuring Ant	321
14.1.1	Using Ant in Eclipse	322
14.2	The Ant Build File: “build.xml”	322
14.2.1	Ant Properties	323
14.2.2	Tasks and Targets	324
14.2.3	Running an Ant Build File in Eclipse	326
14.2.4	Packaging Code with the Ant “jar” Task	328
14.2.5	Target Dependencies	329
14.3	Running Code, Forking, and Classpaths	330
14.3.1	Setting the Classpath with a “path” Element	332
14.4	Running Tests	333
14.4.1	Running Tests with the “junit” and “test” Elements	333
14.4.2	Running Multiple Tests with the “batchtest” Element	335
14.5	Summary	336
15	Java and the Database (JDBC)	337
15.1	An Example Database	337
15.2	Using MySQL	338
15.2.1	Creating a New Database	340
15.2.2	Generating Primary Keys	341
15.2.3	Using a DDL Script	341
15.2.4	Viewing Table Schema	343
15.2.5	Creating an Authorized MySQL User	343
15.3	Java Database Access with JDBC	344
15.3.1	JDBC Drivers	345
15.3.2	Making a Two-Tier Connection	347
15.4	SQL Exceptions	348
15.5	Executing SQL Queries and Updates Using JDBC	349
15.5.1	Processing ResultSets	349
15.5.2	Updating Records	353
15.6	Using Prepared Statements	353
15.7	Summary	355

16 Multithreading	357
16.1 Creating and Running a Thread	358
16.1.1 Thread States	360
16.2 Running Multiple Threads	361
16.3 Thread Priority	364
16.3.1 Setting Thread Priority	365
16.3.2 Yielding	367
16.4 Implementing the Runnable Interface	368
16.5 Synchronizing Threads	377
16.5.1 Monitors and Synchronization	378
16.5.2 “wait,” “notify,” and “notifyAll”	379
16.6 Synchronized Code Example	380
16.6.1 The Sensor Class	380
16.6.2 The Analyzer Class	383
16.7 Concurrent Collections	385
16.8 Summary	386
17 Building GUIs with the JFC Swing Library	387
17.1 Components, Containers, and Frames	388
17.2 Creating a Main Window Frame	390
17.2.1 Setting the JFrame’s Title	390
17.2.2 Selecting the JFrame’s Closing Behavior	390
17.2.3 Sizing the JFrame	391
17.2.4 Showing the JFrame	391
17.3 Swing Component Classes	392
17.3.1 The JLabel Class	392
17.3.2 Adding Components to a Frame	393
17.3.3 Manually Positioning and Sizing Components	395
17.3.4 Common Methods of Component and JComponent	395
17.3.5 The JTextField Class	396
17.3.6 The JButton Class	396
17.3.7 Placing Components Using “setBounds”	397
17.4 Colors, Fonts, and Icons	399
17.4.1 Setting Colors	399
17.4.2 Setting Fonts	400
17.4.3 Font Style and Size	401
17.4.4 Icons on Labels and Buttons	401
17.5 Some Additional Components	403
17.5.1 JTextArea	403
17.5.2 Check Boxes and Radio Buttons	403
17.5.3 JComboBox	404
17.5.4 JSlider	404
17.5.5 JPanel	405
17.6 Setting the Look and Feel	405

17.7	Layout Managers	407
17.7.1	BorderLayout	408
17.7.2	FlowLayout	411
17.7.3	GridLayout	414
17.8	Summary	415
18	Event-Driven Programming	417
18.1	Event Listeners	417
18.2	Event Classes	418
18.3	EventListener Interfaces	418
18.3.1	A JButton ActionListener	419
18.3.2	Adding a Listener	420
18.4	Multiple Action and Focus Listeners	421
18.5	Responding to Mouse Events	425
18.5.1	Mouse Listener Events	426
18.5.2	Mouse Motion Events	429
18.6	Event Handlers as Inner Classes	432
18.6.1	Implementing a DieListener	432
18.6.2	Inner Classes	434
18.6.3	Local Inner Classes	435
18.6.4	Anonymous Inner Classes	436
18.7	JPanel Subclasses and Multi-panel Layouts	437
18.7.1	The GridBagLayout	443
18.8	Separating the “Model” from the “View”	453
18.8.1	The Model Class: FlightDistances	454
18.8.2	The View Class: DistanceViewer	460
18.9	Summary	463
19	Dialogs and Menus, Models and Views	465
19.1	Predefined Dialogs in Swing	465
19.1.1	Message Box Dialogs Using JOptionPane	465
19.1.2	File Chooser Dialogs Using JFileChooser	466
19.1.3	Color Chooser Dialogs Using JColorChooser	468
19.2	Custom Dialogs with JDialog	470
19.3	Using Menus	477
19.3.1	Adding a Menu to a Frame	478
19.3.2	Menu Enhancement: Separators, Mnemonics, and Accelerators	479
19.3.3	MenuListeners	481
19.4	Model View Controller in Swing Components	486
19.4.1	Model and View in JTextPane Components	487
19.4.2	Model and View in the JTable Component	494
19.5	Summary	501