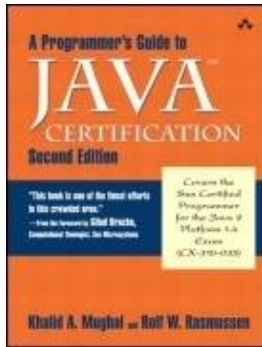


[Team LiB]



[Table of Contents](#)

[Index](#)

[Examples](#)

Programmer's Guide to Java™ Certification, A: A Comprehensive Primer, Second Edition

By [Khalid A. Mughal](#), [Rolf W. Rasmussen](#)

START READING

Publisher: Addison Wesley

Pub Date: August 04, 2003

ISBN: 0-201-72828-1

Pages: 672

To pass the Sun Certified Programmer for Java 2 Platform 1.4 exam (SCPJ2 1.4) you need this book. It is written for any experienced programmer (with or without previous knowledge of Java) interested in mastering the Java programming language and passing the SCPJ2 1.4 exam.

A Programmer's Guide to Java(TM) Certification, Second Edition, contains detailed coverage of all exam topics and objectives, helpful code examples, exercises, review questions, and several full practice exams. In addition, as a comprehensive primer to the Java programming language, this book is an invaluable reference tool for the reader.

This new edition has been thoroughly updated to focus on the latest version of the exam (CX-310-035). In particular, it contains more in-depth explanations of the syntax and usage of core language features that are now emphasized on the exam. The accompanying CD contains a version of the SCPJ2 1.4 Exam Simulator developed by Whizlabs Software, which includes several mock exams and simulates the exam-taking experience. The CD also contains the complete source code for all the examples, and solutions to the programming exercises.

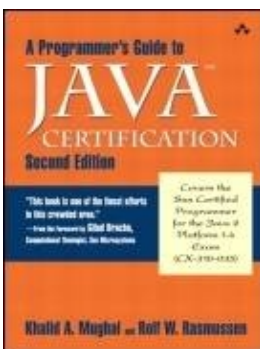
What you will find in this book:

- Extensive coverage of all the objectives defined for the Sun Certified Programmer for Java 2 Platform 1.4 exam
- Easy-to-follow structure with chapters organized according to the exam objectives as laid out by Sun Microsystems
- Summaries that clearly state and differentiate the exam objectives and the supplementary objectives to be covered in each chapter

- A list of Sun's objectives for the SCPJ2 1.4 exam, and a guide to taking the exam
- A complete mock exam with new questions (not repeats of review questions)
- Numerous exam-relevant review questions to test your understanding of each major topic, with annotated answers
- Programming exercises and solutions at the end of each chapter
- Copious code examples illustrating concepts where the code has been compiled and thoroughly tested on multiple platforms
- Program output demonstrating expected results from running the examples
- Extensive use of UML (Unified Modeling Language) for illustration purposes
- An introduction to basic terminology and concepts in object-oriented programming
- Advice on how to avoid common pitfalls in mastering the language and taking the exam
- Platform-independent coverage--platform-specific details are provided where relevant
- Information about the SCPJ2 Upgrade exam

[[Team LiB](#)]

[[Team LiB](#)]



[Table of Contents](#)

[Index](#)

[Examples](#)

Programmer's Guide to Java™ Certification, A: A Comprehensive Primer, Second Edition

By [Khalid A. Mughal](#), [Rolf W. Rasmussen](#)

START READING

Publisher: Addison Wesley

Pub Date: August 04, 2003

ISBN: 0-201-72828-1

Pages: 672

[Copyright](#)

[List of Figures](#)

[List of Tables](#)

[List of Examples](#)

[Foreword](#)

[Preface](#)

[Writing the Second Edition](#)

[About This Book](#)

[Using the Book](#)

[Review Questions](#)

- Chapter Summary
- Programming Exercises
- Book Web Sites
- Request for Feedback
- About the Authors
- Acknowledgments (First Edition)
- Acknowledgments (Second Edition)
- Chapter 1. Basics of Java Programming
 - Section 1.1. Introduction
 - Section 1.2. Classes
 - Section 1.3. Objects
 - Section 1.4. Instance Members
 - Section 1.5. Static Members
 - Section 1.6. Inheritance
 - Section 1.7. Aggregation
 - Section 1.8. Tenets of Java
 - Review Questions
 - Section 1.9. Java Programs
 - Section 1.10. Sample Java Application
 - Review Questions
 - Chapter Summary
 - Programming Exercises
- Chapter 2. Language Fundamentals
 - Section 2.1. Basic Language Elements
 - Review Questions
 - Section 2.2. Primitive Data Types
 - Review Questions
 - Section 2.3. Variable Declarations
 - Section 2.4. Initial Values for Variables
 - Review Questions
 - Section 2.5. Java Source File Structure
 - Review Questions
 - Section 2.6. The `main()` Method
 - Review Questions
 - Chapter Summary
 - Programming Exercises
- Chapter 3. Operators and Assignments
 - Section 3.1. Precedence and Associativity Rules for Operators
 - Section 3.2. Evaluation Order of Operands
 - Section 3.3. Conversions
 - Section 3.4. Simple Assignment Operator =
 - Review Questions
 - Section 3.5. Arithmetic Operators: `*`, `/`, `%`, `+`, `-`
 - Review Questions
 - Section 3.6. The Binary String Concatenation Operator `+`
 - Section 3.7. Variable Increment and Decrement Operators: `++`, `--`
 - Review Questions
 - Section 3.8. Boolean Expressions
 - Section 3.9. Relational Operators: `<`, `<=`, `>`, `>=`
 - Section 3.10. Equality
 - Section 3.11. Boolean Logical Operators: `!`, `^`, `&`, `|`
 - Section 3.12. Conditional Operators: `&&`, `||`
 - Review Questions
 - Section 3.13. Integer Bitwise Operators: `~`, `&`, `|`, `^`

- Section 3.14. Shift Operators: <<, >>, >>>
- Section 3.15. The Conditional Operator: ?
- Section 3.16. Other Operators: new, [], instanceof
- Review Questions
- Section 3.17. Parameter Passing
- Section 3.18. Passing Primitive Data Values
- Section 3.19. Passing Object Reference Values
- Section 3.20. Passing Array References
- Section 3.21. Array Elements as Actual Parameters
- Section 3.22. final Parameters
- Section 3.23. Program Arguments
- Review Questions
- Chapter Summary
- Programming Exercises

Chapter 4. Declarations and Access Control

- Section 4.1. Arrays
- Review Questions
- Section 4.2. Defining Classes
- Section 4.3. Defining Methods
- Section 4.4. Constructors
- Section 4.5. Scope Rules
- Review Questions
- Section 4.6. Packages
- Section 4.7. Accessibility Modifiers for Top-level Classes and Interfaces
- Section 4.8. Other Modifiers for Classes
- Review Questions
- Section 4.9. Member Accessibility Modifiers
- Review Questions
- Section 4.10. Other Modifiers for Members
- Review Questions
- Chapter Summary
- Programming Exercises

Chapter 5. Control Flow, Exception Handling, and Assertions

- Section 5.1. Overview of Control Flow Statements
- Section 5.2. Selection Statements
- Review Questions
- Section 5.3. Iteration Statements
- Section 5.4. Transfer Statements
- Review Questions
- Section 5.5. Stack-based Execution and Exception Propagation
- Section 5.6. Exception Types
- Section 5.7. Exception Handling: try, catch, and finally
- Section 5.8. throw Statement
- Section 5.9. throws Clause
- Review Questions
- Section 5.10. Assertions
- Review Questions
- Chapter Summary
- Programming Exercises

Chapter 6. Object-oriented Programming

- Section 6.1. Single Implementation Inheritance
- Review Questions
- Section 6.2. Overriding and Hiding Members
- Review Questions

- Section 6.3. Chaining Constructors Using `this()` and `super()`
- Review Questions
- Section 6.4. Interfaces
- Review Questions
- Section 6.5. Completing the Type Hierarchy
- Section 6.6. Assigning, Passing, and Casting Reference Values
- Review Questions
- Section 6.7. Polymorphism and Dynamic Method Lookup
- Review Questions
- Section 6.8. Inheritance vs. Aggregation
- Review Questions
- Chapter Summary
- Programming Exercises
- Chapter 7. Nested Classes And Interfaces
 - Section 7.1. Overview of Nested Classes and Interfaces
 - Section 7.2. Static Member Classes and Interfaces
 - Section 7.3. Non-static Member Classes
 - Review Questions
 - Section 7.4. Local Classes
 - Section 7.5. Anonymous Classes
 - Review Questions
 - Chapter Summary
 - Programming Exercise
- Chapter 8. Object Lifetime
 - Section 8.1. Garbage Collection
 - Review Questions
 - Section 8.2. Initializers
 - Review Questions
 - Chapter Summary
- Chapter 9. Threads
 - Section 9.1. Multitasking
 - Section 9.2. Overview of Threads
 - Section 9.3. Thread Creation
 - Review Questions
 - Section 9.4. Synchronization
 - Review Questions
 - Section 9.5. Thread Transitions
 - Review Questions
 - Chapter Summary
 - Programming Exercises
- Chapter 10. Fundamental Classes
 - Section 10.1. Overview of the `java.lang` Package
 - Section 10.2. The `Object` Class
 - Review Questions
 - Section 10.3. The Wrapper Classes
 - Review Questions
 - Section 10.4. The `Math` Class
 - Review Questions
 - Section 10.5. The `String` Class
 - Review Questions
 - Section 10.6. The `StringBuffer` Class
 - Review Questions
 - Chapter Summary
 - Programming Exercises

Chapter 11. Collections and Maps

Section 11.1. The Collections Framework

Section 11.2. Collections

Review Questions

Section 11.3. Sets

Section 11.4. Lists

Review Questions

Section 11.5. Maps

Section 11.6. Sorted Sets and Sorted Maps

Review Questions

Section 11.7. Implementing the `equals()`, `hashCode()`, and `compareTo()` Methods

Section 11.8. Working with Collections

Review Questions

Chapter Summary

Programming Exercises

Appendix A. Taking the SCPJ2 1.4 Exam

Section A.1. Preparing for the Programmer Exam

Section A.2. Registering for the Exam

Section A.3. How the Examination Is Conducted

Section A.4. The Questions

Section A.5. Moving on to the Developer Exam

Appendix B. Objectives for the SCPJ2 1.4 Exam

Section 1: Declarations and Access Control (Chapter 4)

Section 2: Flow Control, Assertions, and Exception Handling (Chapter 5)

Section 3: Garbage Collection (Chapter 8)

Section 4: Language Fundamentals (Chapter 2)

Section 5: Operators and Assignments (Chapter 3)

Section 6: Overloading, Overriding, Runtime Type and Object Orientation (Chapter 6, Chapter 11)

Section 7: Threads (Chapter 9)

Section 8: Fundamental Classes in the `java.lang` Package (Chapter 10)

Section 9: The Collections Framework (Chapter 11)

Appendix C. Objectives for the Java 2 Platform Upgrade Exam

Section 1: Declarations and Access Control

Section 2: Flow Control, Assertions, and Exception Handling

Section 3: Garbage Collection

Section 4: Language Fundamentals

Section 5: Operators and Assignments

Section 6: Overloading, Overriding, Runtime Type and Object Orientation

Section 7: Threads

Section 8: The Collections Framework

Appendix D. Annotated Answers to Review Questions

Section 1. Basics of Java Programming

Section 2. Language Fundamentals

Section 3. Operators and Assignments

Section 4. Declarations and Access Control

Section 5. Control Flow, Exception Handling, and Assertions

Section 6. Object-oriented Programming

Section 7. Nested Classes and Interfaces

Section 8. Object Lifetime

Section 9. Threads

Section 10. Fundamental Classes

Section 11. Collections and Maps

Appendix E. Solutions to Programming Exercises

- Section 1. Basics of Java Programming
- Section 2. Language Fundamentals
- Section 3. Operators and Assignments
- Section 4. Declarations and Access Control
- Section 5. Control Flow, Exception Handling, and Assertions
- Section 6. Object-oriented Programming
- Section 7. Nested Classes and Interfaces
- Section 8. Basics of Java Programming
- Section 9. Threads
- Section 10. Fundamental Classes
- Section 11. Collections and Maps

Appendix F. Mock Exam

- Answers to Questions

Appendix G. Number Systems and Number Representation

- Section G.1. Number Systems
- Section G.2. Relationship between Binary, Octal, and Hexadecimal Numbers
- Section G.3. Converting Decimals
- Section G.4. Representing Integers

Appendix H. About the CD

- Section H.1. Whizlabs Exam Simulators
- Section H.2. Items from the Book

Single-User License Agreement

- Important Terms of Agreement
- Tenure of the License Agreement
- Limited Warranty Under the License
- Remedies Provided Under the License
- Liabilities Under the License
- Other Limitations Under the License
- General Note

Index

[Team LiB]

[Team LiB]

Copyright

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Addison-Wesley was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers discounts on this book when ordered in quantity for bulk purchases and special sales. For more information, please contact:

U.S. Corporate and Government Sales
 (800) 382-3419
corpsales@pearsontechgroup.com

For sales outside of the U.S., please contact:

International Sales
(317) 581-3793
international@pearsontechgroup.com

Visit Addison-Wesley on the Web: www.awprofessional.com

Library of Congress Cataloging-in-Publication Data is available.

Copyright & "docText">All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher. Printed in the United States of America. Published simultaneously in Canada.

For information on obtaining permission for use of material from this work, please submit a written request to:

Pearson Education, Inc.
Rights and Contracts Department
75 Arlington Street, Suite 300
Boston, MA 02116
Fax: (617) 848-7047

Text printed on recycled paper

1 2 3 4 5 6 7 8 9 10 CRS 0706050403

First printing, August 2003

Dedication

To the loving memory of my mother, Zubaida Begum, and my father, Mohammed Azim.

K. A.M.

For Olivia E. Rasmussen and Louise J. Dahlmo.

R.W.R.

[[Team LiB](#)]
[[Team LiB](#)]

List of Figures

[Chapter 1](#)

[1.1 UML Notation for Classes](#)

[1.2 UML Notation for Objects](#)

- 1.3 Aliases
- 1.4 Class Diagram Showing Static Members of a Class
- 1.5 Members of a Class
- 1.6 Class Diagram Depicting Inheritance Relation
- 1.7 Class Diagram Depicting Aggregation

Chapter 2

- 2.1 Primitive Data Types in Java
- 2.2 Java Source File Structure

Chapter 3

- 3.1 Widening Numeric Conversions
- 3.2 Overflow and Underflow in Floating-point Arithmetic
- 3.3 Numeric Promotion in Arithmetic Expressions
- 3.4 Parameter Passing: Primitive Data Values
- 3.5 Parameter Passing: Object Reference Values
- 3.6 Parameter Passing: Arrays

Chapter 4

- 4.1 Array of Arrays
- 4.2 Block Scope
- 4.3 Package Hierarchy
- 4.4 Public Accessibility
- 4.5 Protected Accessibility
- 4.6 Default Accessibility
- 4.7 Private Accessibility

Chapter 5

- 5.1 Activity Diagram for `if` Statements
- 5.2 Activity Diagram for `switch` Statement
- 5.3 Activity Diagram for `while` Statement

- 5.4 Activity Diagram for `do-while` Statement
- 5.5 Activity Diagram for the `for` Statement
- 5.6 Method Execution
- 5.7 Exception Propagation
- 5.8 Partial Exception Inheritance Hierarchy
- 5.9 `try-catch-finally` Block
- 5.10 Exception Handling (Scenario 1)
- 5.11 Exception Handling (Scenario 2)
- 5.12 Exception Handling (Scenario 3)
- 5.13 Execution of the Simple `assert` Statement (When Assertions Are Enabled)

Chapter 6

- 6.1 Inheritance Hierarchy
- 6.2 Inheritance Relationship between `String` and `Object` Classes
- 6.3 Inheritance Relations
- 6.4 Reference Type Hierarchy
- 6.5 Polymorphic Methods
- 6.6 Implementing Data Structures by Inheritance and Aggregation

Chapter 7

- 7.1 Static Member Classes and Interfaces
- 7.2 Outer Object with Associated Inner Objects

Chapter 8

- 8.1 Memory Organization at Runtime

Chapter 9

- 9.1 Spawning Threads Using a `Runnable` Object
- 9.2 Spawning Threads Extending the `Thread` Class
- 9.3 Thread States
- 9.4 Running and Yielding

9.5 Sleeping and Waking up

9.6 Waiting and Notifying

9.7 Thread Communication

9.8 Stack Users

9.9 Joining of Threads

9.10 Deadlock

Chapter 10

10.1 Partial Inheritance Hierarchy in the `java.lang` Package

10.2 Converting Values between Primitive, Wrapper, and `String` Types

Chapter 11

11.1 The Core Interfaces

11.2 The Core Collection Interfaces and Their Implementations

11.3 The Core Map Interfaces and Their Implementations

11.4 Bulk Operations on Collections

G.1 Converting between Binary, Octal, and Hexadecimal

[Team LiB]

[Team LiB]

List of Tables

Chapter 1

1.1 Terminology for Class Members

Chapter 2

2.1 Keywords in Java

2.2 Reserved Literals in Java

2.3 Reserved Keywords not Currently in Use

2.4 Examples of Literals

2.5 Examples of Decimal, Octal, and Hexadecimal Literals

2.6 Examples of Unicode Values

- 2.7 [Escape Sequences](#)
- 2.8 [Examples of Escape Sequence \ddd](#)
- 2.9 [Range of Integer Values](#)
- 2.10 [Range of Character Values](#)
- 2.11 [Range of Floating-point Values](#)
- 2.12 [Boolean Values](#)
- 2.13 [Summary of Primitive Data Types](#)
- 2.14 [Default Values](#)

Chapter 3

- 3.1 [Operator Summary](#)
- 3.2 [Examples of Truncated Values](#)
- 3.3 [Arithmetic Operators](#)
- 3.4 [Examples of Arithmetic Expression Evaluation](#)
- 3.5 [Arithmetic Compound Assignment Operators](#)
- 3.6 [Relational Operators](#)
- 3.7 [Primitive Data Value Equality Operators](#)
- 3.8 [Reference Equality Operators](#)
- 3.9 [Boolean Logical Operators](#)
- 3.10 [Truth-values for Boolean Logical Operators](#)
- 3.11 [Boolean Logical Compound Assignment Operators](#)
- 3.12 [Conditional Operators](#)
- 3.13 [Truth-values for Conditional Operators](#)
- 3.14 [Integer Bitwise Operators](#)
- 3.15 [Result Table for Bitwise Operators](#)
- 3.16 [Examples of Bitwise Operations](#)
- 3.17 [Bitwise Compound Assignment Operators](#)
- 3.18 [Shift Operators](#)

3.19 Shift Compound Assignment Operators

3.20 Parameter Passing

Chapter 4

4.1 Accessing Members within a Class

4.2 Summary of Accessibility Modifiers for Classes and Interfaces

4.3 Summary of Other Modifiers for Classes and Interfaces

4.4 Summary of Accessibility Modifiers for Members

4.5 Summary of Other Modifiers for Members

Chapter 5

5.1 return Statement

5.2 Granularities for Enabling and Disabling Assertions at Runtime

5.3 Enabling and Disabling Assertions in All System Classes at Runtime

Chapter 6

6.1 Types and Values

Chapter 7

7.1 Overview of Classes and Interfaces

Chapter 8

Chapter 9

Chapter 10

10.1 Applying Rounding Functions

Chapter 11

11.1 Core Interfaces in the Collections Framework

11.2 Summary of Collection and Map Implementations

11.3 Bulk Operations and Set Logic

G.1 Number Systems

G.2 Representing Signed byte Values Using 2's Complement

[Team LiB]

[Team LiB]

List of Examples

Chapter 1

- 1.1 Basic Elements of a Class Definition
- 1.2 Static Members in Class Definition
- 1.3 Defining a Subclass
- 1.4 An Application

Chapter 2

- 2.1 Default Values for Fields
- 2.2 Flagging Uninitialized Local Variables of Primitive Data Types
- 2.3 Flagging Uninitialized Local Reference Variables

Chapter 3

- 3.1 Numeric Promotion in Arithmetic Expressions
- 3.2 Bitwise Operations
- 3.3 Passing Primitive Values
- 3.4 Passing Object Reference Values
- 3.5 Passing Arrays
- 3.6 Array Elements as Primitive Data Values
- 3.7 Array Elements as Object Reference Values
- 3.8 Passing Program Arguments

Chapter 4

- 4.1 Using Arrays
- 4.2 Using Anonymous Arrays
- 4.3 Using Multidimensional Arrays
- 4.4 Using `this` Reference
- 4.5 Namespaces
- 4.6 Class Scope
- 4.7 Accessibility Modifiers for Classes and Interfaces

- 4.8 Abstract Classes
- 4.9 Public Accessibility of Members
- 4.10 Accessing Static Members
- 4.11 Accessing Final Members
- 4.12 Synchronized Methods

Chapter 5

- 5.1 Fall Through in `switch` Statement
- 5.2 Using `break` in `switch` Statement
- 5.3 Nested `switch` Statement
- 5.4 `break` Statement
- 5.5 Labeled `break` Statement
- 5.6 `continue` Statement
- 5.7 Labeled `continue` Statement
- 5.8 The `return` Statement
- 5.9 Method Execution
- 5.10 `try-catch` Construct
- 5.11 Exception Propagation
- 5.12 `try-catch-finally` Construct
- 5.13 `try-finally` Construct
- 5.14 `finally` Block and the `return` Statement>
- 5.15 Throwing Exceptions
- 5.16 `throws` Clause
- 5.17 Assertions

Chapter 6

- 6.1 Extending Classes: Inheritance and Accessibility
- 6.2 Illustrating Inheritance
- 6.3 Overriding, Overloading, and Hiding

- 6.4 Overloaded Method Resolution
- 6.5 Using `super` Keyword
- 6.6 Constructor Overloading
- 6.7 `this ()` Constructor Call
- 6.8 `super ()` Constructor Call
- 6.9 Interfaces
- 6.10 Variables in Interfaces
- 6.11 Assigning and Passing Reference Values
- 6.12 `instanceof` and Cast Operator
- 6.13 Using `instanceof` Operator
- 6.14 Polymorphism and Dynamic Method Lookup
- 6.15 Implementing Data Structures by Inheritance and Aggregation

Chapter 7

- 7.1 Overview of Nested Classes and Interfaces
- 7.2 Static Member Classes and Interfaces
- 7.3 Importing Member Classes
- 7.4 Accessing Members in Enclosing Context
- 7.5 Defining Non-static Member Classes
- 7.6 Special Form of `this` and `new` Constructs in Non-static Member Classes
- 7.7 Inheritance Hierarchy and Enclosing Context
- 7.8 Extending Inner Classes
- 7.9 Access in Enclosing Context
- 7.10 Instantiating Local Classes
- 7.11 Defining Anonymous Classes
- 7.12 Accessing Declarations in Enclosing Context

Chapter 8

- 8.1 Garbage Collection Eligibility

- 8.2 Using Finalizers
- 8.3 Invoking Garbage Collection
- 8.4 Initializer Expression Order and Method Calls
- 8.5 Exceptions in Initializer Expressions
- 8.6 Static Initializers and Forward References
- 8.7 Static Initializer Blocks and Exceptions
- 8.8 Instance Initializers and Forward References
- 8.9 Instance Initializer Block in Anonymous Class
- 8.10 Exception Handling in Instance Initializer Blocks
- 8.11 Object State Construction
- 8.12 Initialization under Object State Construction

Chapter 9

- 9.1 Implementing the `Runnable` Interface
- 9.2 Extending the `Thread` Class
- 9.3 Mutual Exclusion
- 9.4 Waiting and Notifying
- 9.5 Joining of Threads
- 9.6 Thread Termination
- 9.7 Deadlock

Chapter 10

- 10.1 Methods in the `Object` class
- 10.2 String Representation of Integers
- 10.3 String Construction and Equality
- 10.4 Reading Characters from a String

Chapter 11

- 11.1 Using an Iterator
- 11.2 Using Sets

- 11.3 [Using Lists](#)
- 11.4 [Using Maps](#)
- 11.5 [Natural Order and Total Order](#)
- 11.6 [Using SortedMaps](#)
- 11.7 [A Simple Class for Version Number](#)
- 11.8 [Implications of Not Overriding the `equals\(\)` Method](#)
- 11.9 [Implementing the `equals\(\)` Method](#)
- 11.10 [Implications of Overriding the `equals\(\)` Method](#)
- 11.11 [Implementing the `hashCode\(\)` Method](#)
- 11.12 [Implications of Overriding the `hashCode\(\)` Method](#)
- 11.13 [Implementing the `compareTo\(\)` Method of the Comparable Interface](#)
- 11.14 [Implications of Implementing the `compareTo\(\)` Method](#)

[\[Team LiB \]](#)

[\[Team LiB \]](#)

Foreword

Consider the following observations:

- Software continues to become ever more pervasive, ever more ubiquitous in our lives.
- Incompetence seems to be the only thing we can count on in today's world and, especially, in the domain of software.
- The Java programming language has become a lingua franca for programmers all over the world.

One can draw varied conclusions from these comments. One of them is that it is of great importance that programmers working with the Java programming language should be as competent as possible.

The Java certification program is an important effort aimed at precisely this goal. Practitioners looking to obtain such certification need good quality training materials, which brings us to this book.

Programming is still more of an art than a science, and will continue to be so for the foreseeable future. Mastering the intricacies of a large and complex programming language is a challenging task that requires time and effort, and above all experience.

Real programming requires more than just mastery of a programming language. It requires mastery of a computing platform, with a rich set of libraries. These libraries are designed to simplify the task of building realistic applications, and they do. Again, the practitioner is faced with a daunting task.

To address the clear need for professional training material, a plethora of books have been written purporting to tutor programmers in the programming language and platform skills they require.

The choice is as mind boggling as the material within the books themselves. Should one try Java for Frontally Lobotomized Simians or Postmodern Java Dialectics? The readership for these books is largely self selecting. I trust that if you, the reader, have gotten this far, you are looking for something that is intelligent, yet practical. This book is one of the finest efforts in this crowded arena. It brings a necessary level of academic rigor to an area much in need of it, while retaining an essentially pragmatic flavor.

The material in this book is probably all you need to pass the Java certification exam. It certainly isn't all you need to be a good software engineer. You must continue learning about new technologies. The hardest part of this is dealing with things that are completely different from what you are familiar with. Yet this is what distinguishes the top flight engineer from the mediocre one. Keep an open mind; it pays.

Gilad Bracha
Computational Theologist
Sun Java Software
<http://java.sun.com/people/gbracha/>
[Team LiB]
[Team LiB]

Preface

[Team LiB]
[Team LiB]

Writing the Second Edition

The exam for the Sun Certified Programmer for Java 2 (SCPJ2) Platform has changed considerably since the first edition of this book was published. The most noticeable change in the current version of the exam (SCPJ2 1.4) is the removal of GUI and I/O, and the shifting of emphasis toward the core features of the language. In our opinion, the new exam demands an even greater understanding and actual experience of the language, rather than mere recitation of facts. Proficiency in the language is the key to success.

The I/O and GUI topics of the first edition have been eliminated, as they have no relevance for the SCPJ2 exam. These topics are more relevant for the Sun Certified Developer for Java 2 (SCDJ2) Platform exam and, therefore, relegated to a possible future volume on the Developer exam topics.

Since the emphasis of the SCPJ2 exam is on the core features of Java, the second edition provides an even greater in-depth coverage of the relevant topics. The book covers not just the exam objectives, but also supplementary topics that aid in mastering the exam topics.

The second edition is still a one-source guide for the SCPJ2 exam: it provides a mixture of theory and practice for the exam. Use the book to learn Java, pass the SCPJ2 exam, and afterwards, use it as a handy language guide. The book also has an appendix devoted to the SCPJ2 Upgrade exam. To facilitate preparation for the exams, the second edition comes with a CD containing additional resources, including several mock exams developed by Whizlabs Software.

We have taken into consideration the feedback we have received from readers. The many hours spent in handling the deluge of e-mail have not been in vain. Every single e-mail is appreciated and is hereby acknowledged.

Preparing the second edition dispelled all our illusions about second editions being, to put it colloquially, a piece of cake. Every sentence from the first edition has been weighed carefully, and not many paragraphs have escaped rewriting.

UML (Unified Modeling Language) is also extensively employed in this edition. Numerous new review questions have been added. In covering the new topics and expanding the existing ones, new examples, figures, and tables were also specifically created for the second edition.

[[Team LiB](#)]

[[Team LiB](#)]

About This Book

This book provides an extensive coverage of the Java programming language and its core Application Programming Interfaces (APIs), with particular emphasis on its syntax and usage. The book is primarily intended for professionals who want to prepare for the Sun Certified Programmer for Java 2 Platform 1.4 exam (referred to as the SCPJ2 exam), but it is readily accessible to any programmer who wants to master the language. For both purposes, it provides an in-depth coverage of essential features of the language and its core APIs.

There is a great and increasing demand for certified Java programmers. Sun Microsystems has defined the SCPJ2 exam as one that professionals can take to validate their skills. The certification provides the IT industry with the standard to use for hiring such professionals, and allows the professionals to turn their Java skills into credentials that are important for career advancement.

The book helps the reader master all core features of the Java language, and this mastering of the language can culminate in accomplishing the exam. It provides an extensive coverage of all the objectives defined for the exam by Sun. Since the exam objectives are selective, they do not cover many of the essential features of Java. This book provides extensive coverage of additional topics that every Java programmer should master in order to be proficient in this field. In this regard, the book is a comprehensive primer for learning the Java programming language.

This book is not a complete reference for Java, as it does not attempt to list every member of every class from the Java System Development Kit (SDK) API documentation. Its purpose is not to document the Java SDK APIs. It is also not a book on teaching programming techniques. Its emphasis is on the Java programming language features, their syntax and correct usage.

The book assumes a background in programming. We believe the exam is accessible to any programmer who works through the book. A Java programmer can easily skip over material that is well understood and concentrate on parts that need reinforcing, whereas a programmer new to Java will find the concepts explained from basic principles.

Each topic is explained and discussed thoroughly with examples, and backed by review questions and exercises to reinforce the concepts. The book is not biased toward any particular platform, but provides platform-specific details where necessary.

[[Team LiB](#)]

[[Team LiB](#)]

Using the Book

The reader can choose a linear or a non-linear route through the book, depending on her programming background. Non-Java programmers wishing to migrate to Java can read [Chapter 1](#), which provides a short introduction to object-oriented programming concepts, and the procedure for compiling and running Java applications. For those preparing for the SCPJ2 exam, the book has a separate appendix providing all the pertinent information on taking the exam.