

OCP

Oracle® Certified Professional
Java® SE 11 Developer

COMPLETE STUDY GUIDE

EXAM 1Z0-819 AND UPGRADE EXAM 1Z0-817

Includes one year of FREE access after activation to the interactive
online learning environment and study tools:

4 practice exams

Over 500 electronic flashcards

Searchable key term glossary

JEANNE BOYARSKY
SCOTT SELIKOFF

 **SYBEX**
A Wiley Brand

Table of Contents

[Cover](#)

[Acknowledgments](#)

[About the Authors](#)

[Introduction](#)

[Understanding the Exam](#)

[Reading This Book](#)

[Preparing for the Exam](#)

[Taking the Exam](#)

[Objective Map](#)

[Assessment Tests](#)

[PART I](#)

[Chapter 1: Welcome to Java](#)

[Learning About the Java Environment](#)

[Identifying Benefits of Java](#)

[Understanding the Java Class Structure](#)

[Writing a *main\(\)* Method](#)

[Understanding Package Declarations and Imports](#)

[Ordering Elements in a Class](#)

[Code Formatting on the Exam](#)

[Summary](#)

[Exam Essentials](#)

[Review Questions](#)

[Chapter 2: Java Building Blocks](#)

[Creating Objects](#)

[Understanding Data Types](#)

[Declaring Variables](#)

[Initializing Variables](#)

[Managing Variable Scope](#)

[Destroying Objects](#)

[Summary](#)

[Exam Essentials](#)

[Review Questions](#)

[Chapter 3: Operators](#)

[Understanding Java Operators](#)

[Applying Unary Operators](#)

[Working with Binary Arithmetic Operators](#)

[Assigning Values](#)

[Comparing Values](#)

[Making Decisions with the Ternary Operator](#)

[Summary](#)

[Exam Essentials](#)

[Review Questions](#)

[Chapter 4: Making Decisions](#)

[Creating Decision-Making Statements](#)

[Writing *while* Loops](#)

[Constructing *for* Loops](#)

[Controlling Flow with Branching](#)

[Summary](#)

[Exam Essentials](#)

[Review Questions](#)

[Chapter 5: Core Java APIs](#)

[Creating and Manipulating Strings](#)

[Using the *StringBuilder* Class](#)

[Understanding Equality](#)

[Understanding Java Arrays](#)

[Understanding an *ArrayList*](#)

[Creating Sets and Maps](#)

[Calculating with Math APIs](#)

[Summary](#)

[Exam Essentials](#)

[Review Questions](#)

[Chapter 6: Lambdas and Functional Interfaces](#)

[Writing Simple Lambdas](#)

[Introducing Functional Interfaces](#)

[Working with Variables in Lambdas](#)

[Calling APIs with Lambdas](#)

[Summary](#)

[Exam Essentials](#)

[Review Questions](#)

[Chapter 7: Methods and Encapsulation](#)

[Designing Methods](#)

[Working with Varargs](#)

[Applying Access Modifiers](#)

[Applying the *static* Keyword](#)

[Passing Data among Methods](#)

[Overloading Methods](#)

[Encapsulating Data](#)

[Summary](#)

[Exam Essentials](#)

[Review Questions](#)

[Chapter 8: Class Design](#)

[Understanding Inheritance](#)

[Creating Classes](#)

[Declaring Constructors](#)

[Inheriting Members](#)

[Understanding Polymorphism](#)

[Summary](#)

[Exam Essentials](#)

[Review Questions](#)

[Chapter 9: Advanced Class Design](#)

[Creating Abstract Classes](#)

[Implementing Interfaces](#)

[Introducing Inner Classes](#)

[Summary](#)

[Exam Essentials](#)

[Review Questions](#)

[Chapter 10: Exceptions](#)

[Understanding Exceptions](#)

[Recognizing Exception Classes](#)

[Handling Exceptions](#)

[Calling Methods That Throw Exceptions](#)

[Summary](#)

[Exam Essentials](#)

[Review Questions](#)

[Chapter 11: Modules](#)

[Introducing Modules](#)

[Creating and Running a Modular Program](#)

[Updating Our Example for Multiple Modules](#)

[Diving into the *module-info* File](#)

[Discovering Modules](#)

[Reviewing Command-Line Options](#)

[Summary](#)

[Exam Essentials](#)

[Review Questions](#)

[PART II](#)

[Chapter 12: Java Fundamentals](#)

[Applying the *final* Modifier](#)

[Working with Enums](#)

[Creating Nested Classes](#)

[Understanding Interface Members](#)

[Introducing Functional Programming](#)

[Summary](#)

[Exam Essentials](#)

[Review Questions](#)

[Chapter 13: Annotations](#)

[Introducing Annotations](#)

[Creating Custom Annotations](#)

[Applying Annotations](#)

[Declaring Annotation-Specific Annotations](#)

[Using Common Annotations](#)

[Summary](#)

[Exam Essentials](#)

[Review Questions](#)

[Chapter 14: Generics and Collections](#)

[Using Method References](#)

[Using Wrapper Classes](#)

[Using the Diamond Operator](#)

[Using Lists, Sets, Maps, and Queues](#)

[Sorting Data](#)

[Working with Generics](#)

[Summary](#)

[Exam Essentials](#)

[Review Questions](#)

[Chapter 15: Functional Programming](#)

[Working with Built-in Functional Interfaces](#)

[Returning an *Optional*](#)

[Using Streams](#)

[Working with Primitive Streams](#)

[Working with Advanced Stream Pipeline](#)

[Concepts](#)

[Summary](#)

[Exam Essentials](#)

[Review Questions](#)

[Chapter 16: Exceptions, Assertions, and Localization](#)

[Reviewing Exceptions](#)

[Creating Custom Exceptions](#)

[Automating Resource Management](#)

[Declaring Assertions](#)

[Working with Dates and Times](#)

[Supporting Internationalization and Localization](#)

[Loading Properties with Resource Bundles](#)

[Summary](#)

[Exam Essentials](#)

[Review Questions](#)

[Chapter 17: Modular Applications](#)

[Reviewing Module Directives](#)

[Comparing Types of Modules](#)

[Analyzing JDK Dependencies](#)

[Migrating an Application](#)

[Creating a Service](#)

[Summary](#)

[Exam Essentials](#)

[Review Questions](#)

[Chapter 18: Concurrency](#)

[Introducing Threads](#)

[Creating Threads with the Concurrency API](#)

[Writing Thread-Safe Code](#)

[Using Concurrent Collections](#)

[Identifying Threading Problems](#)

[Working with Parallel Streams](#)

[Summary](#)

[Exam Essentials](#)

[Review Questions](#)

[Chapter 19: I/O](#)

[Understanding Files and Directories](#)

[Introducing I/O Streams](#)

[Common I/O Stream Operations](#)

[Working with I/O Stream Classes](#)

[Interacting with Users](#)

[Summary](#)

[Exam Essentials](#)

[Review Questions](#)

Chapter 20: NIO.2

[Introducing NIO.2](#)

[Interacting with Paths](#)

[Operating on Files and Directories](#)

[Managing File Attributes](#)

[Applying Functional Programming](#)

[Comparing Legacy java.io.File and NIO.2 Methods](#)

[Summary](#)

[Exam Essentials](#)

[Review Questions](#)

Chapter 21: JDBC

[Introducing Relational Databases and SQL](#)

[Introducing the Interfaces of JDBC](#)

[Connecting to a Database](#)

[Working with a *PreparedStatement*](#)

[Getting Data from a ResultSet](#)

[Calling a *CallableStatement*](#)

[Closing Database Resources](#)

[Summary](#)

[Exam Essentials](#)

[Review Questions](#)

Chapter 22: Security

[Designing a Secure Object](#)

[Introducing Injection and Input Validation](#)

[Working with Confidential Information](#)

[Serializing and Deserializing Objects](#)

[Constructing Sensitive Objects](#)

[Preventing Denial of Service Attacks](#)

[Summary](#)

[Exam Essentials](#)

[Review Questions](#)

[Appendix: Answers to Review Questions](#)

[Chapter 1: Welcome to Java](#)

[Chapter 2: Java Building Blocks](#)

[Chapter 3: Operators](#)

[Chapter 4: Making Decisions](#)

[Chapter 5: Core Java APIs](#)

[Chapter 6: Lambdas and Functional Interfaces](#)

[Chapter 7: Methods and Encapsulation](#)

[Chapter 8: Class Design](#)

[Chapter 9: Advanced Class Design](#)

[Chapter 10: Exceptions](#)

[Chapter 11: Modules](#)

[Chapter 12: Java Fundamentals](#)

[Chapter 13: Annotations](#)

[Chapter 14: Generics and Collections](#)

[Chapter 15: Functional Programming](#)

[Chapter 16: Exceptions, Assertions, and
Localization](#)

[Chapter 17: Modular Applications](#)

[Chapter 18: Concurrency](#)

[Chapter 19: I/O](#)

[Chapter 20: NIO.2](#)

[Chapter 21: JDBC](#)

[Chapter 22: Security](#)

[Index](#)

[Online Test Bank](#)

[Register and Access the Online Test Bank](#)

[End User License Agreement](#)

List of Tables

Introduction

[TABLE I.1 Exam information](#)

Chapter 1

[TABLE 1.1 Running programs](#)

[TABLE 1.2 Setup procedure by operating system](#)

[TABLE 1.3 Options you need to know for the exam:](#)
[javac](#)

[TABLE 1.4 Options you need to know for the exam:](#)
[java](#)

[TABLE 1.5 Options you need to know for the exam:](#)
[jar](#)

[TABLE 1.6 Order for declaring a class](#)

Chapter 2

[TABLE 2.1 Primitive types](#)

[TABLE 2.2 Reserved words](#)

[TABLE 2.3 Default initialization values by type](#)

[TABLE 2.4 Tracking scope by block](#)

Chapter 3

[TABLE 3.1 Order of operator precedence](#)

[TABLE 3.2 Unary operators](#)

[TABLE 3.3 Binary arithmetic operators](#)

[TABLE 3.4 Simple assignment operator](#)

[TABLE 3.5 Compound assignment operators](#)

[TABLE 3.6 Equality operators](#)

[TABLE 3.7 Relational operators](#)

[TABLE 3.8 Logical operators](#)

[TABLE 3.9 Short-circuit operators](#)

Chapter 4

[TABLE 4.1 Advanced flow control usage](#)

Chapter 5

[TABLE 5.1 Binary search rules](#)

[TABLE 5.2 Arrays.compare\(\) examples](#)

[TABLE 5.3 Equality vs. comparison vs. mismatch](#)

[TABLE 5.4 Wrapper classes](#)

[TABLE 5.5 Converting from a String](#)

[TABLE 5.6 Array and list conversions](#)

[TABLE 5.7 Common Map methods](#)

Chapter 6

[TABLE 6.1 Valid lambdas](#)

[TABLE 6.2 Invalid lambdas that return boolean](#)

[TABLE 6.3 Basic functional interfaces](#)

[TABLE 6.4 Rules for accessing a variable from a lambda body inside a method](#)

Chapter 7

[TABLE 7.1 Parts of a method declaration](#)

[TABLE 7.2 Access modifiers](#)

[TABLE 7.3 Static vs. instance calls](#)

[TABLE 7.4 The order that Java uses to choose the right overloaded method](#)

[TABLE 7.5 Naming conventions for getters and setters](#)

Chapter 10

[TABLE 10.1 Types of exceptions and errors](#)

[TABLE 10.2 Legal vs. illegal configurations with a traditional *try* statement](#)

[TABLE 10.3 Legal vs. illegal configurations with a try-with-resources statement](#)

Chapter 11

[TABLE 11.1 Options you need to know for using modules with *javac*](#)

[TABLE 11.2 Options you need to know for using modules with *java*](#)

[TABLE 11.3 Access control with modules](#)

[TABLE 11.4 Modes using *jmod*](#)

[TABLE 11.5 Comparing command-line operations](#)

[TABLE 11.6 Options you need to know for the exam: *javac*](#)

[TABLE 11.7 Options you need to know for the exam: *java*](#)

[TABLE 11.8 Options you need to know for the exam: *jar*](#)

[TABLE 11.9 Options you need to know for the exam: *jdeps*](#)

Chapter 12

[TABLE 12.1 Modifiers in nested classes](#)

[TABLE 12.2 Members in nested classes](#)

[TABLE 12.3 Nested class access rules](#)

[TABLE 12.4 Interface member types](#)

[TABLE 12.5 Interface member access](#)

Chapter 13

[TABLE 13.1 Values for the `@Target` annotation](#)

[TABLE 13.2 Values for the `@Retention` annotation](#)

[TABLE 13.3 Annotation-specific annotations](#)

[TABLE 13.4 Common `@SuppressWarnings` values](#)

[TABLE 13.5 Understanding common annotations](#)

[TABLE 13.6 Applying common annotations](#)

Chapter 14

[TABLE 14.1 Functional interfaces used in this chapter](#)

[TABLE 14.2 Method references](#)

[TABLE 14.3 Wrapper classes](#)

[TABLE 14.4 Factory methods to create a List](#)

[TABLE 14.5 List methods](#)

[TABLE 14.6 Queue methods](#)

[TABLE 14.7 Map methods](#)

[TABLE 14.8 Behavior of the `merge\(\)` method](#)

[TABLE 14.9 Java Collections Framework types](#)

[TABLE 14.10 Collection attributes](#)

[TABLE 14.11 Comparison of Comparable and Comparator](#)

[TABLE 14.12 Helper static methods for building a Comparator](#)

[TABLE 14.13 Helper default methods for building a Comparator](#)

[TABLE 14.14 Types of bounds](#)

[TABLE 14.15 Why we need a lower bound](#)

Chapter 15

[TABLE 15.1 Common functional interfaces](#)

[TABLE 15.2 Convenience methods](#)

[TABLE 15.3 Optional instance methods](#)

[TABLE 15.4 Intermediate vs. terminal operations](#)

[TABLE 15.5 Creating a source](#)

[TABLE 15.6 Terminal stream operations](#)

[TABLE 15.7 Common primitive stream methods](#)

[TABLE 15.8 Mapping methods between types of streams](#)

[TABLE 15.9 Function parameters when mapping between types of streams](#)

[TABLE 15.10 Optional types for primitives](#)

[TABLE 15.11 Common functional interfaces for primitives](#)

[TABLE 15.12 Primitive-specific functional interfaces](#)

[TABLE 15.13 Examples of grouping/partitioning collectors](#)

Chapter 16

[TABLE 16.1 Unchecked exceptions](#)

[TABLE 16.2 Checked exceptions](#)

[TABLE 16.3 Assertion applications](#)

[TABLE 16.4 Date and time types](#)

[TABLE 16.5 Common date/time symbols](#)

[TABLE 16.6 Supported date/time symbols](#)

[TABLE 16.7 Factory methods to get aNumberFormat](#)

[TABLE 16.8DecimalFormat symbols](#)

[TABLE 16.9 Factory methods to get
aDateTimeFormatter](#)

[TABLE 16.10 Locale.Category values](#)

[TABLE 16.11 Picking a resource bundle for
French/France with default locale E...](#)

[TABLE 16.12 Selecting resource bundle properties](#)

Chapter 17

[TABLE 17.1 Common module directives](#)

[TABLE 17.2 Practicing with automatic module
names](#)

[TABLE 17.3 Properties of modules types](#)

[TABLE 17.4 Common modules](#)

[TABLE 17.5 Java modules prefixed with java](#)

[TABLE 17.6 Java modules prefixed with jdk](#)

[TABLE 17.7 Comparing migration strategies](#)

[TABLE 17.8 Reviewing services](#)

Chapter 18

[TABLE 18.1 ExecutorService methods](#)

[TABLE 18.2 Future methods](#)

[TABLE 18.3 `TimeUnit` values](#)

[TABLE 18.4 `ScheduledExecutorService` methods](#)

[TABLE 18.5 Executors factory methods](#)

[TABLE 18.6 Atomic classes](#)

[TABLE 18.7 Common atomic methods](#)

[TABLE 18.8 Lock methods](#)

[TABLE 18.9 Concurrent collection classes](#)

[TABLE 18.10 `BlockingQueue` waiting methods](#)

[TABLE 18.11 Synchronized collections methods](#)

Chapter 19

[TABLE 19.1 Commonly used `java.io.File` methods](#)

[TABLE 19.2 The `java.io` abstract stream base classes](#)

[TABLE 19.3 The `java.io` concrete stream classes](#)

[TABLE 19.4 Common I/O stream methods](#)

[TABLE 19.5 Common print stream `format\(\)` symbols](#)

Chapter 20

[TABLE 20.1 File system symbols](#)

[TABLE 20.2 Common NIO.2 method arguments](#)

[TABLE 20.3 Path methods](#)

[TABLE 20.4 *Files* methods](#)

[TABLE 20.5 The attributes and view types](#)

[TABLE 20.6 Walking a directory with a cycle using breadth-first search](#)

[TABLE 20.7 Comparison of *java.io.File* and NIO.2 methods](#)

Chapter 21

[TABLE 21.1 CRUD operations](#)

[TABLE 21.2 SQL runnable by the execute method](#)

[TABLE 21.3 Return types of execute methods](#)

[TABLE 21.4 PreparedStatement methods](#)

[TABLE 21.5 ResultSet get methods](#)

[TABLE 21.6 Sample stored procedures](#)

[TABLE 21.7 Stored procedure parameter types](#)

Chapter 22

[TABLE 22.1 Types of confidential data](#)

[TABLE 22.2 Methods for serialization and deserialization](#)

List of Illustrations

Introduction

[FIGURE I.1 Past and current Java certifications](#)

[FIGURE I.2 Latest Java certification exams](#)

[FIGURE I.3 Exam prerequisites](#)

Chapter 1

[FIGURE 1.1 Compiling with packages](#)

[FIGURE 1.2 Compiling with packages and directories](#)

Chapter 2

[FIGURE 2.1 An object in memory can be accessed only via a reference.](#)

[FIGURE 2.2 Your drawing after line 5](#)

[FIGURE 2.3 Your drawing after line 7](#)

Chapter 3

[FIGURE 3.1 The logical truth tables for &, |, and ^](#)

Chapter 4

[FIGURE 4.1 The structure of an if statement](#)

[FIGURE 4.2 The structure of an else statement](#)

[FIGURE 4.3 The structure of a switch statement](#)

[FIGURE 4.4 The structure of a while statement](#)

[FIGURE 4.5 The structure of a do/while statement](#)

[FIGURE 4.6 The structure of a basic for loop](#)

[FIGURE 4.7 The structure of an enhanced for-each loop](#)

[FIGURE 4.8 The structure of a break statement](#)

[FIGURE 4.9 The structure of a continue statement](#)

Chapter 5

[FIGURE 5.1 Indexing for a string](#)

[FIGURE 5.2 Indexes for a substring](#)

[FIGURE 5.3 The basic structure of an array](#)

[FIGURE 5.4 An empty array](#)

[FIGURE 5.5 An initialized array](#)

[FIGURE 5.6 An array pointing to strings](#)

[FIGURE 5.7 A sparsely populated multidimensional array](#)

[FIGURE 5.8 An asymmetric multidimensional array](#)

[FIGURE 5.9 Example of a Set](#)

[FIGURE 5.10 Example of a Map](#)

Chapter 6

[FIGURE 6.1 Lambda syntax omitting optional parts](#)

[FIGURE 6.2 Lambda syntax, including optional parts](#)

Chapter 7

[FIGURE 7.1 Method declaration](#)

[FIGURE 7.2 Classes used to show private and default access](#)

[FIGURE 7.3 Classes used to show protected access](#)

[FIGURE 7.4 Copying a reference with pass-by-value](#)

Chapter 8

[FIGURE 8.1 Types of inheritance](#)

[FIGURE 8.2 Java object inheritance](#)

[FIGURE 8.3 Defining and extending a class](#)

[FIGURE 8.4 Object vs. reference](#)

Chapter 9

[FIGURE 9.1 Defining an interface](#)

[FIGURE 9.2 Implementing an interface](#)

[FIGURE 9.3 Interface Inheritance](#)

Chapter 10

[FIGURE 10.1 Categories of exception](#)

[FIGURE 10.2 The syntax of a try statement](#)

[FIGURE 10.3 The syntax of a multi-catch block](#)

[FIGURE 10.4 The syntax of a try statement with finally](#)

[FIGURE 10.5 The syntax of a basic try-with-resources](#)

[FIGURE 10.6 The syntax of try-with-resources including catch/finally](#)

[FIGURE 10.7 A method stack](#)

Chapter 11

[FIGURE 11.1 Design of a modular system](#)

[FIGURE 11.2 Looking inside a module](#)

[FIGURE 11.3 Contents of zoo.animal.feeding](#)

[FIGURE 11.4 Module zoo.animal.feeding directory structure](#)

[FIGURE 11.5 Running a module using java](#)

[FIGURE 11.6 Module zoo.animal.feeding directory structure with class and jar fil...](#)

[FIGURE 11.7 Modules depending on zoo.animal.feeding](#)

[FIGURE 11.8 Contents of zoo.animal.care](#)

[FIGURE 11.9 Module zoo.animal.care directory structure](#)

[FIGURE 11.10 Dependencies for zoo.animal.talks](#)

[FIGURE 11.11 Contents of zoo.animal.talks](#)

[FIGURE 11.12 Contents of zoo.staff](#)

[FIGURE 11.13 Dependencies for zoo.staff](#)

[FIGURE 11.14 Transitive dependency version of our modules](#)

Chapter 12

[FIGURE 12.1 Lambda syntax omitting optional parts](#)

[FIGURE 12.2 Lambda syntax, including optional parts](#)

Chapter 13

[FIGURE 13.1 Annotation declaration](#)

[FIGURE 13.2 Using an annotation](#)

Chapter 14

[FIGURE 14.1 The Collection interface is the root of all collections except m...](#)

[FIGURE 14.2 Example of a List](#)

[FIGURE 14.3 Example of a Set](#)

[FIGURE 14.4 Examples of a HashSet and TreeSet](#)

[FIGURE 14.5 Example of a Queue](#)

[FIGURE 14.6 Working with a queue](#)

[FIGURE 14.7 Example of a Map](#)

Chapter 15

[FIGURE 15.1 Optional](#)

[FIGURE 15.2 Stream pipeline](#)

[FIGURE 15.3 Steps in running a stream pipeline](#)

[FIGURE 15.4 A stream pipeline with a limit](#)

[FIGURE 15.5 Stream pipeline with multiple intermediate operations](#)

Chapter 16

[FIGURE 16.1 The syntax of a try statement](#)

[FIGURE 16.2 The syntax of a try-with-resources statement](#)

[FIGURE 16.3 Categories of exceptions](#)

[FIGURE 16.4 The syntax of assert statements](#)

[FIGURE 16.5 Locale formats](#)

Chapter 17

[FIGURE 17.1 A named module](#)

[FIGURE 17.2 An automatic module](#)

[FIGURE 17.3 An unnamed module](#)

[FIGURE 17.4 Determining the order](#)

[FIGURE 17.5 Determining the order when not unique](#)

[FIGURE 17.6 Bottom-up migration](#)

[FIGURE 17.7 Top-down migration](#)

[FIGURE 17.8 First attempt at decomposition](#)

[FIGURE 17.9 Removing the cyclic dependencies](#)

[FIGURE 17.10 Modules in the tour application](#)

Chapter 18

[FIGURE 18.1 Process model](#)

[FIGURE 18.2 ExecutorService life cycle](#)

[FIGURE 18.3 Lack of thread synchronization](#)

[FIGURE 18.4 Thread synchronization using atomic operations](#)

[FIGURE 18.5 Race condition on user creation](#)

Chapter 19

[FIGURE 19.1 Directory and file hierarchy](#)

[FIGURE 19.2 Visual representation of a stream](#)

[FIGURE 19.3 Serialization process](#)

[FIGURE 19.4 Diagram of I/O stream classes](#)

Chapter 20

[FIGURE 20.1 File system with a symbolic link](#)

[FIGURE 20.2 NIO.2 class and interface relationships](#)

[FIGURE 20.3 Relative paths using path symbols](#)

[FIGURE 20.4 Comparing file uniqueness](#)

[FIGURE 20.5 File and directory as a tree structure](#)

[FIGURE 20.6 File system with cycle](#)

Chapter 21

[FIGURE 21.1 Tables in our relational database](#)

[FIGURE 21.2 Key JDBC interfaces](#)

[FIGURE 21.3 The JDBC URL format](#)

[FIGURE 21.4 Types of statements](#)

[FIGURE 21.5 The ResultSet cursor](#)

Chapter 22

[FIGURE 22.1 Cloneable logic](#)

[FIGURE 22.2 Hours table](#)

[FIGURE 22.3 Directory structure](#)

[FIGURE 22.4 Writing and reading an employee](#)

OCP Oracle[®] Certified Professional Java[®] SE 11 Developer

**Complete Study Guide Exam 1Z0-819
and upgrade Exam 1Z0-817**



Jeanne Boyarsky

Scott Selikoff



Copyright © 2020 by John Wiley & Sons, Inc., Indianapolis, Indiana

Published simultaneously in Canada and the United Kingdom

ISBN: 978-1-119-61913-0

ISBN: 978-1-119-61915-4 (ebk.)

ISBN: 978-1-119-61914-7 (ebk.)

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, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at

<http://www.wiley.com/go/permissions>.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services or to obtain technical support, please contact our Customer Care Department within the U.S. at (877) 762-2974, outside the U.S. at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2020938721

TRADEMARKS: Wiley, the Wiley logo, and the Sybex logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. Oracle and Java are registered trademarks of Oracle, Inc. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

*Happy 20th anniversary to NYC FIRST and StuyPulse
FRC Team 694.*

—Jeanne

*For my daughter, Olivia, your determination and strength
of heart are one of a kind. Your smile brightens even the
darkest days. May your life be filled with happiness and
love.*

—Scott

Acknowledgments

Jeanne and Scott would like to thank numerous individuals for their contribution to this book. Thank you to Kathryn Duggan for guiding us through the process and making the book better in so many ways. Thank you to Janeice DelVecchio for being our technical editor as we wrote this book. Janeice pointed out many subtle errors in addition to the big ones. And thank you to Elena Felder for being our technical proofreader and finding the errors that we managed to sneak by Janeice. This book also wouldn't be possible without many people at Wiley, including Kenyon Brown, Pete Gaughan, Christine O'Connor, Barath Kumar Rajasekaran, Kim Wimpsett, Johnna VanHoose Dinse, and so many others.

Jeanne would personally like to thank Chris Kreussling for knowing more than a decade ago that she would someday write a book. He was a great mentor for many years and definitely shaped her career. Sibon Barman was helpful in getting feedback on the modules chapter, and Susanta Chattopadhyay provided real-life use cases for both service locator and serialization. Stuart Dabbs Halloway's 2001 book provided examples of `serialPeristentFields`. Scott was a great co-author, improving everything Jeanne wrote while writing his own chapters. A big thank you to everyone at `CodeRanch.com` who asked and responded to questions and comments about our books. Finally, Jeanne would like to thank all of the new programmers at `CodeRanch.com` and FIRST robotics teams FRC 694, FTC 310, and FTC 479 for the constant reminders of how new programmers think.

Scott could not have reached this point without his wife, Patti, and family, whose love and support makes this book possible. He would like to thank his twin daughters, Olivia