ORACLE

Digital content includes:

- 170 practice exam questions
- Fully customizable test engine

Java™

OCP Java SE 8 Programmer II Exam Guide

(Exam 1Z0-809)

Complete Exam Preparation

ORACLE Certified Professional Java SE 8 Programmer

Kathy Sierra, SCJP
Bert Bates, SCJP, OCA, OCP
Elisabeth Robson

Oracle Press™

ORACLE®

Oracle Press™

OCP Java™ SE 8 Programmer II Exam Guide

(Exam 1Z0-809)

Kathy Sierra
Bert Bates
Elisabeth Robson

Mc Graw Hill Education

New York Chicago San Francisco
Athens London Madrid
Mexico City Milan New Delhi
Singapore Sydney Toronto

ISBN: 978-1-26-011737-0
MHID: 1-26-011737-5

The material in this eBook also appears in the print version of this title: ISBN: 978-1-26-011738-7, MHID: 1-26-011738-3.

eBook conversion by codeMantra
Version 1.0

ABOUT THE CONTRIBUTORS

About the Authors

Kathy Sierra was a lead developer for the SCJP exam for Java 5 and Java 6. Kathy worked as a Sun "master trainer," and in 1997, founded JavaRanch.com, the world's largest Java community website. Her bestselling Java books have won multiple *Software Development Magazine* awards, and she is a founding member of Oracle's Java Champions program.

These days, Kathy is developing advanced training programs in a variety of domains (from horsemanship to computer programming), but the thread that ties all of her projects together is helping learners reduce cognitive load.

Bert Bates was a lead developer for many of Sun's Java certification exams, including the SCJP for Java 5 and Java 6. Bert was also one of the lead developers for Oracle's OCA 7 and OCP 7 exams and a contributor to the OCP 8 exam. He is a forum moderator on JavaRanch.com and has been developing software for more than 30 years (argh!). Bert is the co-author of several best-selling Java books, and he's a founding member of Oracle's Java Champions program. Now that the book is done, Bert plans to go whack a few tennis balls around and once again start riding his beautiful Icelandic horse, Eyrraros fra Gufudal-Fremri.

Elisabeth Robson has an MSc in Computer Science and was a software programmer and engineering manager at The Walt Disney Company for many years. Since 2012 she has been a freelance writer and instructor. She produces online training and has written four best-selling books, including *Head First Design Patterns* (O'Reilly).

About the Technical Review Team

This is the fifth edition of the book that we've cooked up. The first version

we worked on was for Java 2. Then we updated the book for the SCJP 5, again for the SCJP 6, then for the OCA 7 and OCP 7 exams, and now for the OCA 8 and OCP 8 exams. Every step of the way, we were unbelievably fortunate to have fantastic JavaRanch.com-centric technical review teams at our sides. Over the course of the last 15 years, we've been "evolving" the book more than rewriting it. Many sections from our original work on the Java 2 book are still intact. On the following pages, we'd like to acknowledge the members of the various technical review teams who have saved our bacon over the years.

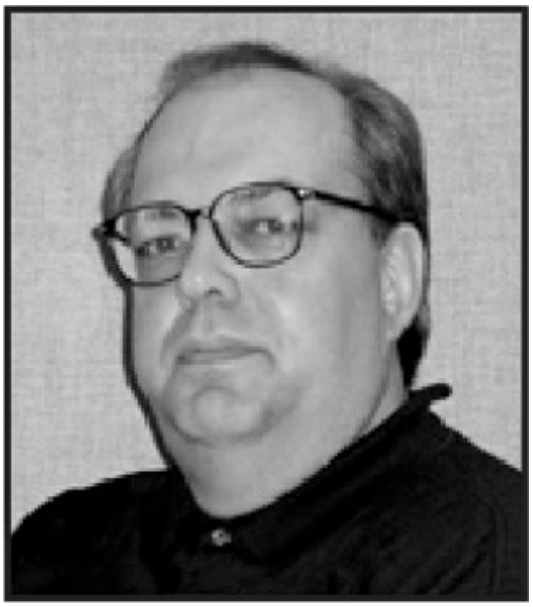
About the Java 2 Technical Review Team

Johannes de Jong has been the leader of our technical review teams forever and ever. (He has more patience than any three people we know.) For the Java 2 book, he led our biggest team ever. Our sincere thanks go out to the following volunteers who were knowledgeable, diligent, patient, and picky, picky, picky!

Rob Ross, Nicholas Cheung, Jane Griscti, Ilja Preuss, Vincent Brabant, Kudret Serin, Bill Seipel, Jing Yi, Ginu Jacob George, Radiya, LuAnn Mazza, Anshu Mishra, Anandhi Navaneethakrishnan, Didier Varon, Mary McCartney, Harsha Pherwani, Abhishek Misra, and Suman Das.

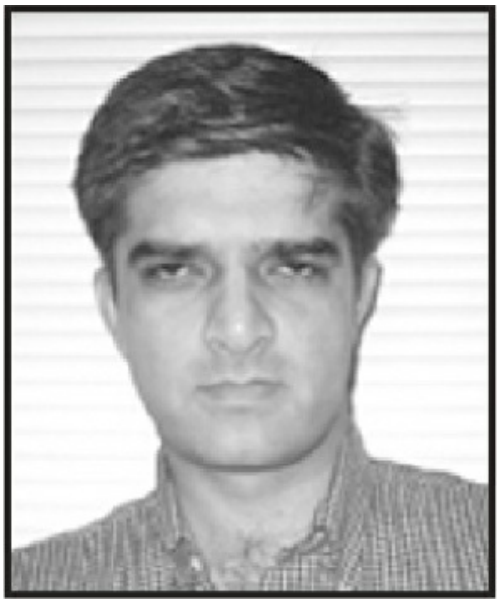
About the SCJP 5 Technical Review Team

Andrew



Bill M.

Burk



Devender

Gian

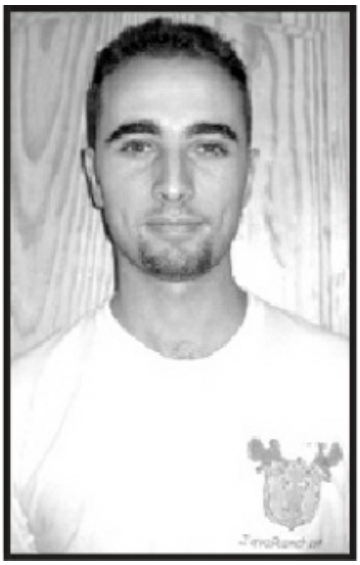
Jef

Jeoren

Jim

Johannes

Kristin



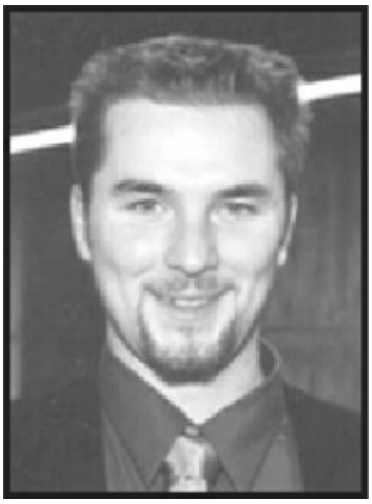
Marcelo

Marilyn

Mark

Mikalai

Seema



Valentin

We don’t know who burned the most midnight oil, but we can (and did) count everybody’s edits—so in order of most edits made, we proudly present our Superstars.

Our top honors go to **Kristin Stromberg**—every time you see a semicolon used correctly, tip your hat to Kristin. Next up is **Burk Hufnagel** who fixed more code than we care to admit. **Bill Mietelski** and **Gian Franco Casula** caught every kind of error we threw at them—awesome job, guys! **Devender Thareja** made sure we didn’t use too much slang, and **Mark Spritzler** kept the humor coming. **Mikalai Zaikin** and **Seema Manivannan** made great catches every step of the way, and **Marilyn de Queiroz** and **Valentin Crettaz** both put in another stellar performance (saving our butts yet again).

Marcelo Ortega, **Jef Cumps** (another veteran), **Andrew Monkhouse**, and **Jeroen Sterken** rounded out our crew of Superstars—thanks to you all. **Jim Yingst** was a member of the Sun exam creation team, and he helped us write and review some of the twistier questions in the book (bwa-ha-ha-ha).

As always, every time you read a clean page, thank our reviewers, and if you do catch an error, it’s most certainly because your authors messed up. And oh, one last thanks to **Johannes**. You rule, dude!

About the SCJP 6 Technical Review Team

Fred

Marc P.

Marc W.

Mikalai

Christophe

Since the upgrade to the Java 6 exam was like a small surgical strike we decided that the technical review team for this update to the book needed to be similarly fashioned. To that end, we hand-picked an elite crew of JavaRanch's top gurus to perform the review for the Java 6 exam.

Our endless gratitude goes to **Mikalai Zaikin**. Mikalai played a huge role in the Java 5 book, and he returned to help us out again for this Java 6 edition. We need to thank Volha, Anastasia, and Daria for letting us borrow Mikalai. His comments and edits helped us make huge improvements to the book. Thanks, Mikalai!

Marc Peabody gets special kudos for helping us out on a double header! In addition to helping us with Sun's new SCWCD exam, Marc pitched in with a great set of edits for this book—you saved our bacon this winter, Marc! (BTW, we didn't learn until late in the game that Marc, Bryan Basham, and Bert all share a passion for ultimate Frisbee!)

Like several of our reviewers, not only does **Fred Rosenberger** volunteer copious amounts of his time moderating at JavaRanch, he also found time to help us out with this book. Stacey and Olivia, you have our thanks for loaning us Fred for a while.

Marc Weber moderates at some of JavaRanch's busiest forums. Marc knows his stuff and uncovered some really sneaky problems that were buried in the book. While we really appreciate Marc's help, we need to warn you all to watch out—he's got a Phaser!

Finally, we send our thanks to **Christophe Verre**—if we can find him. It

appears that Christophe performs his JavaRanch moderation duties from various locations around the globe, including France, Wales, and most recently Tokyo. On more than one occasion Christophe protected us from our own lack of organization. Thanks for your patience, Christophe! It's important to know that these guys all donated their reviewer honorariums to JavaRanch! The JavaRanch community is in your debt.

The OCA 7 and OCP 7 Team

Contributing Authors

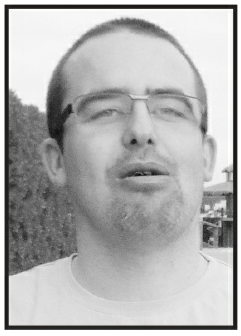
Tom

Jeanne

The OCA 7 exam is primarily a useful repackaging of some of the objectives from the SCJP 6 exam. On the other hand, the OCP 7 exam introduced a vast array of brand-new topics. We enlisted several talented Java gurus to help us cover some of the new topics on the OCP 7 exam. Thanks and kudos to **Tom McGinn** for his fantastic work in creating the massive JDBC chapter. Several reviewers told us that Tom did an amazing job channeling the informal tone

we use throughout the book. Next, thanks to **Jeanne Boyarsky**. Jeanne was truly a renaissance woman on this project. She contributed to several OCP chapters; she wrote some questions for the master exams; she performed some project management activities; and as if that wasn't enough, she was one of our most energetic technical reviewers. Jeanne, we can't thank you enough. Our thanks go to **Matt Heimer** for his excellent work on the concurrency chapter. A really tough topic, nicely handled! Finally, **Roel De Nijs** and **Roberto Perillo** made some nice contributions to the book *and* helped out on the technical review team—thanks, guys!

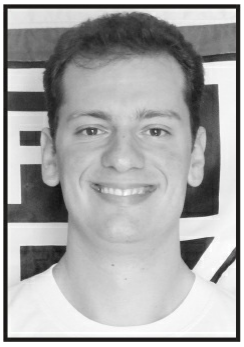
Technical Review Team



Roel

Mikalai

Vijitha



Roberto

Roel, what can we say? Your work as a technical reviewer is unparalleled. Roel caught so many technical errors, it made our heads spin. Between the printed book and all the material on the CD, we estimate that there are over 1,500 pages of "stuff" here. It's huge! Roel grinded through page after page, never lost his focus, and made this book better in countless ways. Thank you, Roel!

In addition to her other contributions, **Jeanne** provided one of the most thorough technical reviews we received. (We think she enlisted her team of killer robots to help her!)

It seems like no K&B book would be complete without help from our old friend **Mikalai Zaikin**. Somehow, between earning 812 different Java certifications, being a husband and father (thanks to **Volha**, **Anastasia**, **Daria**, and **Ivan**), and being a "theoretical fisherman" [sic], Mikalai made substantial contributions to the quality of the book; we're honored that you helped us again, Mikalai.

Next up, we'd like to thank **Vijitha Kumara**, JavaRanch moderator and tech reviewer extraordinaire. We had many reviewers help out during the

long course of writing this book, but Vijitha was one of the few who stuck with us from Chapter 1 all the way through the master exams and on to Chapter 15. Vijitha, thank you for your help and persistence!

Finally, thanks to the rest of our review team: **Roberto Perillo** (who also wrote some killer exam questions), **Jim Yingst** (was this your fourth time?), other repeat offenders: **Fred Rosenberger**, **Christophe Verre**, **Devaka Cooray**, **Marc Peabody**, and newcomer **Amit Ghorpade**—thanks, guys!

The OCP 8 Team

Approximately two-thirds of the OCP 8 exam's objectives are the same as the OCP 7 exam, and about one-third are new topics focused on all of the amazing new features introduced in Java 8. This time around, our entire review team was composed of veterans. In addition, it's about time that we give thanks to all of the folks on JavaRanch who took time to share errata with us. Because of our amazing xiireview team and the generosity of JavaRanchers (we know "CodeRanch"), this book was improved immensely.

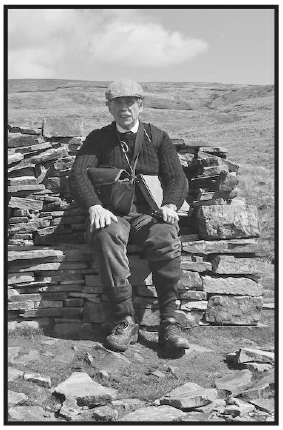
Between the printed book and the two final mock exams, this book has well over 900 pages of material. One way of looking at this book is that every page is a series of factual claims. It's hard to estimate, but perhaps there are 20 such claims on every page. That means we're making about 18,000 factual claims in this book! This makes us authors feel better about saying that our various reviewers found hundreds of errors or areas in which our explanations could have been better. Hundreds out of 18,000 isn't too bad is it?

When we're in the mode of incorporating our reviewers' feedback, we often have to make tough choices. We know that our job isn't to restate the Java Language Spec. We have to make many decisions about how deeply to go into the topics on the exam. You should know that our reviewers constantly challenge us to go deeper. That's a good thing. If you feel as though we haven't covered a topic as deeply as you'd like or that we've oversimplified an explanation, the fault almost certainly lies with us authors, not with the reviewers!

With all of that said, it's time to thank the members of our amazing review team individually.

Technical Review Team

Mikalai



Campbell

Paweł

Frits

Roberto

Vijitha

Tim

Mikalai, wow, wow, wow! This is the fourth time (at least?) that Mikalai Zaikin has been one of our reviewers. Mikalai is a real expert, and he pushes us and makes us think. Mikalai is first and foremost a family man (hooray!), but he's also a geek, and—not satisfied with being only a Java expert—he also pursues other programming approaches as well. It wouldn't surprise us at all if he was into functional programming and other such wackiness. You all have a huge debt to pay to **Campbell**. Campbell Ritchie is a JavaRanch moderator and another true expert. Campbell is passionate about Java, and his edits really taught us a thing or two. Thanks for all your time Campbell! **Paweł Baczyński**, gave us a TON of good feedback. Paweł, send our thanks to your wife and kids; we appreciate their patience! Our next thanks go to veteran reviewer and JavaRanch moderator **Frits Walraven**. Frits is a published mock-exam-question creator (awesome), husband, father, and serial Java certificate holder. Get some sleep, Frits! Once again, we were honored to have **Roberto Perillo** on our review team. This is at least the third time Roberto has helped out. Given what a thankless job this is, Roberto, we can't thank you enough. Roberto is a dad, hooray, and from what we hear he plays a mean guitar. With over 4000 JavaRanch posts to his credit, moderator **Vijitha Kumara** proved once again to be "in it for the long haul!" In addition to traveling and community service, Vijitha was with us right to the final mock exam. Vijitha, thanks for all your help! Last but not least, our thanks go to yet another JavaRanch moderator **Tim Cooke**. Rumor has it that Tim's cat Polly (an Erlang aficionado!?), was at Tim's side throughout the editing process. This might explain some of the attitude that came through in Tim's edits. Tim focused his energies on editing the new FP-ish additions to the exam. Tim, thanks so much for all of your help!

For Jim, Joe, and Solveig

CONTENTS AT A GLANCE

CONTENTS

2 Object Orientation

ACKNOWLEDGMENTS

Kathy and Bert (and in the cases of McGraw-Hill Education and JavaRanch, Elisabeth) would like to thank the following people:

- All the incredibly hard-working folks at McGraw-Hill Education: Tim Green (who's been putting up with us for 15 years now), LeeAnn Pickrell (and team), Lisa McClain, Jody McKenzie, and Jim Kussow. Thanks for all your help, and for being so responsive, patient, flexible, and professional, and the nicest group of people we could hope to work with.
- All of our friends at Kraftur (and our other horse-related friends) and most especially to Sherry; Steinar; Stina and the girls, Jec, Lucy, Cait, and Jennifer; Leslie and David; Annette and Bruce; Kacie; DJ; Gabrielle; and Mary. Thanks to Pedro and Ely, who can't believe it can take so long to finish a book.
- Some of the software professionals and friends who helped us in the early days: Tom Bender, Peter Loerincs, Craig Matthews, Leonard Coyne, Morgan Porter, and Mike Kavenaugh.
- Dave Gustafson and Marc Hedlund for their continued support, insights, and coaching.
- Our good friend at Oracle, Yvonne Prefontiane.
- The crew at Oracle who worked hard to build these exams: Tom McGinn, Matt Heimer, Mike Williams, Stuart Marks, and Mikalai Zaikin.
- Our old wonderful and talented certification team at Sun Educational Services, primarily the most persistent get-it-done person we know, Evelyn Cartagena.
- Our great friends and gurus, Simon Roberts, Bryan Basham, and

Kathy Collina.

- Stu, Steve, Burt, and Eric for injecting some fun into the process.
- To Eden and Skyler, for being horrified that adults—out of school—would study this hard for an exam.
- To the JavaRanch Trail Boss Paul Wheaton, for running the best Java community site on the Web, and to all the generous and patient JavaRanch moderators.
- To all the past and present Sun Ed Java instructors for helping to make learning Java a fun experience, including (to name only a few) Alan Petersen, Jean Tordella, Georgianna Meagher, Anthony Orapallo, Jacqueline Jones, James Cubeta, Teri Cubeta, Rob Weingruber, John Nyquist, Asok Perumainar, Steve Stelting, Kimberly Bobrow, Keith Ratliff, and the most caring and inspiring Java guy on the planet, Jari Paukku.
- Our furry and feathered friends Eyra, Kara, Draumur, Vafi, Boi, Niki, and Bokeh.
- Finally, to Elisabeth Robson (our amazing new co-author) and Eric Freeman, for your continued inspiration.

PREFACE

This book's primary objective is to help you prepare for and pass Oracle's OCP Java SE 8 Programmer II certification exam.

If you already have an SCJP 6 or OCP 7 certification and want to take an upgrade exam, all of the topics covered in the OCP 7 and SCJP 6 Upgrade exams are covered here as well.

This book follows closely both the breadth and the depth of the real exams. For instance, after reading this book, you probably won't emerge as an NIO.2 guru, but if you study the material and do well on the Self Tests, you'll have a basic understanding of NIO.2, and you'll do well on the exam. After completing this book, you should feel confident that you have thoroughly reviewed all of the objectives that Oracle has established for these exams.

In This Book

This book is organized to optimize your learning of the topics covered by the OCP 8 exam. Whenever possible, we've organized the chapters to parallel the Oracle objectives, but sometimes we'll mix up objectives or partially repeat them in order to present topics in an order better suited to learning the material.

In the Chapters

We've created a set of chapter components that call your attention to important items, reinforce important points, and provide helpful exam-taking hints. Take a look at what you'll find in the chapters:

- Every chapter begins with the **Certification Objectives**—what you

need to know in order to pass the section on the exam dealing with the chapter topic. The Certification Objective headings identify the objectives within the chapter, so you'll always know an objective when you see it!

- **On the Job** callouts discuss practical aspects of certification topics that might not occur on the exam, but that will be useful in the real world.

Exam Watch ***notes call attention to information about, and potential pitfalls in, the exam. Since we were on the team that created these exams, we know what you're about to go through!***

- **Exercises** help you master skills that are likely to be an area of focus on the exam. Don't just read through the exercises; they are hands-on practice that you should be comfortable completing. Learning by doing is an effective way to increase your competency with a product.
- The **Certification Summary** is a succinct review of the chapter and a restatement of salient points regarding the exam.

✓

- The **Two-Minute Drill** at the end of every chapter is a checklist of the main points of the chapter. It can be used for last-minute review.

Q&A

- The **Self Test** offers questions similar to those found on the certification exam, including multiple-choice and pseudo drag-and-drop questions. The answers to these questions, as well as explanations of the answers, can be found at the end of every chapter.

By taking the Self Test after completing each chapter, you'll reinforce what you've learned from that chapter, while becoming familiar with the structure of the exam questions.

INTRODUCTION

Organization

This book is organized in such a way as to serve as an in-depth review for the OCP Java SE 8 Programmer II exam for both experienced Java professionals and those in the early stages of experience with Java technologies. Each chapter covers at least one major aspect of the exam, with an emphasis on the "why" as well as the "how to" of programming in the Java language.

Throughout this book and supplemental digital material, you'll find support for three exams:

- OCP Java SE 8 Programmer II
- Upgrade to Java SE 8 Programmer from Java SE 7
- Upgrade to Java SE 8 Programmer from Java SE 6

What This Book Is Not

You will not find a beginner's guide to learning Java in this book. All 900+ pages of this book are dedicated solely to helping you pass the exams. Since you cannot take this exam without another Java certification under your belt, in this book, we assume you have a working knowledge of everything covered in the OCA 8 exam. We do not, however, assume any level of prior knowledge of the individual topics covered. In other words, for any given topic (driven exclusively by the actual exam objectives), we start with the assumption that you are new to that topic. So we assume you're new to the individual topics, but we assume that you are not new to Java.

We also do not pretend to be both preparing you for the exam and simultaneously making you a complete Java being. This is a certification exam study guide, and it's very clear about its mission. That's not to say that preparing for the exam won't help you become a better Java programmer! On the contrary, even the most experienced Java developers often claim that

having to prepare for the certification exam made them far more knowledgeable and well-rounded programmers than they would have been without the exam-driven studying.

About the Digital Content

You'll receive access to online practice exam software with the equivalent of two 85-question exams for OCP Java SE 8 candidates. The digital content included with the book includes three chapters that complete the coverage necessary for the OCP 7 and SCJP 6 upgrade certifications. We've also included an Online Appendix, "Creating Streams from Files Methods," describing additional methods for processing files and directories.

Please see the Appendix for details on accessing the digital content and online practice exams.

Some Pointers

Once you've finished reading this book, set aside some time to do a thorough review. You might want to return to the book several times and make use of all the methods it offers for reviewing the material:

1. *Re-read all the Two-Minute Drills, or have someone quiz you*. You also can use the drills as a way to do a quick cram before the exam. You might want to make some flash cards out of 3×5 index cards that have the Two-Minute Drill material on them.
2. *Re-read all the Exam Watch notes*. Remember that these notes are written by authors who helped create the exam. They know what you should expect—and what you should be on the lookout for.
3. *Re-take the Self Tests*. Taking the tests right after you've read the chapter is a good idea because the questions help reinforce what you've just learned. However, it's an even better idea to go back later and do all the questions in the book in one sitting. Pretend that you're taking the live exam. (Whenever you take the Self Tests, mark your answers on a separate piece of paper. That way, you can run through the questions as many times as you need to until you feel comfortable with the material.)

4. *Complete the exercises*. The exercises are designed to cover exam topics, and there's no better way to get to know this material than by practicing. Be sure you understand why you are performing each step in each exercise. If there is something you are not clear on, re-read that section in the chapter.
5. *Write lots of Java code*. We'll repeat this advice several times. When we wrote this book, we wrote hundreds of small Java programs to help us do our research. We have heard from hundreds of candidates who have passed the exam, and in almost every case, the candidates who scored extremely well on the exam wrote lots of code during their studies. Experiment with the code samples in the book, create horrendous lists of compiler errors—put away your IDE, crank up the command line, and write code!

A Note About the Certification Objectives

Some of the OCP 8 Certification Objectives are not exactly the most clearly written objectives. You may, as we sometimes did, find yourself squinting a little sideways at an objective when attempting to parse what exactly the objective is about and what the objective might be leaving unsaid. We've done our best to cover all the topics we *think* the objective writers meant to include. And, as a reminder, we do try to teach about 115 percent of what we think you'll need to know for the exam, just to cover all our (and your) bases. That said, there are some gray areas here and there. So look carefully at each objective, and if you think there's something we missed, go study that on your own (and do let us know!).

Introduction to the Material in the Book

The OCP 8 exam is considered one of the hardest in the IT industry, and we can tell you from experience that a large chunk of exam candidates goes in to the test unprepared. As programmers, we tend to learn only what we need to complete our current project, given the insane deadlines we're usually under.

But this exam attempts to prove your complete understanding of the Java language, not just the parts of it you've become familiar with in your work.

Experience alone will rarely get you through this exam with a passing mark, because even the things you think you know might work just a little

differently than you imagined. It isn't enough to be able to get your code to work correctly; you must understand the core fundamentals in a deep way and with enough breadth to cover virtually anything that could crop up in the course of using the language.

Who Cares About Certification?

Employers do. Headhunters do. Programmers do. Passing this exam proves three important things to a current or prospective employer: you're smart; you know how to study and prepare for a challenging test; and, most of all, you know the Java language. If an employer has a choice between a candidate who has passed xxxvithe exam and one who hasn't, the employer knows that the certified programmer does not have to take time to learn the Java language.

But does it mean that you can actually develop software in Java? Not necessarily, but it's a good head start.

Taking the Programmer's Exam

In a perfect world, you would be assessed for your true knowledge of a subject, not simply how you respond to a series of test questions. But life isn't perfect, and it just isn't practical to evaluate everyone's knowledge on a one-to-one basis.

For the majority of its certifications, Oracle evaluates candidates using a computer-based testing service operated by Pearson VUE. To discourage simple memorization, Oracle exams present a potentially different set of questions to different candidates. In the development of the exam, hundreds of questions are compiled and refined using beta testers. From this large collection, questions are pulled together from each objective and assembled into many different versions of the exam.

Each Oracle exam has a specific number of questions, and the test's duration is designed to be generous. The time remaining is always displayed in the corner of the testing screen. If time expires during an exam, the test terminates, and incomplete answers are counted as incorrect.

exam Watch

Many experienced test-takers do not go back and change answers unless they have a good reason to do so. Only change an answer when you feel you may have misread or misinterpreted the question the first time. Nervousness may make you secondguess every answer and talk yourself out of a correct one.

Question Format

Oracle's Java exams pose questions in multiple-choice format.

Multiple-Choice Questions

In earlier versions of the exam, when you encountered a multiple-choice question, you were not told how many answers were correct, but with each version of the exam, the questions have become more difficult, so today, each multiple-choice question tells you how many answers to choose. The Self Test questions at the end of each chapter closely match the format, wording, and difficulty of the real exam questions, with two exceptions:

- Whenever we can, our questions will *not* tell you how many correct answers exist (we will say "Choose all that apply"). We do this to help you master the material. Some savvy test-takers can eliminate wrong answers when the number of correct answers is known. It's also possible, if you know how many answers are correct, to choose the most plausible answers. Our job is to toughen you up for the real exam!
- The real exam typically numbers lines of code in a question. Sometimes we do not number lines of code—mostly so that we have the space to add comments at key places. On the real exam, when a code listing starts with line 1, it means that you're looking at an entire source file. If a code listing starts at a line number greater than 1, that means you're looking at a partial source file. When looking at a partial source file, assume that the code you can't see is correct. (For

instance, unless explicitly stated, you can assume that a partial source file will have the correct import and package statements.)

When you find yourself stumped answering multiple-choice questions, use your scratch paper (or whiteboard) to write down the two or three answers you consider the strongest, then underline the answer you feel is most likely correct. Here is an example of what your scratch paper might look like when you've gone through the test once:

- ***21. B or C***
- ***33. A or C***

This is extremely helpful when you mark the question and continue on. You can then return to the question and immediately pick up your thought process where you left off. Use this technique to avoid having to re-read and rethink questions. You will also need to use your scratch paper during complex, text-based scenario questions to create visual images to better understand the question. This technique is especially helpful if you are a visual learner.

Tips on Taking the Exam

The number of questions and passing percentages for every exam are subject to change. Always check with Oracle before taking the exam, at www.Oracle.com.

You are allowed to answer questions in any order, and you can go back and check your answers after you've gone through the test. There are no penalties for wrong answers, so it's better to at least attempt an answer than to not give one at all.

A good strategy for taking the exam is to go through once and answer all the questions that come to you quickly. You can then go back and do the others. Answering one question might jog your memory for how to answer a previous one.

Be very careful on the code examples. Check for syntax errors first: count curly braces, semicolons, and parentheses and then make sure there are as many left ones as right ones. Look for capitalization errors and other such syntax problems before trying to figure out what the code does.

Many of the questions on the exam will hinge on subtleties of syntax. You will need to have a thorough knowledge of the Java language in order to succeed.

This brings us to another issue that some candidates have reported. The testing center is supposed to provide you with sufficient writing implements so you can work problems out "on paper." In some cases, the centers have provided inadequate markers and dry-erase boards that are too small and cumbersome to use effectively. We recommend that you call ahead and verify that you will be supplied with a sufficiently large whiteboard, sufficiently fine-tipped markers, and a good eraser. What we'd really like to encourage is for everyone to complain to Oracle and Pearson VUE and have them provide actual pencils and at least several sheets of blank paper.

Tips on Studying for the Exam

First and foremost, give yourself plenty of time to study. Java is a complex programming language, and you can't expect to cram what you need to know into a single study session. It is a field best learned over time, by studying a subject and then applying your knowledge. Build yourself a study schedule and stick to it, but be reasonable about the pressure you put on yourself, especially if you're studying in addition to your regular duties at work.

One easy technique to use in studying for certification exams is the 15-minutes-per-day effort. Simply study for a minimum of 15 minutes every day. It is a small but significant commitment. If you have a day where you just can't focus, then give up at 15 minutes. If you have a day where it flows completely for you, study longer. As long as you have more of the "flow days," your chances of succeeding are excellent.

We strongly recommend you use flash cards when preparing for the programmer's exams. A flash card is simply a 3×5 or 4×6 index card with a question on the front and the answer on the back. You construct these cards yourself as you go through a chapter, capturing any topic you think might need more memorization or practice time. You can drill yourself with them by reading the question, thinking through the answer, and then turning the

card over to see if you're correct. Or you can get another person to help you by holding up the card with the question facing you and then verifying your answer. Most of our students have found these to be tremendously helpful, especially because they're so portable that while you're in study mode, you can take them everywhere. Best not to use them while driving, though, except at red lights. We've taken ours everywhere—the doctor's office, restaurants, theaters, you name it.

Certification study groups are another excellent resource, and you won't find a larger or more willing community than on the JavaRanch.com Big Moose Saloon certification forums. If you have a question from this book, or any other mock exam question you may have stumbled upon, posting a question in a certification forum will get you an answer in nearly all cases within a day—usually, within a few hours. You'll find us (the authors) there several times a week, helping those just starting out on their exam preparation journey. (You won't actually think of it as anything as pleasant sounding as a "journey" by the time you're ready to take the exam.)

Finally, we recommend that you write a lot of little Java programs! During the course of writing this book, we wrote hundreds of small programs, and if you listen to what the most successful candidates say (you know, those guys who got 98 percent), they almost always report that they wrote a lot of code.

Scheduling Your Exam

You can purchase your exam voucher from Oracle or Pearson VUE. Visit Oracle.com (follow the training/certification links) or visit PearsonVue.com for exam scheduling details and locations of test centers.

Arriving at the Exam

As with any test, you'll be tempted to cram the night before. Resist that temptation. You should know the material by this point, and if you're groggy in the morning, you won't remember what you studied anyway. Get a good night's sleep.

Arrive early for your exam; it gives you time to relax and review key facts. Take the opportunity to review your notes. If you get burned out on studying, you can usually start your exam a few minutes early. We don't recommend arriving late. Your test could be cancelled, or you might not have

enough time to complete the exam.

When you arrive at the testing center, you'll need to provide current, valid photo identification. Visit PearsonVue.com for details on the ID requirements. They just want to be sure that you don't send your brilliant Java guru next-door neighbor who you've paid to take the exam for you.

Aside from a brain full of facts, you don't need to bring anything else to the exam room. In fact, your brain is about all you're allowed to take into the exam!

All the tests are closed book, meaning you don't get to bring any reference materials with you. You're also not allowed to take any notes out of the exam room. The test administrator will provide you with a small marker board. If you're allowed to, we do recommend that you bring a water bottle or a juice bottle (call ahead for details of what's allowed). These exams are long and hard, and your brain functions much better when it's well hydrated. In terms of hydration, the ideal approach is to take frequent, small sips. You should also verify how many "bio-breaks" you'll be allowed to take during the exam!

Leave your pager and telephone in the car or turn them off. They only add stress to the situation, since they are not allowed in the exam room, and can sometimes still be heard if they ring outside of the room. Purses, books, and other materials must be left with the administrator before entering the exam.

Once in the testing room, you'll be briefed on the exam software. You might be asked to complete a survey. The time you spend on the survey is *not* deducted from your actual test time—nor do you get more time if you fill out the survey quickly. Also, remember that the questions you get on the exam will *not* change depending on how you answer the survey questions. Once you're done with the survey, the real clock starts ticking and the fun begins.

The testing software allows you to move forward and backward between questions. Most important, there is a Mark check box on the screen—this will prove to be a critical tool, as explained in the next section.

Test-Taking Techniques

Without a plan of attack, candidates can become overwhelmed by the exam or become sidetracked and run out of time. For the most part, if you are comfortable with the material, the allotted time is more than enough to complete the exam. The trick is to keep the time from slipping away during

any one particular problem.

Your obvious goal is to answer the questions correctly and quickly, but other factors can distract you. Here are some tips for taking the exam more efficiently.

Size Up the Challenge

First, take a quick pass through all the questions in the exam. “Cherry-pick” the easy questions, answering them on the spot. Briefly read each question, noticing the type of question and the subject. As a guideline, try to spend less than 25 percent of your testing time in this pass.

This step lets you assess the scope and complexity of the exam, and it helps you determine how to pace your time. It also gives you an idea of where to find potential answers to some of the questions. Sometimes the wording of one question might lend clues or jog your thoughts for another question.

If you’re not entirely confident in your answer to a question, answer it anyway, but check the Mark box to flag it for later review. In the event that you run out of time, at least you’ve provided a “first guess” answer, rather than leaving it blank.

Second, go back through the entire test, using the insight you gained from the first go-through. For example, if the entire test looks difficult, you’ll know better than to spend more than a minute or two on each question. Create a pacing with small milestones—for example, “I need to answer 10 questions every 15 minutes.”

At this stage, it’s probably a good idea to skip past the time-consuming questions, marking them for the next pass. Try to finish this phase before you’re 50 to 60 percent through the testing time.

Third, go back through all the questions you marked for review, using the Review Marked button in the question review screen. This step includes taking a second look at all the questions you were unsure of in previous passes, as well as tackling the time-consuming ones you deferred until now. Chisel away at this group of questions until you’ve answered them all.

If you’re more comfortable with a previously marked question, unmark the Review Marked button now. Otherwise, leave it marked. Work your way xliithrough the time-consuming questions now, especially those requiring manual calculations. Unmark them when you’re satisfied with the answer.

By the end of this step, you've answered every question in the test, despite having reservations about some of your answers. If you run out of time in the next step, at least you won't lose points for lack of an answer. You're in great shape if you still have 10 to 20 percent of your time remaining.

Review Your Answers

Now you're cruising! You've answered all the questions, and you're ready to do a quality check. Take yet another pass (yes, one more) through the entire test, briefly re-reading each question and your answer.

Carefully look over the questions again to check for "trick" questions. Be particularly wary of those that include a choice of "Does not compile." Be alert for last-minute clues. You're pretty familiar with nearly every question at this point, and you may find a few clues that you missed before.

The Grand Finale

When you're confident with all your answers, finish the exam by submitting it for grading. After you finish your exam, you'll receive an e-mail from Oracle giving you a link to a page where your exam results will be available. As of this writing, you must ask for a hard copy certificate specifically or one will not be sent to you.

Retesting

If you don't pass the exam, don't be discouraged. Try to have a good attitude about the experience, and get ready to try again. Consider yourself a little more educated. You'll know the format of the test a little better, and you'll have a good idea of the difficulty level of the questions you'll get next time around.

If you bounce back quickly, you'll probably remember several of the questions you might have missed. This will help you focus your study efforts in the right area.

Ultimately, remember that Oracle certifications are valuable because they're hard to get. After all, if anyone could get one, what value would it have? In the end, it takes a good attitude and a lot of studying, but you can do it!

OCP 8 Objectives Map

The following table for the OCP Java SE 8 Programmer II Exam describes the objectives and where you will find them in the book.

Oracle Certified Professional Java SE 8 Programmer II (Exam IZ0-804)

(Note: Some of the OCP objectives detailed here are similar to or duplicates of OCA 8 objectives. If you've read our *OCA 8 Java SE 8 Programmer I Exam Guide*, you will recognize some of the material covering those objectives, particularly material in Chapters 1 and 2.)

Official Objective	Exam Guide Coverage
Java Class Design	
Implement encapsulation (1.1)	Chapter 1
Implement inheritance including visibility modifiers and composition (1.2)	Chapter 1
Implement polymorphism (1.3)	Chapter 1
Override the hashcode, equals, and toString methods from Object class (1.4)	Chapter 1
Create and use singleton classes and immutable classes (1.5)	Chapter 1
Develop code that uses the static keyword on initialize blocks, variables, methods, and classes (1.6)	Chapter 1
Advanced Java Class Design	
Develop code that uses abstract classes and methods (2.1)	Chapter 2
Develop code that uses the final keyword (2.2)	Chapter 2
Create inner classes including static inner class, local class, nested class, and anonymous inner class (2.3)	Chapter 2
Use enumerated types including methods, and constructors in an enum type (2.4)	Chapter 2
Develop code that declares, implements and/or extends interfaces and use the @Override annotation (2.5)	Chapter 2
Create and use Lambda expressions (2.6)	Chapters 6, 7, and 8
Generics and Collections	
Create and use a generic class (3.1)	Chapter 6
Create and use ArrayList, TreeSet, TreeMap, and ArrayDeque objects (3.2)	Chapter 6
Use java.util.Comparator and java.lang.Comparable interfaces (3.3)	Chapter 6
Collections Streams and Filters (3.4)	Chapter 6
Iterate using forEach methods of Streams and List (3.5)	Chapters 8 and 9
Describe Stream interface and Stream pipeline (3.6)	Chapter 9
Filter a collection by using lambda expressions (3.7)	Chapter 9
Use method references with Streams (3.8)	Chapter 9

Lambda Built-in Functional Interfaces	
Use the built-in interfaces included in the java.util.function package such as Predicate, Consumer, Function, and Supplier (4.1)	Chapter 8
Develop code that uses primitive versions of functional interfaces (4.2)	Chapter 8
Develop code that uses binary versions of functional interfaces (4.3)	Chapter 8
Develop code that uses the UnaryOperator interface (4.4)	Chapter 8
Java Stream API	
Develop code to extract data from an object using peek() and map() methods including primitive versions of the map() method (5.1)	Chapter 9
Search for data by using search methods of the Stream classes including findFirst, findAny, anyMatch, allMatch, noneMatch (5.2)	Chapter 9
Develop code that uses the Optional class (5.3)	Chapter 9
Develop code that uses Stream data methods and calculation methods (5.4)	Chapter 9
Sort a collection using Stream API (5.5)	Chapter 9
Save results to a collection using the collect method and group/partition data using the Collectors class (5.6)	Chapter 9
Use flatMap() methods in the Stream API (5.7)	Chapter 9
Exceptions and Assertions	
Use try-catch and throw statements (6.1)	Chapter 3
Use catch, multi-catch, and finally clauses (6.2)	Chapter 3
Use Autoclose resources with a try-with-resources statement (6.3)	Chapter 3
Create custom exceptions and Autocloseable resources (6.4)	Chapter 3
Test invariants by using assertions (6.5)	Chapter 3
Use Java SE 8 Date/Time API	
Create and manage date-based and time-based events including a combination of date and time into a single object using LocalDate, LocalTime, LocalDateTIme, Instant, Period, and Duration (7.1)	Chapter 4
Work with dates and times across timezones and manage changes resulting from daylight savings including Format date and times values (7.2)	Chapter 4
Define and create and manage date-based and time-based events using Instant, Period, Duration, and TemporalUnit (7.3)	Chapter 4
Java I/O Fundamentals	
Read and write data from the console (8.1)	Chapter 5
Use BufferedReader, BufferedWriter, File, FileReader, FileWriter, FileInputStream, FileOutputStream, ObjectOutputStream, ObjectInputStream, and PrintWriter in the java.io package (8.2)	Chapter 5

Java File I/O (NIO.2)	
Use Path interface to operate on file and directory paths (9.1)	Chapter 5
Use Files class to check, read, delete, copy, move, manage metadata of a file or directory (9.2)	Chapter 5
Use Stream API with NIO.2 (9.3)	Chapters 5, 9, and Online Appendix
Java Concurrency	
Create worker threads using Runnable, Callable and use an ExecutorService to concurrently execute tasks (10.1)	Chapters 10, 11
Identify potential threading problems among deadlock, starvation, livelock, and race conditions (10.2)	Chapter 10
Use synchronized keyword and java.util.concurrent.atomic package to control the order of thread execution (10.3)	Chapters 10, 11
Use java.util.concurrent collections and classes including CyclicBarrier and CopyOnWriteArrayList (10.4)	Chapter 11
Use parallel Fork/Join Framework (10.5)	Chapter 11
Use parallel Streams including reduction, decomposition, merging processes, pipelines and performance (10.6)	Chapter 11
Building Database Applications with JDBC	
Describe the interfaces that make up the core of the JDBC API including the Driver, Connection, Statement, and ResultSet interfaces and their relationships to provider implementations (11.1)	Chapter 12
Identify the components required to connect to a database using the DriverManager class including the JDBC URL (11.2)	Chapter 12
Submit queries and read results from the database including creating statements, returning result sets, iterating through the results, and properly closing result sets, statements, and connections (11.3)	Chapter 12
Localization	
Read and set the locale using the Locale object (12.1)	Chapter 4
Create and read a Properties file (12.2)	Chapter 4
Build a resource bundle for each locale and load a resource bundle in an application (12.3)	Chapter 4

Taking the Java SE 7 or Java 6 Upgrade Exam

For those of you who have your OCP 7 or SCJP 6 certification and want to take the upgrade exam to get your OCP 8 certification, this book contains everything you'll need to study. To be fair, there is a lot in this book that you won't need to study, but by comparing the upgrade objectives to the Table of Contents, you should be able to determine which chapters apply to you. There are a handful of objectives that do NOT overlap with the OCP 8 exam, and you can find coverage for those objectives in the included digital content. The digital content contains chapters that include coverage for

- Java 6, Objective 1.1, Develop code that uses String objects in the switch statement, binary literals, and numeric literals, including underscores in literals
- Java 7, Objective 6.2, Develop code that uses Java SE 8 I/O improvements, including Files.find(), Files.walk(), and lines() methods

Please refer to the Appendix for details on accessing the additional digital content.