OCA Java SE 7

Programmer I

CERTIFICATION GUIDE

Prepare for the 1Z0-803 exam

Mala Gupta

FOREWORD BY **Jeanne Boyarsky**

MANNING

OCA Java SE 7 Programmer I
Certification Guide

OCA Java SE 7 Programmer I Certification Guide

PREPARE FOR THE 1Z0-803 EXAM

MALA GUPTA

MANNING
SHELTER ISLAND

For online information and ordering of this and other Manning books, please visit www.manning.com. The publisher offers discounts on this book when ordered in quantity. For more information, please contact

Special Sales Department
Manning Publications Co.
20 Baldwin Road
PO Box 261
Shelter Island, NY 11964
Email: orders@manning.com

Manning Publications Co.
20 Baldwin Road
PO Box 261
Shelter Island, NY 11964

Development editor: Cynthia Kane
Technical editor: Brent Watson
Technical proofreader: Jeanne Boyarsky
Copyeditors: Tara Walsh, Bob Herbstman, Nancy Wolfe Kotary
Proofreader: Andy Carroll
Typesetter: Dennis Dalinnik
Illustrator: Martin Murtonen
Cover designer: Marija Tudor

ISBN: 9781617291043
Printed in the United States of America
1 2 3 4 5 6 7 8 9 10 – MAL – 19 18 17 16 15 14 13

To my pillar of strength, my best friend,
and my husband, Dheeraj Prakash

brief contents

contents

foreword

Taking the OCA Java Programmer I exam is a bit like taking a driving test. First you learn the basics, like where the brakes are. Then you start driving, and then you get ready to take the driving test to get your license. The written test includes things everyone should know, things that you'll never use after the road test, and some things that are tricky edge cases. While the programmer exam cares about breaks more than brakes, it certainly likes edge cases!

Consider Mala Gupta your driving instructor to get you ready for the programmer exam. Mala points out what you'll need to know when programming in the real world—on your first job.

And consider this book your driver's manual. It gives you the rules of the road of Java, plus the gotchas that show up on that pesky written test. But don't worry, it is much more fun to read this book than the driver's manual. Just like the driver's manual won't teach you everything about driving, this book won't teach you everything there is to know about Java. If you haven't yet, read an intro to a Java book first. Start with a book like *Head First Java* or *Thinking in Java* and then come back to this book to study for the exam.

As the technical proofreader of this book, I got to see it evolve and get better as Mala worked on it. Through the conversations we had on little things, I learned that Mala knows her stuff and is a great teacher of Java. While I've only technical proofread a handful of books, I've posted reviews of over 150 technical books on Amazon, which makes it easy to spot a book that isn't clear or helpful. I'm happy to say that Mala's explanations are all very clear, and the pointers are great.

I also got to read Mala's posts in the certification forums at coderanch.com. She's been sharing updates about the exam as it comes out and posting fairly regularly for over a year. As a senior moderator at coderanch.com, it is great to see an author sharing her wisdom. It's also nice to see the similarity in writing style between the forum posts and the book. This shows the book is readable and written in an easy-to-understand, casual style.

I particularly liked the diagrams, flow charts, and cartoons in this book. And, of course, the annotated code examples I've come to expect from any Manning book. Each chapter ends with sample mock exam questions and there is a full mock exam at the end. This gives you good practice in getting ready for the exam. Wrong answers are well explained so you don't make the same mistakes over and over.

My favorite part of the book is the "Twist in the Tale" exercises. Mala gives a number of examples of how making a seemingly minor change to the code can have major consequences. These exercises develop your attention to detail so you are more observant for the mock exam questions and the exam itself.

I had already passed the OCA Java Programmer exam with a score of 98% before reading this book. If I hadn't, the questions would have prepared me for the exam. Studying from this book will give you the skills and confidence you need to become an Oracle Certified Associate Java Programmer. Happy coding and good luck on the exam!

JEANNE BOYARSKY
SENIOR DEVELOPER & MODERATOR
CODERANCH

preface

Java programmer certifications are designed to tell would-be employers whether you really know your stuff, and cracking the OCA Java SE 7 Programmer Certification is not an easy task. Thorough preparation is crucial if you want to pass the exam the first time with a score that you can be proud of. You need to know Java inside and out, and you need to understand the certification process so that you're ready for the challenging questions you'll face in the exam.

This book is a comprehensive guide to the 1Z0-803 exam. You'll explore a wide range of important Java topics as you systematically learn how to pass the certification exam. Each chapter starts with a list of the exam objectives covered in that chapter. Throughout the book you'll find sample questions and exercises designed to reinforce key concepts and prepare you for what you'll see in the real exam, along with numerous tips, notes, and visual aids.

Unlike many other exam guides, this book provides multiple ways to digest important techniques and concepts, including comic conversations, analogies, pictorial representations, flowcharts, UML diagrams, and, naturally, lots of well-commented code. The book also gives insight into typical exam question mistakes and guides you in avoiding traps and pitfalls. It provides

- 100% coverage of exam topics, all mapped to chapter and section numbers
- Hands-on coding exercises, including particularly challenging ones that throw in a twist

- Instruction on what's happening behind the scenes using the actual code from the Java API source
- Mastery of both the concepts and the exam

This book is written for developers with a working knowledge of Java. My hope is that the book will deepen your knowledge, prepare you well for the exam, and that you will pass it with flying colors!

acknowledgments

First and foremost, I thank Dheeraj Prakash—my pillar of strength, my best friend and my husband. This book wouldn't exist without his efforts. His constant guidance, encouragement, and love kept me going. He helped me to get started with this book and got me over the goal line.

My sincere gratitude to Marjan Bace, publisher at Manning, for giving me the opportunity to author this book. The Manning team has been wonderful—Scott Meyers ensured that it was worth it for Manning to have a book on this subject. Cynthia Kane, my development editor, played a major role in shaping the organization of individual chapters and the overall book. It has been a real pleasure to work with her. Copyeditors Tara Walsh, Bob Herbstman, and Nancy Wolfe Kotary not only applied their magic to sentence and language constructions but also supplemented their editing with valuable suggestions on technical content.

Technical Editor Brent Watson did a brilliant job of reviewing the complete book contents in a limited time, catching even the smallest errors in the book. Technical Proofreader Jeanne Boyarsky was outstanding and an amazing person to work with. She was very quick at reviewing the book, with an eye for detail. Proofreader Andy Carroll was extremely capable and talented. He reviewed the final manuscript with great precision.

The technical reviewers on this book did an awesome job of reviewing the contents and sharing their valuable feedback and comments: Roel De Nijs, Ivan Todorovic, Michael Piscatello, Javier Valverde, Anayonkar Shivalkar, Kyle Smith, Niklas Rosencrantz, Ashwin Mhatre, Janki Shah, Dmitriy Andrushko, Nitesh Nandwana, and Priyanka Manchanda. I would also like to thank Ozren Harlovic, Review Editor, for managing

the whole review process and meticulously funneling the feedback to make this book better.

Martin Murtonen did an outstanding job of converting the black and white hand-drawn illustrations into glorious images. It was amazing to scrutinize the page proofs. I thank Dennis Dalinnik for adjusting the images in the final page proofs, which was a lot of work. Janet Vail and Mary Piergies were awesome in their expertise at turning all text, code, and images into publishable form. I am also grateful to Candace Gillhoolley and Nermina Miller for their efforts in promoting the book.

I thank the MEAP readers for buying the book while it was being developed and for their suggestions, corrections, and encouragement: Samuel Prette, David C., Diego Poggioli, Baptize, Jayagopi Jagadeesan, David Vonka, Joel Rainey, Steve Breese, and Jörgen Persson.

I would also like to thank my former colleagues Harry Mantheakis, Paul Rosenthal, and Selvan Rajan, whose names I use in coding examples throughout the book. I have always looked up to them.

I thank my nine-year-old daughter, Shreya, an artist, who often advised me on the images that I created for the book. I'm also grateful to my younger daughter, Pavni, who patiently waited for my attention all these months when my focus was on the book. I thank my family for their unconditional support. The book would have been not been possible without their love and encouragement.

about this book

This book is written for developers with a working knowledge of Java who want to earn the OCA Java SE 7 Programmer certification. It uses powerful tools and features to make reaching your goal of certification a quick, smooth, and enjoyable experience. This section will explain the features used in the book and tell you how to use the book to get the most out of it as you prepare for the certification exam. More information on the exam and on how the book is organized is available in the Introduction.

Start your preparation with the chapter-based exam objective map

I strongly recommend a structured approach to preparing for this exam. To help you with this task, I've developed a chapter-based exam objective map, as shown in figure 1. The full version is in the Introduction (table I.3).

	Exam objectives	Covered in chapter/section	Your readiness score
1	**Java basics**	**Chapters 1 and 3**	
1.1	Define the scope of variables	Section 3.1	☆☆☆
1.2	Define the structure of a Java class	Section 1.1	☆☆☆
1.3	Create executable Java applications with a main method	Section 1.2	☆☆☆

Figure 1 The Introduction to this book provides a list of all exam objectives and the corresponding chapter and section numbers where they are covered. See the full table in the Introduction (table I.3).

As you go through your preparation, mark your readiness score for each section. Self-assessment is an important tool that will help you determine when you are ready to take the exam.

The map in the Introduction shows the complete exam objective list mapped to the relevant chapter and section numbers. You can jump to the relevant section number to work on a particular exam topic.

Chapter-based objectives

Each chapter starts with a list of the exam objectives covered in that chapter, as shown in figure 2. This list is followed by a quick comparison of the major concepts and topics covered in the chapter with real-world objects and scenarios.

Exam objectives covered in this chapter	What you need to know
[3.4] Create `if` and `if-else` constructs.	How to use `if`, `if-else`, `if-else-if-else`, and nested `if` constructs. The differences between using these `if` constructs with and without curly braces `{}`.

Figure 2 An example of the list of exam objectives and brief explanations at the beginning of each chapter

Section-based objectives

Each main section in a chapter starts by identifying the exam objective(s) that it covers. Each listed exam topic starts with the exam objective and its subobjective number.

In figure 3, the number 4.4 refers to section 4.4 in chapter 4 (the complete list of chapters and sections can be found in the table of contents). The 4.3 preceding the exam objective refers to the objective's numbering in the list of exam objectives on Oracle's website (the complete numbered list of exam objectives is given in table I.3 in the Introduction).

4.4 ArrayList

[4.3] Declare and use an ArrayList

In this section, I'll cover how to use `ArrayList`, its commonly used methods, and the advantages it offers over an array.

Figure 3 An example of the beginning of a section, identifying the exam objective that it covers

Exam tips

Each chapter provides multiple *exam tips* to re-emphasize the points that are the most confusing, overlooked, or frequently answered incorrectly by candidates and that therefore require special attention for the exam. Figure 4 shows an example.

EXAM TIP An `ArrayList` preserves the order of insertion of its elements. `Iterator`, `ListIterator`, and the enhanced `for` loop will return the elements in the order in which they were added to the `ArrayList`.

Figure 4 Example of an exam tip; they occur multiple times in a chapter

Notes

All chapters also include multiple notes, which draw your attention to points that should be noted while you're preparing for the exam. Figure 5 shows an example.

NOTE Though the terms *method parameters* and *method arguments* are not the same, you may have noticed that they are used interchangeably by many programmers. *Method parameters* are the variables that appear in the definition of a method. *Method arguments* are the actual values that are passed to a method while executing it. In figure 3.13, variables `phNum` and `msg` are method parameters. If you execute this method as `sendMsg("123456", "Hello")`, then the `String` values `"123456"` and `"Hello"` are method arguments.

Figure 5 Example note

Sidebars

Sidebars contain information that may not be directly relevant to the exam but that is related to it. Figure 6 shows an example.

Static classes and interfaces

Certification aspirants frequently ask questions about `static` classes and interfaces, so I'll quickly cover these in this section to ward off any confusion related to them. But note that `static` classes and interfaces are types of nested classes and interfaces that aren't covered by the OCA Java 7 Programmer I exam.

You can't prefix the definition of a top-level class or an interface with the keyword `static`. A top-level class or interface is one that isn't defined within another class or interface. The following code will fail to compile:

```
static class Person {}
```

Figure 6 Example sidebar

Images

I've used a lot of images in the chapters for an immersive learning experience. I believe that a simple image can help you understand a concept quickly, and a little humor can help you to retain information longer.

Simple images are used to draw your attention to a particular line of code (as shown in figure 7).

```
public String replace(char oldChar, char newChar) {
    if (oldChar != newChar) {
        // code to create a new char array and
        // replace the desired char with the new char

        return new String(0, len, buf);
    }
    return this;
}
```

`replace` creates and returns a new `String` object. It doesn't modify the existing array value.

Figure 7 An example image that draws your attention to a particular line of code

I've used pictorial representation of data in arrays (figure 8) and other data types to aid visualization and understanding.

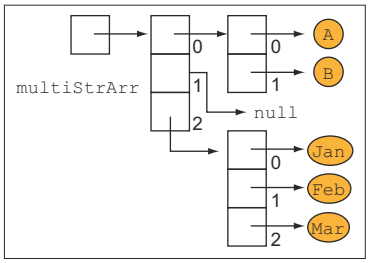

Figure 8 An example pictorial representation of data in an array

To reinforce important points and help you retain them longer, a little humor has been added using comic strips (as in figure 9).

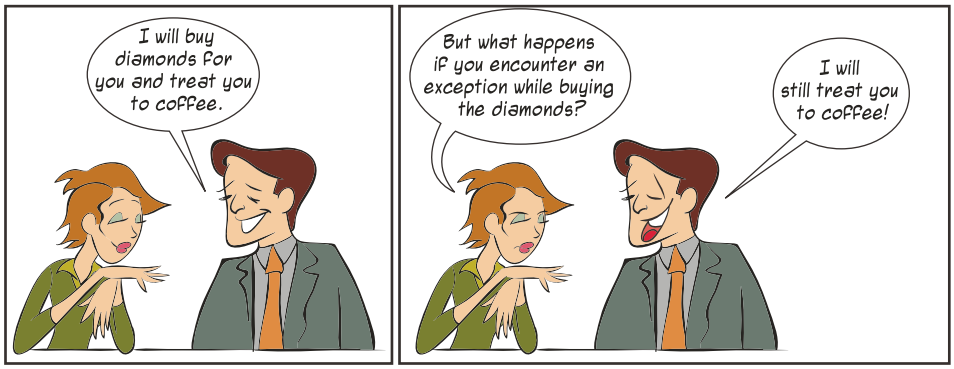

Figure 9 An example of a little humor to help you remember that the `finally` block always executes

I've also used images to group and represent information for quick reference. Figure 10 shows an example of the protected members that can be accessed by derived

or unrelated classes in the same or separate packages. I strongly recommend that you try to create a few of your own figures like these.

Figure 10 An example of grouping and representing information for quick reference

An image can also add more meaning to a sequence of steps also explained in the text. For example, figure 11 seems to bring the Java compiler to life by allowing it to talk with you and convey what it does when it gets to compile a class that doesn't define a constructor. Again, try a few of your own! It'll be fun!

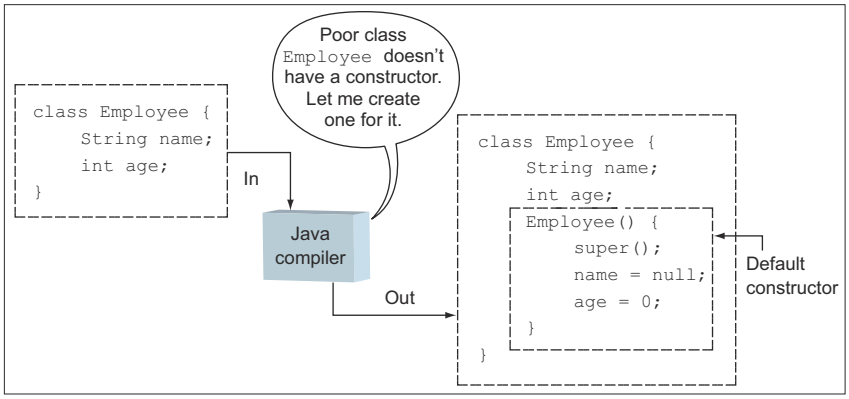

Figure 11 An example pictorial representation of steps executed by the Java compiler when it compiles a class without a constructor

The exam requires that you know multiple methods from classes such as `String`, `StringBuilder`, `ArrayList`, and others. The number of these methods can be overwhelming, but grouping these methods according to their functionality can make this task a lot more manageable. Figure 12 shows an example of an image that groups methods of the `String` class according to their functionality.

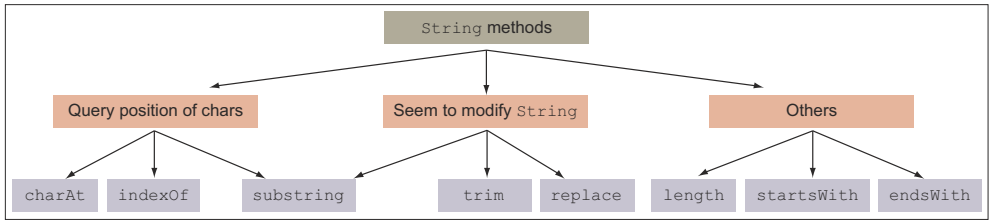

Figure 12 An example image used to group methods of the `String` class according to their functionality.

Expressions that involve multiple operands can be hard to comprehend. Figure 13 is an example of an image that can save you from the mayhem of unary increment and decrement operators used in prefix and postfix notation.

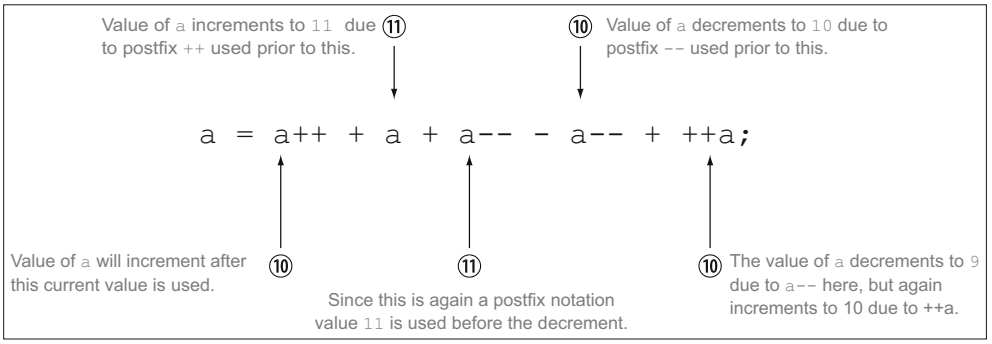

Figure 13 Example of values taken by the operands during execution of an expression

Code snippets that define multiple points and that may result in the nonlinear execution of code can be very difficult to comprehend. These may include selection statements, loops, or exception-handling code. Figure 14 is an example of an image that clearly outlines the lines of code that will execute.

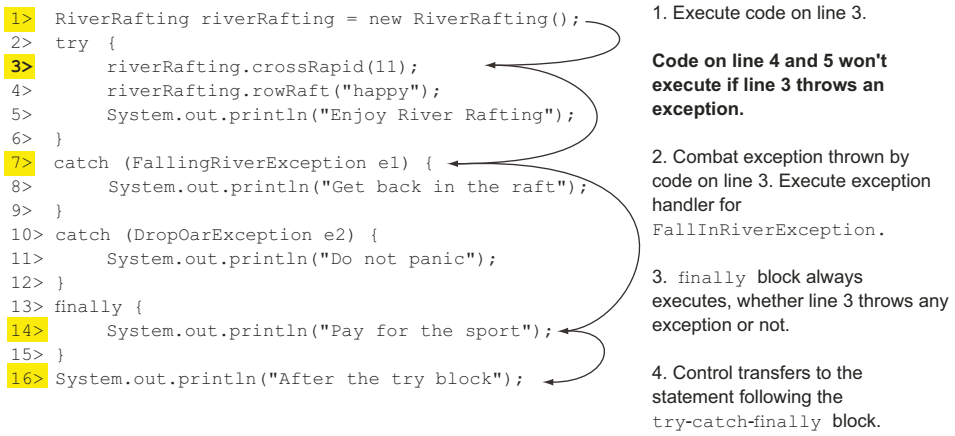

Figure 14 An example of flow of control in a code snippet that may define multiple points of nonlinear execution of code

Twist in the Tale exercises

Each chapter includes a few Twist in the Tale exercises. For these exercises, I've tried to use modified code from the examples already covered in a chapter, and the "Twist in the Tale" title refers to modified or tweaked code. These exercises highlight how

even small code modifications can change the behavior of your code. They should encourage you to carefully examine all of the code in the exam.

My main reason for including these exercises is that on the real exam, you may get to answer more than one question that seems to define exactly the same question and answer options. But upon closer inspection, you'll realize that these questions differ slightly, and that these differences change the behavior of the code and the correct answer option.

The answers to all of the Twist in the Tale exercises are given in the appendix.

Code Indentation

Some of the examples in this book show incorrect indentation of code. This has been done on purpose because on the real exam you can't expect to see perfectly indented code. You should be able to comprehend incorrectly indented code to answer an exam question correctly.

Review notes

When you're ready to take your exam, don't forget to reread the review notes a day before or on the morning of the exam. These notes contain important points from each chapter as a quick refresher.

Exam questions

Each chapter concludes with a set of 10 to 11 exam questions. These follow the same pattern as the real exam questions. Attempt these exam questions after completing a chapter.

Answers to exam questions

The answers to all exam questions provide detailed explanations, including why options are correct or incorrect. Mark your incorrect answers and identify the sections that you need to reread. If possible, draw a few diagrams—you'll be amazed at how much they can help you retain the concepts. Give it a try—it'll be fun!

Author Online

The purchase of *OCA Java SE 7 Programmer I Certification Guide* includes free access to a private forum run by Manning Publications where you can make comments about the book, ask technical questions, and receive help from the author and other users. You can access and subscribe to the forum at www.manning.com/OCAJavaSE7ProgrammerICertification-Guide. This page provides information on how to get on the forum once you're registered, what kind of help is available, and the rules of conduct in the forum.

Manning's commitment to our readers is to provide a venue where a meaningful dialogue among individual readers and between readers and the author can take place. It's not a commitment to any specific amount of participation on the part of the authors, whose contribution to the book's forum remains voluntary (and unpaid). We suggest you try asking the author some challenging questions, lest her interest stray!

The Author Online forum and the archives of previous discussions will be accessible from the publisher's website as long as the book is in print.

about the author

Mala Gupta has a Master's degree in Computer Applications (MCA). She is an Oracle Certified Associate-Java SE 7 Programmer, Java Sun Certified Web Component Developer (SCWCD), and Sun Certified Java 2 Programmer (SCJP).

She has more than 12 years of experience in software design and development and training. Her work experience is in Java technologies, primarily as an analyst, programmer, and mentor.

Mala has worked with international training and software services organizations in Europe and development centers in India on various Java-based portals and web applications. She has experience in mentoring and ramping up teams' technical and process skills.

She is the founder and lead mentor of a portal (http://ejavaguru.com) that has offered an online Java course in Java Programmer certification since 2006.

about the cover illustration

The figure on the cover of the *OCA Java SE 7 Programmer I Certification Guide* is captioned a "Morlach." This illustration is taken from a recent reprint of Balthasar Hacquet's *Images and Descriptions of Southwestern and Eastern Wenda, Illyrians, and Slavs* published by the Ethnographic Museum in Split, Croatia, in 2008. Hacquet (1739–1815) was an Austrian physician and scientist who spent many years studying the botany, geology, and ethnography of many parts of the Austrian Empire, as well as the Veneto, the Julian Alps, and the western Balkans, inhabited in the past by peoples of many different tribes and nationalities. Hand-drawn illustrations accompany the many scientific papers and books that Hacquet published.

Morlachs were a rural population that lived in the Dinaric Alps in the western Balkans hundreds of years ago. Many of them were shepherds who migrated in search of better pastures for their flocks, alternating between the mountains in the summer and the seashore in the winter. They were also called "Vlachs" in Serbian and Croatian. The rich diversity of the drawings in Hacquet's publications speaks vividly of the uniqueness and individuality of Alpine and Balkan regions just 200 years ago. This was a time when the dress codes of two villages separated by a few miles identified people uniquely as belonging to one or the other, and when members of an ethnic tribe, social class, or trade could be easily distinguished by what they were wearing.

Dress codes have changed since then and the diversity by region, so rich at the time, has faded away. It is now often hard to tell the inhabitant of one continent from another and the residents of the picturesque towns and villages in the Balkans are not readily distinguishable from people who live in other parts of the world.

We at Manning celebrate the inventiveness, the initiative, and the fun of the computer business with book covers based on costumes from two centuries ago brought back to life by illustrations such as this one.

Introduction

This introduction covers

- Introduction to the Oracle Certified Associate (OCA) Java SE 7 Programmer certification (exam number 1Z0-803)
- Importance of OCA Java SE 7 Programmer certification
- Comparison of the OCA Java SE 7 Programmer I exam with OCA Java SE 5/6 exam
- Comparison of the OCA Java SE 7 Programmer I exam (1Z0-803) with OCP Java SE 7 Programmer II exam (1Z0-804)
- Detailed exam objectives, mapped to book chapters
- Readiness checklist to determine your readiness level for writing the exam
- FAQ on exam preparation and on taking the exam
- Introduction to the testing engine used for the exam

This book is intended specifically for individuals who wish to earn the Oracle Certified Associate (OCA) Java SE 7 Programmer certification (exam number 1Z0-803). It assumes that you are familiar with Java and have some experience working with it.

If you are completely new to Java or to object-oriented languages, I suggest that you start your journey with an entry-level book and then come back to this one.

1 Disclaimer

The information in this chapter is sourced from Oracle.com, public websites, and user forums. Input has been taken from real people who have earned Java certification, including the author. All efforts have been made to maintain the accuracy of the content, but the details of the exam—including the exam objectives, pricing, exam pass score, total number of questions, maximum exam duration, and others—are subject to change per Oracle's policies. The author and publisher of the book shall not be held responsible for any loss or damage accrued due to any information contained in this book or due to any direct or indirect use of this information.

2 Introduction to OCA Java SE 7 Programmer certification

The Oracle Certified Associate (OCA) Java SE 7 Programmer I exam (1Z0-803) covers the fundamentals of Java SE 7 programming, such as the importance of object-oriented programming, its implementation in code, and using flow control, arrays, and other constructs.

This exam is the first of the two steps in earning the title of Oracle Certified Professional (OCP) Java SE 7 Programmer. It certifies that an individual possesses a strong foundation in the Java programming language. Table 1 lists the details of this exam.

Table 1 Details for OCA Java SE 7 Programmer I exam (1Z0-803)

Exam number	1Z0-803
Java version	Based on Java version 7
Number of questions	90
Passing score	77%
Time duration	140 minutes
Pricing	US$300
Type of questions	Multiple-choice questions

2.1 The importance of OCA Java SE 7 Programmer certification

The OCA Java SE 7 Programmer I exam (1Z0-803) is an entry-level exam in your Java certification roadmap, as shown in figure 1. This exam is a prerequisite for the OCP Java SE 7 Programmer II exam (1Z0-804), which is itself a prerequisite for most of the other Oracle certifications in Java. The dashed lines and arrows in figure 1 depict the prerequisites for a certification.

As shown in figure 1, the Java certification tracks are offered under the categories Associate, Professional, Expert, and Master.

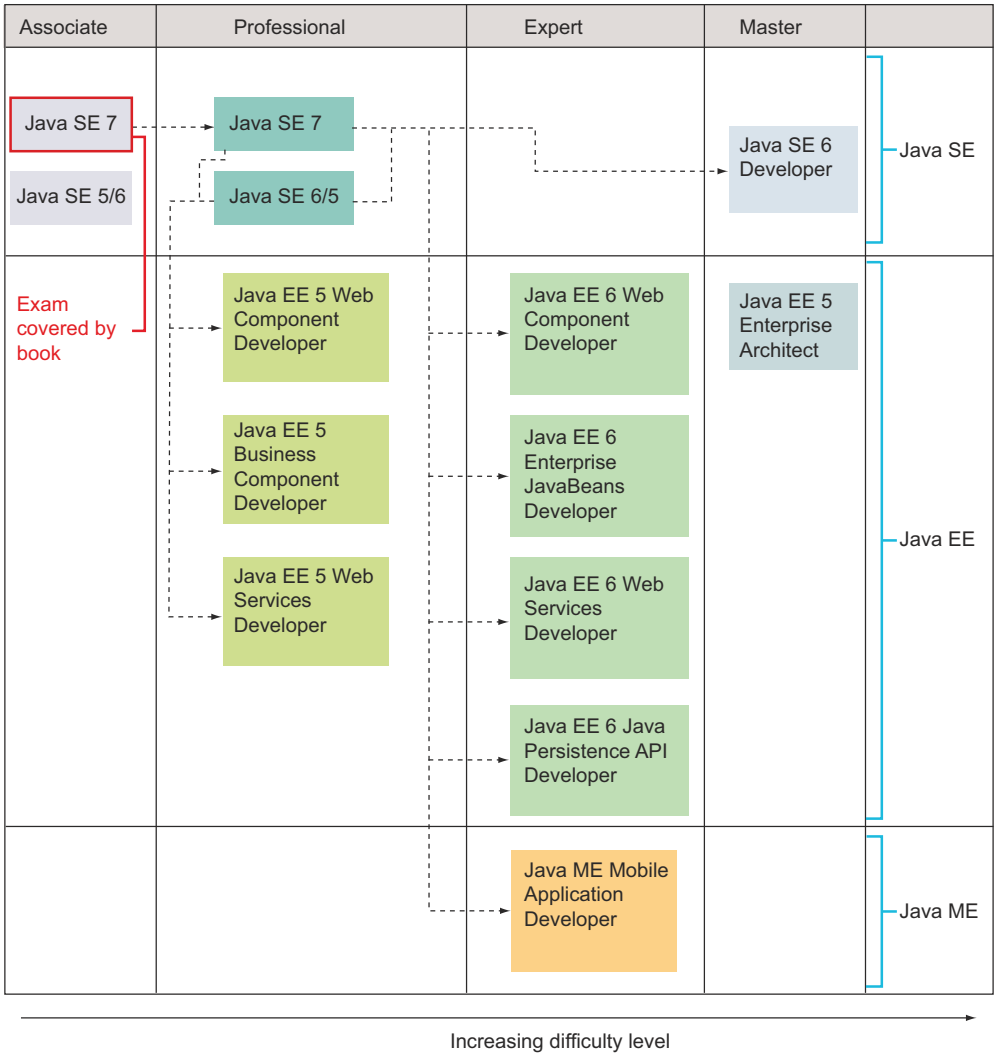

Figure 1 OCA Java SE 7 Programmer certification is the entry-level certification in the Java certification roadmap. It's a prerequisite for the OCP Java SE 7 Programmer II exam (1Z0-804), which is a prerequisite for most of the other certifications in Java.

2.2 Comparing OCA Java exam versions

This section will clear up any confusion surrounding the different versions of the OCA Java exam. As of now, Oracle offers two versions of the OCA certification in Java:

- OCA Java SE 7 Programmer I (exam number: 1Z0-803)
- OCA Java SE 5/SE 6 (exam number: 1Z0-850)

These two exam versions are quite different as far target audience, total number of questions, passing score, and exam duration are concerned, as listed in table 2.

Table 2 Comparing exams: OCA Java SE 7 Programmer I and OCA Java SE 5/6

	OCA Java SE 7 Programmer I (1Z0-803)	OCA Java SE 5/SE 6 (1Z0-850)
Target audience	Java programmers	Java programmers and IT managers
Java version	Based on Java version 7	Based on Java version 5/6
Total number of questions	90	51
Exam duration	140 minutes	115 minutes
Passing score	77%	68%
Pricing	US$300	US$300

Figure 2 shows a detailed comparison of the exam objectives of OCA Java SE 5/6 (1Z0-850) and OCA Java SE 7 Programmer I (1Z0-803). It shows objectives that are exclusive to each of these exam versions and those that are common to both. The first column shows the objectives that are included only in OCA Java SE 5/6 (1Z0-850), the middle column shows common exam objectives, and the right column shows exam objectives covered only in OCA Java SE 7 Programmer I (1Z0-803).

2.3 Comparing the OCA Java SE 7 Programmer I (1Z0-803) and OCP Java SE 7 Programmer II (1Z0-804) exams

The confusion between these two exams is due to the similarity in their names, but these are two separate exams. Starting with Java 7, Oracle has raised the bar to earn the title of Oracle Certified Professional Java SE 7 Programmer, which now requires successfully completing the following two exams:

- OCA Java SE 7 Programmer I (exam number: 1Z0-803)
- OCP Java SE 7 Programmer II (exam number: 1Z0-804)

The OCP Java SE 7 Programmer certification is designed for individuals who possess advanced skills in the Java programming language. This certification covers comparatively advanced Java features, such as threads, concurrency, Java file I/O, inner classes, localization, and others.

2.4 Complete exam objectives, mapped to book chapters, and readiness checklist

Table 3 includes a complete list of exam objectives for the OCA Java SE 7 Programmer I exam, which was taken from Oracle's website. All the objectives are mapped to the book's chapters and the section numbers that cover them. You can also check your readiness to take the exam by selecting the appropriate stars. Here's the legend:

☆ Basic knowledge
☆☆ Intermediate (you can use it in code)
☆☆☆ Advanced (you can answer all questions about it)

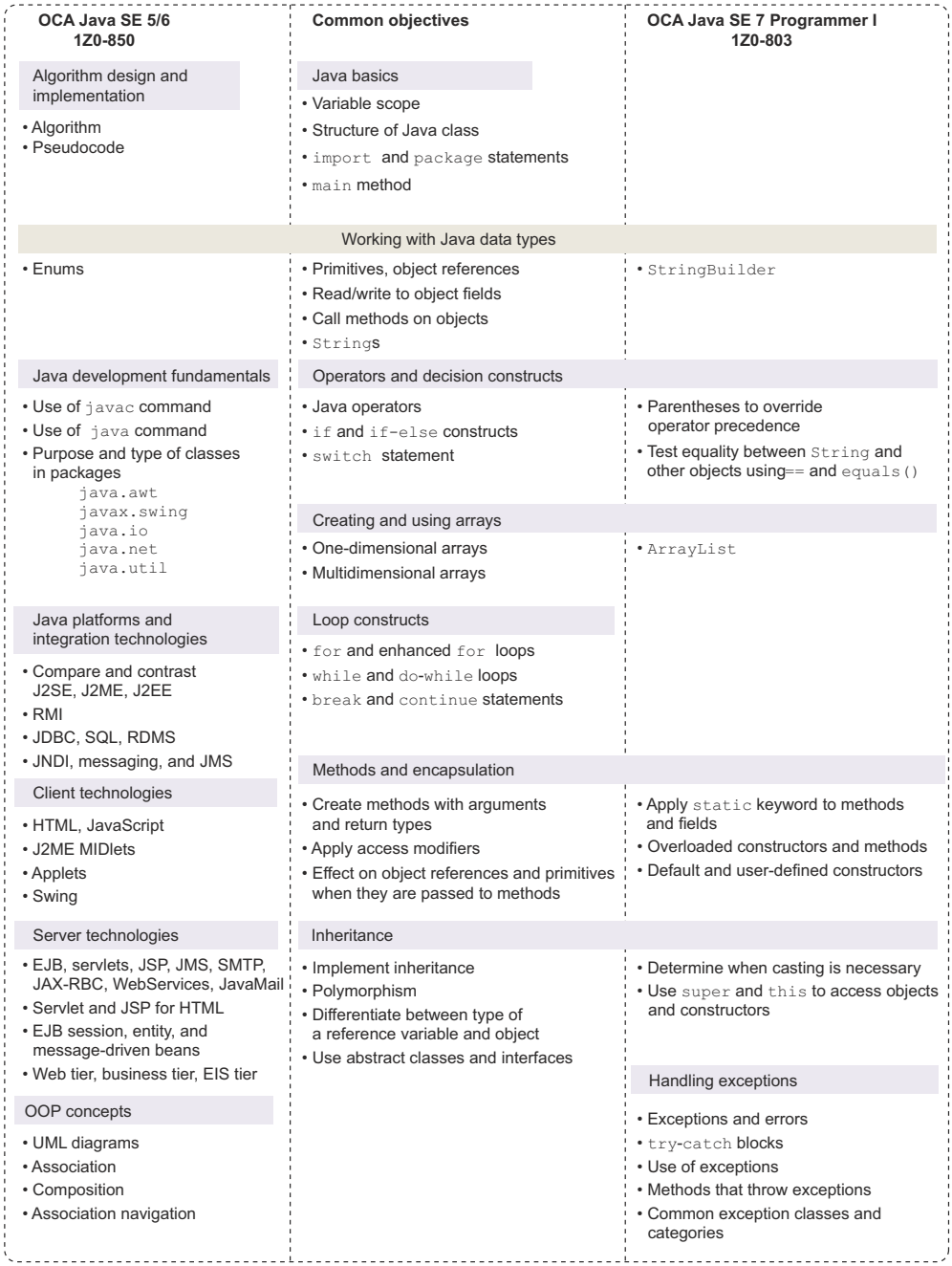

Figure 2 Comparing objectives of exams OCA Java SE 5/6 and OCA Java SE 7 Programmer I

Table 3 Exam objectives and subobjectives mapped to chapter and section numbers, with readiness score

	Exam objectives	Covered in chapter/section	Your readiness score
1	**Java basics**	**Chapters 1 and 3**	
1.1	Define the scope of variables	Section 3.1	☆☆☆
1.2	Define the structure of a Java class	Section 1.1	☆☆☆
1.3	Create executable Java applications with a main method	Section 1.2	☆☆☆
1.4	Import other Java packages to make them accessible in your code	Section 1.3	☆☆☆
2	**Working with Java data types**	**Chapters 2, 3, and 4**	
2.1	Declare and initialize variables	Sections 2.1 and 2.3	☆☆☆
2.2	Differentiate between object reference variables and primitive variables	Sections 2.1 and 2.3	☆☆☆
2.3	Read or write to object fields	Section 3.6	☆☆☆
2.4	Explain an object's life cycle	Section 3.2	☆☆☆
2.5	Call methods on objects	Section 3.6	☆☆☆
2.6	Manipulate data using the `StringBuilder` class and its methods	Section 4.2	☆☆☆
2.7	Create and manipulate strings	Section 4.1	☆☆☆
3	**Using operators and decision constructs**	**Chapters 2, 4, and 5**	
3.1	Use Java operators	Section 2.4	☆☆☆
3.2	Use parentheses to override operator precedence	Section 2.4	☆☆☆
3.3	Test equality between strings and other objects using `==` and `equals()`	Section 4.1	☆☆☆
3.4	Create `if` and `if-else` constructs	Section 5.1	☆☆☆
3.5	Use a `switch` statement	Section 5.2	☆☆☆
4	**Creating and using arrays**	**Chapter 4**	
4.1	Declare, instantiate, initialize, and use a one-dimensional array	Section 4.3	☆☆☆
4.2	Declare, instantiate, initialize, and use a multidimensional array	Section 4.3	☆☆☆
4.3	Declare and use an `ArrayList`	Section 4.4	☆☆☆

Table 3 Exam objectives and subobjectives mapped to chapter and section numbers, with readiness score *(continued)*

	Exam objectives	Covered in chapter/section	Your readiness score
5	**Using loop constructs**	**Chapter 5**	
5.1	Create and use `while` loops	Section 5.5	☆☆☆
5.2	Create and use `for` loops, including the enhanced `for` loop	Sections 5.3 and 5.4	☆☆☆
5.3	Create and use `do-while` loops	Section 5.5	☆☆☆
5.4	Compare loop constructs	Section 5.6	☆☆☆
5.5	Use `break` and `continue`	Section 5.7	☆☆☆
6	**Working with methods and encapsulation**	**Chapters 1 and 3**	
6.1	Create methods with arguments and return values	Section 3.3	☆☆☆
6.2	Apply the `static` keyword to methods and fields	Section 1.5	☆☆☆
6.3	Create an overloaded method	Section 3.4	☆☆☆
6.4	Differentiate between default and user-defined constructors	Section 3.5	☆☆☆
6.5	Create and overload constructors	Section 3.5	☆☆☆
6.6	Apply access modifiers	Section 1.4	☆☆☆
6.7	Apply encapsulation principles to a class	Section 3.7	☆☆☆
6.8	Determine the effect upon object references and primitive values when they are passed into methods that change the values	Section 3.8	☆☆☆
7	**Working with inheritance**	**Chapters 1 and 6**	
7.1	Implement inheritance	Section 6.1	☆☆☆
7.2	Develop code that demonstrates the use of polymorphism	Section 6.6	☆☆☆
7.3	Differentiate between the type of a reference and the type of an object	Section 6.3	☆☆☆
7.4	Determine when casting is necessary	Section 6.4	☆☆☆
7.5	Use `super` and `this` to access objects and constructors	Section 6.5	☆☆☆
7.6	Use abstract classes and interfaces	Sections 1.5, 6.2, and 6.6	☆☆☆

Table 3 Exam objectives and subobjectives mapped to chapter and section numbers, with readiness score *(continued)*

	Exam objectives	Covered in chapter/section	Your readiness score
8	**Handling exceptions**	**Chapter 7**	
8.1	Differentiate among checked exceptions, `RuntimeExceptions`, and `Errors`	Section 7.3	☆☆☆
8.2	Create a `try-catch` block and determine how exceptions alter normal program flow	Section 7.2	☆☆☆
8.3	Describe what exceptions are used for in Java	Section 7.1	☆☆☆
8.4	Invoke a method that throws an exception	Section 7.2	☆☆☆
8.5	Recognize common exception classes and categories	Section 7.4	☆☆☆

When you are ready to take the exam, you should ideally be able to select three stars for each item in the table. But let's define a better way to evaluate your exam readiness. Once you have marked all the stars in the previous chart, calculate your total points using the following values:

☆ 1 point
☆☆ 2 points
☆☆☆ 4 points

As the maximum number of points is 172 (43 objectives × 4), a score in the range of 150–172 is considered a good score.

You can download a PDF version of the form from the book's web page at http://manning.com/gupta/ if you wish to mark yourself more than once.

3 FAQs

You might be anxious when you start your exam preparation or even think about getting certified. This section can help calm your nerves by answering frequently asked questions on exam preparation and on writing the exam.

3.1 FAQs on exam preparation

This sections answers frequently asked questions on how to prepare for the exam, including the best approach, study material, preparation duration, how to test self-readiness, and more.

WILL THE EXAM DETAILS EVER CHANGE FOR THE OCA JAVA SE 7 PROGRAMMER I EXAM?

Oracle can change the exam details for a certification even after the certification is made live. The changes can be to the exam objectives, pricing, exam duration, exam questions, and other parts. In the past, Oracle has made similar changes to certification

exams. Such changes may not be major, but it is always advisable to check Oracle's website for the latest exam information when you start your exam preparation.

WHAT IS THE BEST WAY TO PREPARE FOR THIS EXAM?

At the time of writing this book, there weren't many resources available to prepare for this exam. Apart from this book, Oracle offers an online course on this exam.

Generally, candidates use a combination of resources, such as books, online study materials, articles on the exam, free and paid mock exams, and training to prepare for the exam. Different combinations work best for different people, and there is no one perfect formula to prepare. Depending on whether training or self-study works best for you, you can select the method that is most appropriate for you. Combine it with a lot of code practice and mock exams.

HOW DO I KNOW WHEN I AM READY FOR THE EXAM?

You can be sure about your exam readiness by *consistently* getting a good score in the mock exams. Generally, a score of 80% and above in approximately seven mock exams (the more the better) attempted consecutively will assure you of a similar score in the real exam. You can also test your exam readiness using table 3. This table contains exam objectives and subobjectives with multiple stars representing different levels of expertise.

HOW MANY MOCK TESTS SHOULD I ATTEMPT BEFORE THE REAL EXAM?

Ideally, you should attempt at least 10 mock exams before you attempt the real exam. The more the better!

I HAVE TWO YEARS' EXPERIENCE WORKING WITH JAVA. DO I STILL NEED TO PREPARE FOR THIS CERTIFICATION?

It is important to understand that there is a difference between the practical knowledge of having worked with Java and the knowledge required to pass this certification exam. The authors of the Java certification exams employ multiple tricks to test your knowledge. Hence, you need a structured preparation and approach to succeed in the certification exam.

WHAT IS THE IDEAL TIME REQUIRED TO PREPARE FOR THE EXAM?

The preparation time frame mainly depends on your experience with Java and the amount of time that you can spend to prepare yourself. On average, you will require approximately 150 hours of study over two or three months to prepare for this exam. Again, the number of study hours required depends on individual learning curves and backgrounds.

It's important to be consistent with your exam preparation. You cannot study for a month and then restart after, say, a gap of a month or more.

DOES THIS EXAM INCLUDE ANY UNSCORED QUESTIONS?

A few of the questions that you write in any Oracle exam may be marked unscored. Oracle's policy states that while writing an exam, you won't be informed whether a question will be scored. You may be surprised to learn that as many as 10 questions out of the 90 questions in the OCA Java SE 7 Programmer I exam may be unscored. Even if you answer a few questions incorrectly, you stand a chance of scoring 100%.

Oracle regularly updates its question bank for all its certification exams. These unscored questions may be used for research and to evaluate new questions that can be added to an exam.

CAN I START MY EXAM PREPARATION WITH THE MOCK EXAMS?

If you are quite comfortable with the Java language features, then yes, you can start your exam preparation with the mock exams. This will also help you to understand the types of questions to expect in the real certification exam. But if you have little or no experience working with Java, or if you are not quite comfortable with the language features of Java, I don't advise you to start with the mock exams. The exam authors often use a lot of tricks to evaluate a candidate in the real certification exam. Starting your exam preparation with mock exams will only leave you confused about the Java concepts.

SHOULD I REALLY BOTHER GETTING CERTIFIED?

Yes, you should, for the simple reason that employers bother about the certification of employees. Organizations prefer a certified Java developer over a noncertified Java developer with similar IT skills and experience. The certification can also get you a higher paycheck than uncertified peers with comparable skills.

3.2 FAQs on taking the exam

This section contains a list of frequently asked questions related to the exam registration, exam coupon, do's and don'ts while taking the exam, and exam retakes.

WHERE AND HOW DO I WRITE THIS EXAM?

You can write this exam at an Oracle Testing Center or Pearson VUE Authorized Testing Center. To sit for the exam, you must register for the exam and purchase an exam voucher. The following options are available:

- Register for the exam and pay Pearson VUE directly.
- Purchase an exam voucher from Oracle and register at Pearson VUE to take the exam.
- Register at an Oracle Testing Center.

Look for the nearest testing centers in your area, register yourself, and schedule an exam date and time. Most of the popular computer training institutes also have a testing center on their premises. You can locate a Pearson VUE testing site at www.pearsonvue.com/oracle/, which contains detailed information on locating testing centers and scheduling or rescheduling an exam. At the time of registration, you'll need to provide the following details along with your name, address, and contact numbers:

- Exam title and number (OCA Java SE 7 Programmer I, 1Z0-803)
- Any discount code that should be applied during registration
- Oracle Testing ID/Candidate ID, if you have written any other Oracle/Sun certification exam
- Your OPN Company ID (if your employer is in the Oracle Partner Network, you can find out the company ID and use any available discounts on the exam fee)

How long is the exam coupon valid for?

Each exam coupon is printed with an expiry date. Beware of any discounted coupons that come with an assurance that they can be used past the expiration date.

Can I refer to notes or books while writing this exam?

You can't refer to any books or notes while writing this exam. You are not allowed to carry any blank paper for rough work or even your mobile phone inside the testing cubicle.

What is the purpose of marking a question while writing the exam?

By marking a question, you can manage your time efficiently. Don't spend a lot of time on a single question. You can mark a difficult question to defer answering it while writing your exam. The exam gives you an option to review answers to the marked questions at the end of the exam. Also, navigating from one question to another using the Back and Next buttons is usually time consuming. If you are unsure of an answer, mark it and review it at the end.

Can I write down the exam questions and bring them back with me?

No. The exam centers no longer provide sheets of paper for the rough work that you may need to do while taking the exam. The testing center will provide you with either erasable or nonerasable boards. If you're provided with a nonerasable board, you may request another one if you need it.

Oracle is quite particular about certification candidates distributing or circulating the memorized questions in any form. If Oracle finds out that this is happening, it may cancel a candidate's certificate, bar that candidate forever from writing any Oracle certification, inform the employer, or take legal action.

What happens if I complete the exam before or after the total time?

If you complete the exam before the total exam time has elapsed, revise your answers and click the Submit or Finish button. The screen will display your score within 10 seconds of clicking the Submit button!

If you have not clicked the Submit button and you use up all the exam time, the exam engine will no longer allow you to modify any of the exam answers and will present the screen with the Submit button.

Will I receive my score immediately after the exam?

Yes, you will. When you click the Submit button, the screen will show your total score. It will also show what you scored on each objective. The testing center will also give you hard copies of your certification score. The certificate itself will arrive via post within six to eight weeks.

What happens if I fail? Can I retake the exam?

It's not the end of the world. Don't worry if you fail. You can retake the exam after 14 days (and the world will not know it's a retake).

However, you cannot retake a passed exam to improve your score. Also, you cannot retake a beta exam.

4 The testing engine used in the exam

The user interface of the testing engine used for the certification exam is quite simple. (You could even call it primitive, compared to today's web, desktop, and smartphone applications.)

Before you can start the exam, you will be required to accept the terms and conditions of the Oracle Certification Candidate Agreement. Your computer screen will display all these conditions and give you an option to accept the conditions. You can proceed with writing the exam only if you accept these conditions.

Here are the features of the testing engine used by Oracle:

- *Engine UI is divided into three sections*—The UI of the testing engine is divided into the following three segments:
  - *Static upper section*—Displays question number, time remaining, and a checkbox to mark a question for review.
  - *Scrollable middle section*—Displays the question text and the answer options.
  - *Static bottom section*—Displays buttons to display the previous question, display the next question, end the exam, and review marked questions.
- *Each question is displayed on a separate screen*—The exam engine displays one question on the screen at a time. It does not display multiple questions on a single screen, like a scrollable web page. All effort is made to display the complete question and answer options without scrolling, or with little scrolling.
- *Code Exhibit button*—Many questions include code. Such questions, together with their answers, may require significant scrolling to be viewed. As this can be quite inconvenient, such questions include a Code Exhibit button that displays the code in a separate window.
- *Mark questions to be reviewed*—The question screen displays a checkbox with the text "Mark for review" at the top-left corner. A question can be marked using this option. The marked questions can be quickly reviewed at the end of the exam.
- *Buttons to display the previous and next questions*—The test includes buttons to display the previous and next questions within the bottom section of the testing engine.
- *Buttons to end the exam and review marked questions*—The engine displays buttons to end the exam and to review the marked questions in the bottom section of the testing engine.
- *Remaining time*—The engine displays the time remaining for the exam at the top right of the screen.
- *Question number*—Each question displays its serial number.
- *Correct number of answer options*—Each question displays the correct number of options that should be selected from multiple options.

On behalf of all at Manning Publications, I wish you good luck and hope that you score very well on your exam.