

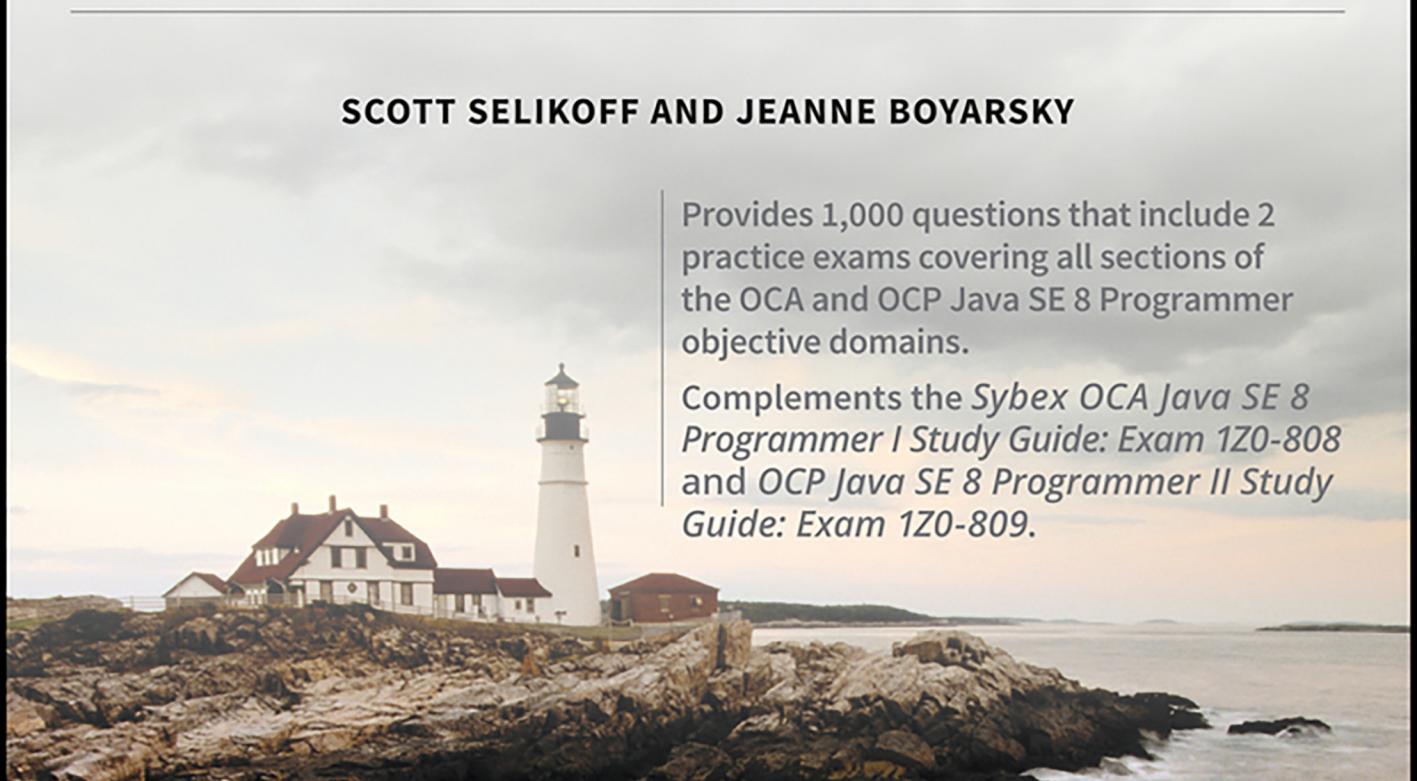
# OCA/OCP JAVA® SE 8 PROGRAMMER

---

## PRACTICE TESTS

---

SCOTT SELIKOFF AND JEANNE BOYARSKY



Provides 1,000 questions that include 2 practice exams covering all sections of the OCA and OCP Java SE 8 Programmer objective domains.

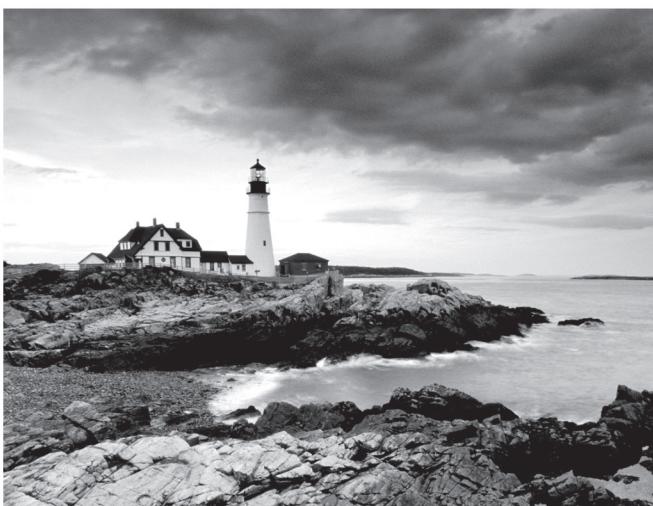
Complements the *Sybex OCA Java SE 8 Programmer I Study Guide: Exam 1Z0-808* and *OCP Java SE 8 Programmer II Study Guide: Exam 1Z0-809*.



# OCA/OCP

# Java® SE 8 Programmer

## Practice Tests



Scott Selikoff  
Jeanne Boyarsky



Senior Acquisitions Editor: Kenyon Brown  
Development Editor: David Clark  
Technical Editors: Janeice DelVecchio, Elena Felder  
Production Editor: Rebecca Anderson  
Copy Editor: Judy Flynn  
Editorial Manager: Mary Beth Wakefield  
Production Manager: Kathleen Wisor  
Executive Editor: Jim Minatel  
Book Designer: Judy Fung and Bill Gibson  
Proofreader: Nancy Carrasco  
Indexer: Ted Laux  
Project Coordinator, Cover: Brent Savage  
Cover Designer: Wiley  
Cover Image: ©Getty Images Inc./Jeremy Woodhouse  
Copyright © 2017 by John Wiley & Sons, Inc., Indianapolis, Indiana  
Published simultaneously in Canada  
ISBN: 978-1-119-36339-2  
ISBN: 978-1-119-36336-1 (ebk.)  
ISBN: 978-1-119-36340-8 (ebk.)

Manufactured in the United States of America

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

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

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

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

**Library of Congress Control Number:** 2017931099

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

10 9 8 7 6 5 4 3 2 1

*To the new little bundle of joy my wife is carrying.*

— Scott

*Remembering Einstein and CV in same year. Congrats 694!*

— Jeanne

# Acknowledgments

Scott and Jeanne would like to thank numerous individuals for their contribution to this book. Thank you to David Clark for guiding us through the process and making the book better in so many ways. Thank you to Janeice DelVecchio for being our technical editor as we wrote this book. Janeice pointed out many subtle errors in addition to the big ones. And thank you to Elena Felder for being our technical proofreader and finding the errors that we managed to sneak by Janeice. This book also wouldn't be possible without many people at Wiley, including Kenyon Brown, Rebecca Anderson, Judy Flynn, Nancy Carrasco, and so many others.

Scott could not have reached this point without the help of a small army of people, led by his perpetually understanding wife, Patti, the most wonderful mother their twin daughters, Olivia and Sophia, could ask for. Professors Johannes Gehrke and John H. Hubbard of Cornell University always believed in him and knew he would excel in his career. Scott would like to extend his gratitude to his wonderfully patient co-author Jeanne, on this their third book. "I don't know how she puts up with me, but I'm glad she does and thrilled at the quality of books we produce." A big thanks to Matt Dalen, who has been a great friend, sounding board, and caring father to the wonderfully sweet Olivia Dalen. Joel McNary introduced Scott to CodeRanch.com and encouraged him to post regularly, a step that changed his life. Finally, Scott would like to thank his mother and retired teacher, Barbara Selikoff, for teaching him the value of education and his father, Mark Selikoff, for instilling in him the benefits of working hard.

Jeanne would personally like to thank everyone at CodeRanch.com who asked and responded to OCA and OCP questions and comments about our first two books. Having dialog with the readers made this book even stronger. Jeanne would like to thank the members of FIRST robotics FRC team 694 in addition to FTC teams 310 and 479 for their support. Your questions make it so she can never forget what new learners are thinking. Go StuyPulse! Jeanne would also like to thank Onur Otlu and Thomas Campos for the feedback on some localization questions. Finally, Jeanne would like to thank Scott for being a great co-author and "not getting tired of me."

Last but not least, both Scott and Jeanne would like to give a big thank you to the readers of our OCA 8 and OCP 8 books. Hearing from all of you who enjoyed the book and passed the exam is a great feeling. We'd also like to thank those who pointed out errors and made suggestions for improvements in our OCP book. As of December 2016, the top three were Guillaume Bailly, Thalita Vergilio, and Sébastien Canonica. We also would like to thank Olivier Chalet, Mihaela Hetea, Peter Deak, Anton Shaikin, Ramya R, Tim Moroz, Aurelien Gamet, Javid Azimli, Salim Rahal, and Sander Wamelink.

# About the Authors

**Scott Selikoff** is a professional software consultant, author, and owner of Selikoff Solutions, LLC, which provides software development solutions to businesses in the tri-state New York City area. Skilled in a plethora of software languages and platforms, Scott specializes in full-stack database-driven systems, cloud-based applications, and service-oriented architectures.

A native of Toms River, New Jersey, Scott achieved his Bachelor of Arts from Cornell University in Mathematics and Computer Science in 2002, after three years of study. In 2003, he received his Masters of Engineering in Computer Science, also from Cornell University.

As someone with a deep love of education, Scott has always enjoyed teaching others new concepts. He's given lectures at Cornell University and Rutgers University, as well as conferences including the Server Side Java Symposium. Scott lives in New Jersey with his loving wife, Patti; amazing twin toddler daughters, Olivia and Sophia; and two very playful dogs, Webby and Georgette. You can find out more about Scott at <http://www.linkedin.com/in/selikoff>.

**Jeanne Boyarsky** has worked as a Java developer for over 14 years at a bank in New York City where she develops, mentors, and conducts training. Besides being a senior moderator at CodeRanch.com in her free time, she works on the forum's code base. Jeanne also mentors the programming division of a FIRST robotics team, where she works with students just getting started with Java.

Jeanne got her Bachelor of Arts degree in 2002 in Computer Science and her Masters in Computer Information Technology in 2005. She enjoyed getting her Masters degree in an online program while working full time. This was before online education was cool! Jeanne is also a Distinguished Toastmaster and a Scrum Master. You can find out more about Jeanne at <https://coderanch.com/wiki/660334>.

Scott and Jeanne are both moderators on the CodeRanch.com forums, and they can be reached there for questions and comments. They also co-author a technical blog called Down Home Country Coding at <http://www.selikoff.net>.

In addition to this book, Scott and Jeanne are also the authors of *OCA Oracle Certified Associate Java SE 8 Programmer I Study Guide* (Sybex, 2015) and *OCP Oracle Certified Professional Java SE 8 Programmer II Study Guide* (Sybex, 2016). More recently, these two books have been combined into the single release *OCA / OCP Java SE 8 Programmer Certification Kit: Exam 1Z0-808 and Exam 1Z0-809* (Sybex 2016).



# Contents

<i>Introduction</i>	<i>ix</i>
---------------------	-----------

<b>Part I</b>	<b>OCA</b>	<b>1</b>
<b>Chapter 1</b>	<b>Java Basics</b>	<b>3</b>
<b>Chapter 2</b>	<b>Working with Java Data Types</b>	<b>17</b>
<b>Chapter 3</b>	<b>Using Operators and Decision Constructs</b>	<b>33</b>
<b>Chapter 4</b>	<b>Creating and Using Arrays</b>	<b>49</b>
<b>Chapter 5</b>	<b>Using Loop Constructs</b>	<b>63</b>
<b>Chapter 6</b>	<b>Working with Methods and Encapsulation</b>	<b>79</b>
<b>Chapter 7</b>	<b>Working with Inheritance</b>	<b>97</b>
<b>Chapter 8</b>	<b>Handling Exceptions</b>	<b>115</b>
<b>Chapter 9</b>	<b>Working with Selected Classes from the Java API</b>	<b>135</b>
<b>Chapter 10</b>	<b>OCA Practice Exam</b>	<b>151</b>
<b>Part II</b>	<b>OCP</b>	<b>185</b>
<b>Chapter 11</b>	<b>Java Class Design</b>	<b>187</b>
<b>Chapter 12</b>	<b>Advanced Java Class Design</b>	<b>205</b>
<b>Chapter 13</b>	<b>Generics and Collections</b>	<b>223</b>
<b>Chapter 14</b>	<b>Lambda Built-in Functional Interfaces</b>	<b>239</b>
<b>Chapter 15</b>	<b>Java Stream API</b>	<b>255</b>
<b>Chapter 16</b>	<b>Exceptions and Assertions</b>	<b>271</b>
<b>Chapter 17</b>	<b>Use Java SE 8 Date/Time API</b>	<b>289</b>

<b>Chapter 18</b>	<b>Java I/O Fundamentals</b>	<b>301</b>
<b>Chapter 19</b>	<b>Java File I/O (NIO.2)</b>	<b>319</b>
<b>Chapter 20</b>	<b>Java Concurrency</b>	<b>337</b>
<b>Chapter 21</b>	<b>Building Database Applications with JDBC</b>	<b>357</b>
<b>Chapter 22</b>	<b>Localization</b>	<b>371</b>
<b>Chapter 23</b>	<b>OCP Practice Exam</b>	<b>385</b>
<b>Appendix</b>	<b>Answers to Review Questions</b>	<b>427</b>
<i>Index</i>		<b>565</b>

# Introduction

This book is intended for those taking either the 1Z0-808 or 1Z0-809 Oracle Java Programmer exams as well as those who want to test their knowledge of Java 8. If you are new to Java 8, we strongly recommend you start with a study guide to learn all of the facets of the language and come back to this book once you are thinking of taking the exam.

We recommend the best-selling *OCA Oracle Certified Associate Java SE 8 Programmer I Study Guide* and *OCP Oracle Certified Professional Java SE 8 Programmer II Study Guide*, which we happen to be the authors of, to start in your studies. Regardless of which study guide you are using to prepare, you can use this book to hone your skills, since it is based on topics on the actual exams.

Unlike the questions in our study guides, which are designed to be harder than the real exam, the questions in this book mirror the exam format. All the questions in this book tell you how many answers are correct. They will say “Choose two” or “Choose three” if more than one answer is correct.

Throughout this book, we use the same set of assumptions that Oracle uses for its exams. In many cases, these assumptions are actually accounting for unintended omissions or type-setting errors that Oracle is directing you to ignore when solving a question focused on a particular exam objective. The assumptions are listed at <http://education.oracle.com> and listed here for your convenience:

- **Missing package and import statements:** If sample code does not include package or import statements, and the question does not explicitly refer to these missing statements, then assume that all sample code is in the same package, and import statements exist to support them.
- **No file or directory path names for classes:** If a question does not state the file names or directory locations of classes, then assume one of the following, whichever will enable the code to compile and run:
  - All classes are in one file.
  - Each class is contained in a separate file, and all files are in one directory.
- **Unintended line breaks:** Sample code might have unintended line breaks. If you see a line of code that looks like it has wrapped, and this creates a situation where the wrapping is significant (for example, a quoted String literal has wrapped), assume that the wrapping is an extension of the same line, and the line does not contain a hard carriage return that would cause a compilation failure.
- **Code fragments:** A code fragment is a small section of source code that is presented without its context. Assume that all necessary supporting code exists, and that the supporting environment fully supports the correct compilation and execution of the code shown and its omitted environment.
- **Descriptive comments:** Take descriptive comments, such as “setter and getters go here,” at face value. Assume that correct code exists, compiles, and runs successfully to create the described effect.

# Choosing an Exam

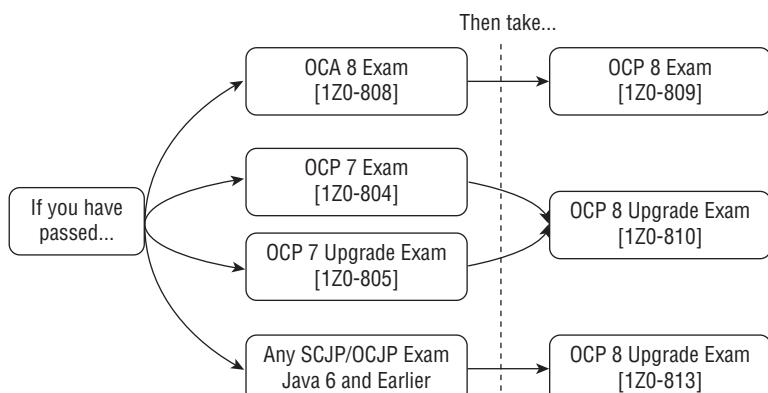
The first step on the road to a Java certification is the Oracle Certified Associate 1Z0-808 (OCA 8) exam. The material includes the basic day-to-day tools every Java developer should be familiar with. Once you have completed that exam, you can move on to the more difficult Oracle Certified Professional 1Z0-809 (OCP 8) exam. This includes every topic on the first exam, along with many additional libraries and APIs that you may not even be familiar with in your career, such as the Concurrency and NIO.2 APIs.

If you already hold a Java certification, you are eligible for an upgrade exam. Table I.1 and Figure I.1 show the various pathways to the OCP 8 certification title. This book will help you prepare for all four of the exams listed in the table.

**TABLE I.1** Exams this book covers

Exam Code	Name	Who should take
1Z0-808	Java Programmer I	Everyone can take this exam.
1Z0-809	Java Programmer II	Those who pass the OCA 8 exam
1Z0-810	Upgrade Java SE 7 to Java SE 8 OCP Programmer	Holders of an OCP 7 certification
1Z0-813	Upgrade to Java SE 8 OCP (Java SE 6 and all prior versions)	Holders of any of the following certifications: <ul style="list-style-type: none"> <li>▪ SCJP/OCJP 6</li> <li>▪ SCJP/OCJP 5</li> <li>▪ SCJP 1.4</li> </ul> Any older SCJP certs

**FIGURE I.1** Exam prerequisites



For the OCA exam, we recommend a study plan that uses our *OCA Oracle Certified Associated Java SE 8 Programmer I Study Guide* along with Chapters 1 through 10 of this book. For the OCP exam, we recommend a study plan that uses our *OCP Oracle Certified Professional Java SE 8 Programmer II Study Guide* along with Chapters 11 through 23 of this book. That's right, this book is actually two books in one! It assists you for both the OCA and OCP exams.

## Taking an Upgrade Exam

There are some subtle differences in how you should prepare if you are taking one of the upgrade exams. The good news is that if you currently hold a Java 7 OCP certification and are taking the 1Z0-810 upgrade exam, this book is all you need. While the exam contains a stronger focus on changes between Java 7 and Java 8, there aren't any new topics not covered with the existing OCP 7 material. This means you have the same study plan as those taking the OCP for the first time.

On the other hand, if you are taking the 1Z0-813 upgrade exam and hold a Sun Certified Java Programmer (SCJP) or Oracle Certified Java Programmer (OCJP) title, then you will need additional material. This upgrade inexplicably contains material that was in the Java 7 OCP exam but removed from the Java 8 version, such as ReentrantLock and WatchService. For that, we have no explanation, although this does make the 1Z0-813 arguably the most difficult of the three OCP exams because it requires knowing everything from multiple versions of Java.

Have no fear! When we wrote our *OCP Oracle Certified Professional Java SE 8 Programmer II Study Guide*, we added a special 50-page appendix, Appendix C, "Upgrading from Java 6 or Earlier," which covers the topics solely on the 1Z0-813 exam in detail. If you are taking the 1Z0-813 exam, then after completing your studies with this book, you should consult our appendix in our OCP book or a Java 7 OCP study guide so that you are prepared for all of the objectives.

## Considering Other Study Paths

If you already hold a Java certification, you are not required to take an upgrade exam. Some developers find it more straightforward to start with the easier 1Z0-808 OCA exam and then take the more challenging 1Z0-809 OCP exam. If you have no experience with Java 8 or just want more practice with it, taking the OCA exam followed by the OCP exam could be a better, albeit more expensive, learning experience than taking a single upgrade exam.

Also remember, if you do hold an old Java title and want to take an upgrade exam, you will need to prove it before you will be granted the OCP 8 title, even if you pass the upgrade exam. This requires registering your older certification with Oracle's CertView system. We've heard feedback from some readers who had to search through old binders and emails from over a decade ago to submit the proper documentation that allows them to be granted the OCP 8 title.

Finally, if you are new to Java certification, it is traditional to take the exams in order, with the OCA exam followed by the OCP exam. However, this is a not strict requirement. It is possible, although quite unorthodox, to take the harder OCP exam first and then circle back to take the OCA exam. In this case, you'll be granted both OCA and OCP titles upon completion of the OCA exam. While we don't recommend this strategy, Oracle does allow it.

## Who Should Buy This Book

If you are looking to take the OCA 8 exam (1Z0-808), then Chapters 1 through 10 of this book are for you. And once you've passed with flying colors, you'll have the OCP material handy in case you decide to take the exam.

If you are looking to take the OCP 8 exam (1Z0-809) or OCP 8 upgrade exam (1Z0-810), then Chapters 1 through 23 of this book are for you. While Chapters 1 through 10 are primarily focused on the OCA exam, the OCP exam is cumulative. We recommend you start reviewing the OCA Chapters (1–10) to make sure you have a solid foundation and then move on to the OCP Chapters (11–23) when you are ready.

As mentioned earlier, if you are looking to take the OCP 8 upgrade exam (1Z0-813), this book will still help for the vast majority of topics. You will need to supplement this book with a Java 7 study guide or use Appendix C in our OCP book, which focuses on the precise topics on the 1Z0-813 exam that you need to know.

Regardless of which exam you plan to take, make sure to always keep your study guide handy. This book is about honing your knowledge of Java 8, while your study guide is about building it.

## How This Book Is Organized

For this book, we decided to write two books in one, divided into Parts I and II. Part I includes Chapters 1 through 10, with nine objective-based chapters for each of the OCA exam objective sets followed by a simulated OCA practice exam. Part II encompasses Chapters 11 through 23, with 12 objective-based chapters for each of the OCP exam objective sets followed by a simulated OCP practice exam.

There are some subtle differences between the objective-based chapters and practice exam chapters that you should be aware of while reading this book.

## Using the Objective-Based Chapters

We designed the structure and style of each question in the objective-based chapters to reflect a more positive learning experience, allowing you to spend less time on each question but covering a broader level of material. For example, you may see two questions that

look similar within a chapter but contain a subtle difference that has drastic implications on whether or not the code compiles, or what output it produces.

Each question in the objective-based chapters has exactly four options with only one correct answer. Just like the review questions in our study guide, these questions are designed so that you can answer them many times. While these questions may be easier than exam questions, they will reinforce concepts if you keep taking them on a topic you don't feel strongly on.

In our study guides, we often group related topics into chapters. For example, if/then statements and loops, which are in separate Oracle objective sets, were presented in a single chapter on Operators and Statements in our OCA book. In this book, we decided to design our chapters solely around Oracle's objectives so you can strengthen your skills. While you don't need to read an entire study guide before using an objective-based chapter in this book, you do need to study the relevant objectives. Tables I.2 and I.3 show what chapters you need to have read in our study guides as a minimum before practicing with the questions in this book.

**TABLE I.2** Reference in OCA study guide

Chapter in This Book	Objectives	OCA Study Guide Chapters
1	Java Basics	1
2	Working with Java Data Types	1, 3
3	Using Operators and Decision Constructs	2, 3
4	Creating and Using Arrays	3
5	Using Loop Constructs	2
6	Working with Methods and Encapsulation	4
7	Working with Inheritance	5
8	Handling Exceptions	6
9	Working with Selected Classes from the Java API	1, 3, 4

**TABLE I.3** Reference in OCP study guide

Chapter in This Book	Objectives	OCP Study Guide Chapters
11	Java Class Design	1, 2
12	Advanced Java Class Design	1, 2, 3
13	Generics and Collections	3, 4
14	Lambda Built-in Functional Interfaces	2, 4
15	Java Stream API	3, 4
16	Exceptions and Assertions	6
17	Use Java SE 8 Date/Time API	5
18	Java I/O Fundamentals	8
19	Java File I/O (NIO.2)	9
20	Java Concurrency	7
21	Building Database Applications with JDBC	10
22	Localization	5

## Using the Practice Exam Chapters

This book contains two full-length practice exam chapters, with Chapter 10 being an 80-question OCA practice exam and Chapter 23 being an 85-question OCP practice exam. The questions in these two chapters are quite different from the objective-based chapters in a number of important ways. These practice exam questions tend to be harder because they are designed to test your cumulative knowledge rather than reinforcing your existing skill set.

While all of the objective-based chapters had four options with only one correct answer, these questions have up to six options, with up to three correct answer choices. Based on feedback from our first two books, we do indicate exactly how many answers are correct in the practice exam chapters, as is done on the real exam. Some readers thought the lack of knowing the correct number of answers made the questions too challenging for studying.

Both practice exam chapters are designed to be taken within 150 minutes and have a passing score of 65 percent. Remember not to take the practice exam until you feel ready. There are only so many practice exams available, so you don't want to waste a fresh attempt.

While an objective-based chapter can be completed over the course of a few days, the practice exam chapters were designed to be completed in one sitting. You should try simulating the exam experience as much as possible. This means setting aside two and a half hours, grabbing a whiteboard or scrap paper, and answering every question even if you aren't sure of the answer. Remember, there is no penalty for guessing, and the more incorrect answers you can eliminate the better.

## Reviewing Exam Changes

Oracle does change the number of questions, passing score, and time limit from time to time. In fact, the exam writers changed the number of OCA 8 questions from 77 to 80 while this book was being written! Scott and Jeanne maintain a blog that tracks updates to the real exams, as quickly as Oracle updates them:

<https://www.selikoff.net/jpt>

We recommend you read this page before you take the real exam, in case any of the information since the time this book was published has changed. Although less common, Oracle does add, remove, or reword objectives. When this happens, we offer free supplemental material on our website as blog entries.

## Ready to Take the Exam

If you can score above 80 percent consistently on all of the chapters related to the exam you want to take, including the simulated practice exam, then you are probably ready to take the real exam. Just remember there's a big difference between taking a practice test at home and spending hundreds of dollars to take a real exam at a test center.

We could write an entire chapter on test taking skills and study tips. Oh wait, we did! Both our OCA 8 and OCP 8 books each contain an appendix chock-full of helpful tips and suggestions that are designed to help you manage your time. They also include notes on how to eliminate obviously wrong answers so that when you have to guess, you're choosing between two choices and not five.

Finally, although a lot of people are inclined to cram as much material as they can in the hours leading up to the exam, most studies have shown that this is a poor test-taking strategy. The best thing we can recommend that you do before the exam is to get a good night's rest!

## Need More Help Preparing?

Both of the authors are moderators at CodeRanch.com, a very large and active programming forum that is very friendly toward Java beginners. It has separate forums for each of the exams:

OCA Forum: <https://coderanch.com/f/117>

OCP Forum: <https://coderanch.com/f/24>

If you don't understand a question, even after reading the explanation, feel free to ask about it in one of those forums. You'll get an answer from a knowledgeable Java programmer. It might even be one of us.

Good luck on the exam and happy studies!

## Bonus Contents

This book has a web page that provides all the questions in this book using Wiley's interactive online test engine.



---

You can link to this from [www.wiley.com/go/sybextestprep](http://www.wiley.com/go/sybextestprep).

# OCA

# PART

I





# Chapter

1



# Java Basics

---

**THE OCA EXAM TOPICS COVERED  
IN THIS PRACTICE TEST INCLUDE  
THE FOLLOWING:**

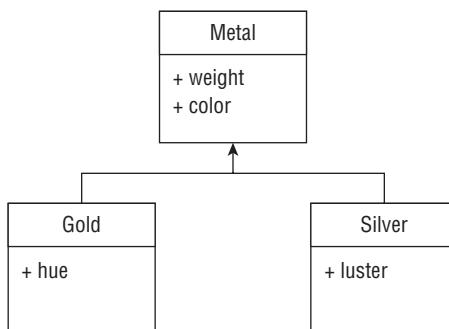
✓ **Java Basics**

- Define the scope of variables
- Define the structure of a Java class
- Create executable Java applications with a main method; run a Java program from the command line; produce console output
- Import other Java packages to make them accessible in your code
- Compare and contrast the features and components of Java such as: platform independence, object orientation, encapsulation, etc.

1. Which of the following method signatures is a valid declaration of an entry point in a Java application?

- A. public void main(String[] args)
- B. public static void main()
- C. private static void start(String[] mydata)
- D. public static final void main(String[] mydata)

2. The following class diagram demonstrates the relationship between Gold and Silver, which extend the Metal class. Assume the attributes are all declared public. Which statement about the diagram is not true?



- A. The diagram demonstrates platform independence in Java.
  - B. The diagram demonstrates object-oriented design in Java.
  - C. The Gold and Silver classes inherit weight and color attributes from the Metal class.
  - D. Gold does not inherit the luster attribute.
3. What is the proper filename extension for a Java bytecode compiled file?
- A. .java
  - B. .bytecode
  - C. .class
  - D. .dll
4. Given that a Date class exists in both the java.util and java.sql packages, what is the result of compiling the following class?

```

1: import java.util.*;
2: import java.sql.*;
3: public class BirthdayManager {
4:     private Date rob = new Date();
5:     private java.util.Date sharon = new java.util.Date();
6: }
  
```

- A. The code does not compile because of lines 1 and 2.
  - B. The code does not compile because of line 4.
  - C. The code does not compile because of line 5.
  - D. The code compiles without issue.
5. Which of the following is not a facet of traditional object-oriented programming languages?
- A. Objects are grouped as procedures, separate from the data they act on.
  - B. An object can take many forms via casting.
  - C. An object can hold data, referred to as attributes.
  - D. An object can perform actions, via methods.
6. Which variables have a scope limited to a method?
- A. Interface variables
  - B. Class variables
  - C. Instance variables
  - D. Local variables
7. Which package is imported into every Java class by default?
- A. java.util
  - B. java.lang
  - C. system.lang
  - D. java.system
8. Which of the following is not a valid code comment in Java?
- A. // Add 5 to the result
  - B. /\*\* TODO: Fix bug 12312 \*\*/
  - C. # Add configuration value
  - D. /\* Read file from system \*\*\*\*/
9. Which statement about a valid .java file is true?
- A. It can only contain one class declaration.
  - B. It can contain one public class declaration and one public interface definition.
  - C. It must define at least one public class.
  - D. It may define at most one public class.
10. Given the following application, fill in the missing values in the table starting from the top and going downward.

```
package competition;
public class Robot {
    static String weight = "A lot";
    /* default */ double ageMonths = 5, ageDays = 2;
```

```
private static boolean success = true;  
public void main(String[] args) {  
    final String retries = "1";  
    // P1  
}  
}
```

---

**Variable Type      Number of Variables Accessible at P1**

---

Class \_\_\_\_\_

Instance \_\_\_\_\_

Local \_\_\_\_\_

---

- A. 2, 0, 1
- B. 2, 2, 1
- C. 1, 0, 1
- D. 0, 2, 1

11. Which statement about `import` statements is true?

- A. The class will not compile if it contains unused `import` statements.
- B. Unused `import` statements can be removed from the class without causing a class to become unable to be compiled.
- C. The class will not compile if a duplicate `import` statement is present.
- D. If a class contains an `import` statement for a class used in the program that cannot be found, it can still compile.

12. What is the result of compiling and executing the following class?

```
1: public class ParkRanger {  
2:     int birds = 10;  
3:     public static void main(String[] data) {  
4:         int trees = 5;  
5:         System.out.print(trees+birds);  
6:     }  
7: }
```

- A. It does not compile.
- B. It compiles but throws an exception at runtime.
- C. It compiles and outputs 5.
- D. It compiles and outputs 15.

- 13.** Which statements about Java are true?
- I.** The `java` command can execute `.java` and `.class` files.
  - II.** Java is not object oriented.
  - III.** The `javac` command compiles directly into native machine code.
- A.** I only
  - B.** III only
  - C.** II and III
  - D.** None are true.
- 14.** Which of the following lines of code is not allowed as the first line of a Java class file?
- A.** `import widget.*;`
  - B.** `// Widget Manager`
  - C.** `package sprockets;`
  - D.** `int facilityNumber;`
- 15.** Which one of the following statements is true about using packages to organize your code in Java?
- A.** Every class is required to include a package declaration.
  - B.** To create a new package, you need to add a `package.init` file to the directory.
  - C.** Packages allow you to limit access to classes, methods, or data from classes outside the package.
  - D.** It is not possible to restrict access to objects and methods within a package.
- 16.** Given that the current directory is `/user/home`, with an application Java file in `/user/home/Manager.java` that uses the default package, which are the correct commands to compile and run the application in Java?
- A.** `javac Manager`  
`java Manager`
  - B.** `javac Manager.java`  
`java Manager`
  - C.** `javac Manager`  
`java Manager.class`
  - D.** `javac Manager.java`  
`java Manager.class`
- 17.** Structuring a Java class such that only methods within the class can access its instance variables is referred to as \_\_\_\_\_.
- A.** platform independence
  - B.** object orientation
  - C.** inheritance
  - D.** encapsulation

18. What is the output of the following code snippet?

```
String tree = "pine";
int count = 0;
if (tree.equals("pine")) {
    int height = 55;
    count = count + 1;
}
System.out.print(height + count);
```

- A. 1
- B. 55
- C. 56
- D. It does not compile.

19. Which of the following is true of a Java bytecode file?

- A. It can be run on any computer with a compatible JVM.
- B. It can only be executed on the same type of computer that it was created on.
- C. It can be easily read and modified in a standard text editor.
- D. It requires the corresponding .java that created it to execute.

20. What is the correct character for terminating a statement in Java?

- A. A colon (:)
- B. An end-of-line character
- C. A tab character
- D. A semicolon (;)

21. What is the result of compiling and executing the following class?

```
1: public class Tolls {
2:     private static int yesterday = 1;
3:     int tomorrow = 10;
4:     public static void main(String[] args) {
5:         Tolls tolls = new Tolls();
6:         int today=20, tomorrow = 40;
7:         System.out.print(today + tolls.tomorrow + tolls.yesterday);
8:     }
9: }
```

- A. The code does not compile due to line 6.
- B. The code does not compile due to line 7.
- C. 31
- D. 61

- 22.** Given the following class definition, which is the only line that does not contain a compilation error?

```
1: public ThisClassDoesNotCompile {  
2:     double int count;  
3:     void errors() {}  
4:     static void private limit; }
```

- A.** Line 1
- B.** Line 2
- C.** Line 3
- D.** Line 4

- 23.** Which of the following features allows a Java class to be run on a wide variety of computers and devices?

- A.** Encapsulation
- B.** Object oriented
- C.** Inheritance
- D.** Platform independence

- 24.** Which of the following is not a property of a JVM?

- A.** It prevents Java bytecode from being easily decoded/decompiled.
- B.** It supports platform independence.
- C.** It manages memory for the application.
- D.** It translates Java instructions to machine instructions.

- 25.** Which of the following variables are always in scope for the entire program?

- A.** Package variables
- B.** Class variables
- C.** Instance variables
- D.** Local variables

- 26.** Given the following wildcard import statements, which class would be included in the import?

```
import television.actor.*;  
import movie.director.*;  
  
A. television.actor.recurring.Marie  
B. movie.directors.John  
C. television.actor.Package  
D. movie.NewRelease
```

27. Which is the correct order of statements for a Java class file?
- A. import statements, package statement, class declaration
  - B. package statement, class declaration, import statement
  - C. class declaration, import statements, package declaration
  - D. package statement, import statements, class declaration
28. Given the following class definition, what is the maximum number of `import` statements that can be discarded and still have the code compile? For this question, assume that the `Blackhole` class is defined only in the `stars` package.

```
package planetarium;  
import java.lang.*;  
import stars.*;  
import java.lang.Object;  
import stars.Blackhole;  
  
public class Observer {  
    public void find(Blackhole blackhole) {}  
}
```

- A. Zero
  - B. One
  - C. Two
  - D. Three
29. Given the following class definition, which command will cause the application to output the message `White-tailed`?

```
package forest;  
public class Deer {  
    public static void main(String... deerParams) {  
        System.out.print(theInput[2]);  
    }  
}
```

- A. `java forest.Deer deer 5 "White-tailed deer"`
  - B. `java forest.Deer "White-tailed deer" deer 3`
  - C. `java forest.Deer Red deer White-tailed deer`
  - D. `java forest.Deer My "deer White-tailed"`
30. Which of the following is a true statement?
- A. The `java` command compiles a `.java` file into a `.class` file.
  - B. The `javac` command compiles a `.java` file into a `.class` file.
  - C. The `java` command compiles a `.class` file into a `.java` file.
  - D. The `javac` command compiles a `.class` file into a `.java` file.

31. Which of the following statements about Java is true?

- A. Java is a procedural programming language.
- B. Java allows method overloading.
- C. Java allows operator overloading.
- D. Java allows direct access to objects in memory.

32. Given the following code, what values inserted in order into the blank lines, allow the code to compile?

```
_____ agent;  
public _____ Banker {  
    private static _____ getMaxWithdrawal() {  
        return 10;  
    }  
}
```

- A. import, class, null
- B. import, interface, void
- C. package, int, int
- D. package, class, long

33. What is the output of the following application?

```
public class Airplane {  
    static int start = 2;  
    final int end;  
    public Airplane(int x) {  
        x = 4;  
        end = x;  
    }  
    public void fly(int distance) {  
        System.out.print(end-start+" ");  
        System.out.print(distance);  
    }  
    public static void main(String... start) {  
        new Airplane(10).fly(5);  
    }  
}
```

- A. 2 5
- B. 8 5
- C. 6 5
- D. The code does not compile.

- 34.** What is one of the most important reasons that Java supports extending classes via inheritance?
- A.** Inheritance requires that a class that extends another class be in the same package.
  - B.** The program must spend extra time/resources at runtime jumping through multiple layers of inheritance to determine precise methods and variables.
  - C.** Method signature changes in parent classes may break subclasses that use overloaded methods.
  - D.** Developers minimize duplicate code in new classes by sharing code in a common parent class.
- 35.** Which of the following is a valid code comment in Java?
- A.** ////////////// Walk my dog
  - B.** #! Go team!
  - C.** / Process fails at runtime /
  - D.** None of the above
- 36.** Which of the following method signatures is not a valid declaration of an entry point in a Java application?
- A.** public static void main(String... arguments)
  - B.** public static void main(String arguments)
  - C.** public static final void main(String[] arguments)
  - D.** public static void main(String[] arguments)
- 37.** Given the file Magnet.java below, which of the marked lines can you independently insert the line `public String color;` into and still have the code compile?
- ```
// line a1
public class Magnet {
    // line a2
    public void attach() {
        // line a3
    }
    // line a4
}
```
- A.** a1 and a3
  - B.** a2 and a4
  - C.** a2, a3, and a4
  - D.** a1, a2, a3, and a4
- 38.** What is required to define a valid Java class file?
- A.** A class declaration
  - B.** A package statement
  - C.** At least one import statement
  - D.** The public modifier

39. What is the proper filename extension for a Java source file?

- A. .jav
- B. .class
- C. .source
- D. .java

40. Given that a Math class exists in both the `java.lang` and `pocket.complex` packages, what is the result of compiling the following class?

```
1: package pocket;
2: import pocket.complex.*;
3: import java.util.*;
4: public class Calculator {
5:     public static void main(String[] args) {
6:         System.out.print(Math.floor(5));
7:     }
8: }
```

- A. The code does not compile because of line 2.
- B. The code does not compile because of line 3.
- C. The code does not compile because of line 6.
- D. The code compiles without issue.

41. Given a class that uses the following `import` statements, which class would not be automatically accessible within the class without using its full package name?

```
import dog.*;
import dog.puppy.*;
```

- A. dog.puppy.female.KC
- B. dog.puppy.Georgette
- C. dog.Webby
- D. java.lang.Object

42. \_\_\_\_\_ is the technique of structuring programming data as a unit consisting of attributes, with actions defined on the unit.

- A. Encapsulation
- B. Object orientation
- C. Platform independence
- D. Polymorphism

- 43.** Given the following class definition, what is the maximum number of `import` statements that can be discarded and still have the code compile? For this question, assume that the `Broccoli` class is in the `food.vegetables` package, and the `Apple` class is in the `food.fruit` package.

```
package food;
import food.vegetables.*;
import food.fruit.*;
import java.util.Date;

public class Grocery {
    Apple a; Broccoli b; Date c;
}
```

- A.** 0
- B.** 1
- C.** 2
- D.** 3

- 44.** Given the following application, what is the expected output?

```
public class Keyboard {
    private boolean numLock = true;
    static boolean capLock = false;
    public static void main(String... shortcuts) {
        System.out.print(numLock+" "+capLock);
    }
}
```

- A.** true false
- B.** false false
- C.** It does not compile.
- D.** It compiles but throws an exception at runtime.

- 45.** What is the result of compiling and executing the following class?

```
public class RollerSkates {
    static int wheels = 1;
    int tracks = 5;
    public static void main(String[] arguments) {
        RollerSkates s = new RollerSkates();
        int feet=4, tracks = 15;
        System.out.print(feet + tracks + s.wheels);
    }
}
```

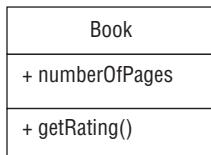
- A. The code does not compile.
  - B. 5
  - C. 10
  - D. 20
46. What is the result of compiling and executing the following class?

```
package sports;
public class Bicycle {
    String color = "red";
    private void printColor(String color) {
        color = "purple";
        System.out.print(color);
    }
    public static void main(String[] rider) {
        new Bicycle().printColor("blue");
    }
}
```

- A. red
  - B. purple
  - C. blue
  - D. It does not compile.
47. Which statements about calling the compilation command `javac` and the execution command `java` are true?
- I. java may use a period `.` to separate packages.
  - II. javac takes a `.java` file and returns a `.class` file.
  - III. java may use a slash `(/)` to separate packages.
- A. I only
  - B. II only
  - C. I and II
  - D. I, II, and III
48. What is the result of compiling and executing the following application?

```
package forecast;
public class Weather {
    private static boolean heatWave = true;
    public static void main() {
        boolean heatWave = false;
        System.out.print(heatWave);
    }
}
```

- A. true
  - B. false
  - C. It does not compile.
  - D. It compiles but throws an error at runtime.
49. Given the following class diagram, which Java implementation most closely matches this structure?



- A. 

```
public class Book {  
    public int numOfPages;
```
  - B. 

```
public class Book {  
    public String getRating() {return null;} }
```
  - C. 

```
public class Book {  
    public int numberOfPages;  
    public String getRating() {return null;} }
```
  - D. 

```
public class Book {  
    void numOfPages; }
```
50. Which statement about the JVM is true?
- A. The JVM schedules garbage collection on a predictable schedule.
  - B. The JVM ensures that the application will always terminate.
  - C. The JVM requires a properly defined entry point method to execute the application.
  - D. A Java compiled code can be run on any computer.

# Chapter 2



# Working with Java Data Types

---

**THE OCA EXAM TOPICS COVERED  
IN THIS PRACTICE TEST INCLUDE  
THE FOLLOWING:**

✓ **Working with Java Data Types**

- Declare and initialize variables (including casting of primitive data types)
- Differentiate between object reference variables and primitive variables
- Know how to read or write to object fields
- Explain an Object's Lifecycle (creation, "dereference by reassignment" and garbage collection)
- Develop code that uses wrapper classes such as Boolean, Double, and Integer

1. Which of the following declarations does not compile?

- A. double num1, int num2 = 0;
- B. int num1, num2;
- C. int num1, num2 = 0;
- D. int num1 = 0, num2 = 0;

2. What is the output of the following?

```
public static void main(String... args) {  
    String chair, table = "metal";  
    chair = chair + table;  
    System.out.println(chair);  
}
```

- A. metal
- B. metalmetal
- C. nullmetal
- D. The code does not compile.

3. Which is correct about an instance variable of type String?

- A. It defaults to an empty string.
- B. It defaults to null.
- C. It does not have a default value.
- D. It will not compile without initializing on the declaration line.

4. Which of the following is not a valid variable name?

- A. \_blue
- B. 2blue
- C. blue\$
- D. Blue

5. Which of these class names best follows standard Java naming conventions?

- A. fooBar
- B. FooBar
- C. FOO\_BAR
- D. F\_o\_o\_B\_a\_r

6. How many of the following methods compile?

```
public String convert(int value) {  
    return value.toString();  
}  
public String convert(Integer value) {  
    return value.toString();
```

```
    }
    public String convert(Object value) {
        return value.toString();
    }
```

- A.** None
  - B.** One
  - C.** Two
  - D.** Three
- 7.** Which of the following does not compile?
- A.** int num = 999;
  - B.** int num = 9\_9\_9;
  - C.** int num = \_9\_99;
  - D.** None of the above; they all compile.
- 8.** Which of the following is a wrapper class?
- A.** int
  - B.** Int
  - C.** Integer
  - D.** Object
- 9.** What is the result of running this code?
- ```
public class Values {
    integer a = Integer.valueOf("1");
    public static void main(String[] nums) {
        integer a = Integer.valueOf("2");
        integer b = Integer.valueOf("3");
        System.out.println(a + b);
    }
}
```
- A.** 4
  - B.** 5
  - C.** The code does not compile.
  - D.** The code compiles but throws an exception at runtime.
- 10.** Which best describes what the new keyword does?
- A.** Creates a copy of an existing object and treats it as a new one
  - B.** Creates a new primitive
  - C.** Instantiates a new object
  - D.** Switches an object reference to a new one

11. Which is the first line to trigger a compiler error?

```
double d1 = 5f;      // p1
double d2 = 5.0;    // p2
float f1 = 5f;      // p3
float f2 = 5.0;    // p4
```

- A. p1
- B. p2
- C. p3
- D. p4

12. Which of the following lists of primitive types are presented in order from smallest to largest data type?

- A. byte, char, float, double
- B. byte, char, double, float
- C. char, byte, float, double
- D. char, double, float, bigint

13. Which of the following is not a valid order for elements in a class?

- A. Constructor, instance variables, method names
- B. Instance variables, constructor, method names
- C. Method names, instance variables, constructor
- D. None of the above: all orders are valid.

14. Which of the following lines contains a compiler error?

```
String title = "Weather";           // line x1
int hot, double cold;              // line x2
System.out.println(hot + " " + title); // line x3
```

- A. x1
- B. x2
- C. x3
- D. None of the above

15. How many instance initializers are in this code?

```
1: public class Bowling {
2:     { System.out.println(); }
3:     public Bowling () {
4:         System.out.println();
5:     }
6:     static { System.out.println(); }
```

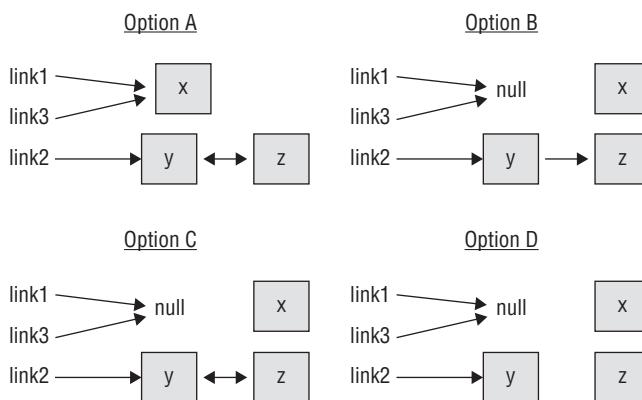
```
7:     { System.out.println(); }  
8: }
```

- A. None
  - B. One
  - C. Two
  - D. Three
16. Of the types double, int, and short, how many could fill in the blank to have this code output 0?
- ```
public static void main(String[] args) {  
    _____ defaultValue;  
    System.out.println(defaultValue);  
}
```
- A. None
  - B. One
  - C. Two
  - D. Three
17. What is true of the finalize() method?
- A. It may be called zero or one times.
  - B. It may be called zero or more times.
  - C. It will be called exactly once.
  - D. It may be called one or more times.
18. Which of the following is not a wrapper class?
- A. Double
  - B. Integer
  - C. Long
  - D. String
19. Suppose you have the following code. Which of the images best represents the state of the references right before the end of the main method, assuming garbage collection hasn't run?

```
1: public class Link {  
2:     private String name;  
3:     private Link next;  
4:     public Link(String name, Link next) {  
5:         this.name = name;  
6:         this.next = next;  
7:     }
```

```

8:     public void setNext(Link next) {
9:         this.next = next;
10:    }
11:    public Link getNext() {
12:        return next;
13:    }
14:    public static void main(String... args) {
15:        Link link1 = new Link("x", null);
16:        Link link2 = new Link("y", link1);
17:        Link link3 = new Link("z", link2);
18:        link2.setNext(link3);
19:        link3.setNext(link2);
20:        link1 = null;
21:        link3 = null;
22:    }
23: }
```



- A. Option A  
 B. Option B  
 C. Option C  
 D. Option D
20. Which type can fill in the blank?

\_\_\_\_\_ pi = 3.14;

- A. byte  
 B. float  
 C. double  
 D. short

**21.** What is the first line in the following code to not compile?

```
public static void main(String[] args) {  
    int Integer = 0;          // k1  
    Integer int = 0;          // k2  
    Integer ++;              // k3  
    int++;                  // k4  
}
```

- A.** k1
- B.** k2
- C.** k3
- D.** k4

**22.** Suppose foo is a reference to an instance of a class. Which of the following is not true about foo.bar?

- A.** bar is an instance variable.
- B.** bar is a local variable.
- C.** It can be used to read from bar.
- D.** It can be used to write to bar.

**23.** Which of the following is not a valid class declaration?

- A.** class building {}
- B.** class Cost\$ {}
- C.** class 5MainSt {}
- D.** class \_Outside {}

**24.** Which of the following can fill in the blanks to make this code compile?

\_\_\_\_\_ d = new \_\_\_\_\_(1\_000\_000\_.00);

- A.** double, double
- B.** double, Double
- C.** Double, double
- D.** None of the above

**25.** Which is correct about a local variable of type String?

- A.** It defaults to an empty string.
- B.** It defaults to null.
- C.** It does not have a default value.
- D.** It will not compile without initializing on the declaration line.

26. Of the types double, int, long, and short, how many could fill in the blank to have this code output 0?

```
static _____ defaultValue;  
  
public static void main(String[] args) {  
    System.out.println(defaultValue);  
}
```

- A. One
- B. Two
- C. Three
- D. Four

27. Which of the following is true about primitives?

- A. You can call methods on a primitive.
- B. You can convert a primitive to a wrapper class object simply by assigning it.
- C. You can convert a wrapper class object to a primitive by calling valueOf().
- D. You can store a primitive directly into an ArrayList.

28. What is the output of the following?

```
Integer integer = new Integer(4);  
System.out.print(integer.byteValue());  
  
System.out.print("-");
```

```
int i = new Integer(4);  
System.out.print(i.byteValue());
```

- A. 4-0
- B. 4-4
- C. The code does not compile.
- D. The code compiles but throws an exception at runtime.

29. Given the following code, fill in the blank to have the code print bounce.

```
public class TennisBall {  
    public TennisBall() {  
        System.out.println("bounce");  
    }  
    public static void main(String[] slam) {  
        _____  
    }  
}
```

- A. TennisBall;
  - B. TennisBall();
  - C. new TennisBall;
  - D. new TennisBall();
30. Which of the following correctly assigns animal to both variables?
- I. String cat = "animal", dog = "animal";
  - II. String cat = "animal"; dog = "animal";
  - III. String cat, dog = "animal";
  - IV. String cat, String dog = "animal";
- A. I
  - B. I, II
  - C. I, III
  - D. I, II, III, IV
31. Which two primitives have wrapper classes that are not merely the name of the primitive with an uppercase letter?
- A. byte and char
  - B. byte and int
  - C. char and int
  - D. None of the above
32. Which of the following is true about String instance variables?
- A. They can be set to null.
  - B. They can never be set from outside the class they are defined in.
  - C. They can only be set in the constructor.
  - D. They can only be set once per run of the program.
33. Which statement is true about primitives?
- A. Primitive types begin with a lowercase letter.
  - B. Primitive types can be set to null.
  - C. String is a primitive.
  - D. You can create your own primitive types.
34. How do you force garbage collection to occur at a certain point?
- A. Call System.forceGc()
  - B. Call System.gc()
  - C. Call System.requireGc()
  - D. None of the above

35. How many of the `String` objects are eligible for garbage collection right before the end of the `main` method?

```
public static void main(String[] fruits) {  
    String fruit1 = new String("apple");  
    String fruit2 = new String("orange");  
    String fruit3 = new String("pear");  
  
    fruit3 = fruit1;  
    fruit2 = fruit3;  
    fruit1 = fruit2;  
}
```

- A. None
- B. One
- C. Two
- D. Three

36. Which of the following can fill in the blanks to make this code compile?

```
_____ d = new _____(1_000_000.00);
```

- A. double, double
- B. double, Double
- C. Double, double
- D. None of the above

37. What does the following output?

```
1: public class InitOrder {  
2:     public String first = "instance";  
3:     public InitOrder() {  
4:         first = "constructor";  
5:     }  
6:     { first = "block"; }  
7:     public void print() {  
8:         System.out.println(first);  
9:     }  
10:    public static void main(String... args) {  
11:        new InitOrder().print();  
12:    }  
13: }
```

- A. block  
B. constructor  
C. instance  
D. The code does not compile.
38. How many of the following lines compile?
- ```
int i = null;
Integer in = null;
String s = null;
```
- A. None  
B. One  
C. Two  
D. Three
39. Which pairs of statements can accurately fill in the blanks in this table?
- | Variable Type | Can be called within the class from what type of method |
|---------------|---|
| Instance      | Blank 1: _____  |
| Static        | Blank 2: _____  |
- A. Blank 1: an instance method only, Blank 2: a static method only  
B. Blank 1: an instance or static method, Blank 2: a static method only  
C. Blank 1: an instance method only, Blank 2: an instance or static method  
D. Blank 1: an instance or static method, Blank 2: an instance or static method
40. Which of the following does not compile?
- A. double num = 2.718;  
B. double num = 2.\_718;  
C. double num = 2.7\_1\_8;  
D. None of the above; they all compile.
41. Which of the following lists of primitive numeric types is presented in order from smallest to largest data type?
- A. byte, short, int, long  
B. int, short, byte, long  
C. short, byte, int, long  
D. short, int, byte, long

**42.** Fill in the blank to make the code compile:

```
package animal;
public class Cat {
    public String name;
    public static void main(String[] meow) {
        Cat cat = new Cat();
        _____ = "Sadie";
    }
}
```

- A. cat.name
- B. cat-name
- C. cat.setName
- D. cat[name]

**43.** Which of the following is the output of this code, assuming it runs to completion?

```
package store;
public class Toy {
    public void play() {
        System.out.print("play-");
    }
    public void finalizer() {
        System.out.print("clean-");
    }
    public static void main(String[] fun) {
        Toy car = new Toy();
        car.play();
        System.gc();
        Toy doll = new Toy();
        doll.play();
    }
}
```

- A. play-
- B. play-play-
- C. play-clean-play-
- D. play-play-clean-clean-

44. Which is the most common way to fill in the blank to implement this method?

```
public class Penguin {  
    private double beakLength;  
    public static void setBeakLength(Penguin p, int b) {  
        _____  
    }  
}
```

- A. p.beakLength = b;
- B. p['beakLength'] = b;
- C. p[beakLength] = b;
- D. None of the above

45. Fill in the blanks to indicate whether a primitive or wrapper class can be assigned without the compiler using the autoboxing feature.

```
_____ first = Integer.parseInt("5");  
_____ second = Integer.valueOf("5");
```

- A. int, int
- B. int, Integer
- C. Integer, int
- D. Integer, Integer

46. How many objects are eligible for garbage collection right before the end of the `main` method?

```
1:  public class Person {  
2:      public Person youngestChild;  
3:  
4:      public static void main(String... args) {  
5:          Person elena = new Person();  
6:          Person diana = new Person();  
7:          elena.youngestChild = diana;  
8:          diana = null;  
9:          Person zoe = new Person();  
10:         elena.youngestChild = zoe;  
11:         zoe = null;  
12:     }  
13: }
```

- A. None
- B. One
- C. Two
- D. Three

47. Which is a valid constructor for this class?

```
public class TennisBall {  
}
```

- A. public TennisBall static create() { return new TennisBall(); }
- B. public TennisBall static newInstance() { return new TennisBall():}
- C. public TennisBall() {}
- D. public void TennisBall() {}

48. Which of the following is not a possible output of this code, assuming it runs to completion?

```
package store;  
public class Toy {  
    public void play() {  
        System.out.print("play-");  
    }  
    public void finalize() {  
        System.out.print("clean-");  
    }  
    public static void main(String[] args) {  
        Toy car = new Toy();  
        car.play();  
        System.gc();  
        Toy doll = new Toy();  
        doll.play();  
    }  
}
```

- A. play-
- B. play-play-
- C. play-play-clean-
- D. play-play-clean-clean-

**49.** Which converts a primitive to a wrapper class object without using autoboxing?

- A.** Call the `asObject()` method
- B.** Call the constructor of the wrapper class
- C.** Call the `convertToObject()` method
- D.** Call the `toObject()` method

**50.** What is the output of the following?

```
package beach;
public class Sand {
    public Sand() {
        System.out.print("a");
    }
    public void Sand() {
        System.out.print("b");
    }
    public void run() {
        new Sand();
        Sand();
    }
    public static void main(String... args) {
        new Sand().run();
    }
}
```

- A.** a
- B.** ab
- C.** aab
- D.** None of the above



# Chapter 3



# Using Operators and Decision Constructs

---

**THE OCA EXAM TOPICS COVERED IN THIS PRACTICE TEST INCLUDE THE FOLLOWING:**

✓ **Using Operators and Decision Constructs**

- Use Java operators; use parentheses to override operator precedence
- Test equality between Strings and other objects using == and equals()
- Create if and if/else and ternary constructs
- Use a switch statement

1. Which of the following variable types is not permitted in a switch statement?

- A. String
- B. double
- C. int
- D. char

2. What is the value of tip after executing the following code snippet?

```
int meal = 5;
int tip = 2;
int total = meal + (meal>6 ? ++tip : --tip);
```

- A. 1
- B. 2
- C. 3
- D. 6

3. What is the output of the following application?

```
package registration;
public class NameCheck {
    public static void main(String... data) {
        String john = "john";
        String jon = new String(john);
        System.out.print((john==jon)+" "+(john.equals(jon)));
    }
}
```

- A. true true
- B. true false
- C. false true
- D. false false

4. What is the output of the following application?

```
package planning;
public class ThePlan {
    public static void main(String[] input) {
        int plan = 1;
        plan = plan++ + --plan;
        if(plan==1) {
            System.out.print("Plan A");
        } else { if(plan==2) System.out.print("Plan B");
        } else System.out.print("Plan C");
    }
}
```

- A. Plan A
  - B. Plan B
  - C. Plan C
  - D. None of the above
5. Which of the following statements about a `default` branch in a `switch` statement is correct?
- A. All `switch` statements must include a `default` statement.
  - B. The `default` statement is required to be placed after all `case` statements.
  - C. Unlike a `case` statement, the `default` statement does not take a value.
  - D. A `default` statement can only be used when at least one `case` statement is present.
6. What is the value of `thatNumber` after the execution of the following code snippet?
- ```
long thatNumber = 5 >= 5 ? 1+2 : 1*1;  
if(++thatNumber < 4)  
    thatNumber += 1;
```
- A. 3
  - B. 4
  - C. 5
  - D. The answer cannot be determined until runtime.
7. Which statement immediately exits a `switch` statement, skipping all remaining `case` or `default` branches?
- A. `exit`
  - B. `break`
  - C. `goto`
  - D. `continue`
8. Which statement about ternary expressions is true?
- A. In some cases, both expressions to the right of the conditional operator in a ternary expression will be evaluated at runtime.
  - B. Ternary expressions require parentheses for proper evaluation.
  - C. The ternary expressions are a convenient replacement for an `if-then-else` statement.
  - D. Ternary expressions support `int` and `boolean` expressions for the left-most operand.
9. What is the output of the following application?

```
package voting;  
1: public class Election {  
2:     public void calculateResult(Integer candidateA, Integer candidateB) {  
3:         boolean process = candidateA == null || candidateA.intValue() < 10;  
4:         boolean value = candidateA && candidateB;  
5:         System.out.print(process || value);
```

```
6:      }
7:      public static void main(String[] unused) {
8:          new Election().calculateResult(null,203);
9:      }
10: }
```

- A. true
- B. false
- C. The code does not compile.
- D. The code compiles but throws a `NullPointerException` on line 3 at runtime.

10. What is the output of the following application?

```
package dinosaur;
public class Park {
    public final static void main(String... arguments) {
        int pterodactyl = 6;
        long triceratops = 3;
        if(pterodactyl % 3 >= 1)
            triceratops++;
        triceratops--;
        System.out.print(triceratops);
    }
}
```

- A. 2
- B. 3
- C. 4
- D. The code does not compile.

11. Which statement about if-then statements is true?

- A. An if-then statement is required to have an `else` statement.
- B. If the boolean test of an if-then statement evaluates to `false`, then the target clause of the if-then statement will still be evaluated.
- C. An if-then statement is required to cast an object.
- D. An if-then statement can execute a single statement or a block {}.

12. What is the output of the following application?

```
package restaurant;
public class Pieces {
    public static void main(String[] info) {
        int flair = 15;
        if(flair >= 15 && flair < 37) {
            System.out.print("Not enough");
        }
    }
}
```

```
        } if(flair==37) {  
            System.out.print("Just right");  
        } else {  
            System.out.print("Too many");  
        }  
    }  
}
```

- A. Not enough  
B. Just right  
C. Too many  
D. None of the above
13. Which statement about case statements of a switch statement is not true?
- A. A case value can be final.  
B. A case statement must be terminated with a break statement.  
C. A case value can be a literal expression.  
D. A case value must match the data type of the switch variable, or be able to be promoted to that type.
14. Given the following truth table, which operator for the boolean expressions x and y corresponds to this relationship?

|                  | <b>x = true</b> | <b>x = false</b> |
|------------------|-----------------|------------------|
| <b>y = true</b>  | true            | false            |
| <b>y = false</b> | false           | false            |

- A. --  
B. ++  
C. ||  
D. &&
15. What is the output of the following code snippet?

```
int hops = 0;  
int jumps = 0;  
jumps = hops++;  
if(jumps)  
    System.out.print("Jump!");  
else  
    System.out.print("Hop!");
```

- A. Jump!
  - B. Hop!
  - C. The code does not compile.
  - D. The code compiles but throws an exception at runtime.
16. Fill in the blanks: The \_\_\_\_\_ operator increases the value of a variable by 1 and returns the new value, while the \_\_\_\_\_ operator decreases the value of a variable by 1 and returns the original value.
- A. pre-increment [++v], pre-decrement [--v]
  - B. pre-increment [++v], post-decrement [v--]
  - C. post-increment [v++], pre-decrement [--v]
  - D. post-increment [v++], post-decrement [v--]
17. What is the output of the following application?
- ```
package jungle;
public class TheBigRace {
    public static void main(String[] in) {
        int tiger = 2;
        short lion = 3;
        long winner = lion+2*(tiger + lion);
        System.out.print(winner);
    }
}
```
- A. 11
  - B. 13
  - C. 25
  - D. None of the above

18. Given the following code snippet, assuming dayOfWeek is an `int`, what variable type of `saturday` is not permitted?

```
final ____ saturday = 6;
switch(dayOfWeek) {
    default:
        System.out.print("Another Weekday");
        break;
    case saturday:
        System.out.print("Weekend!");
}
```

- A. byte
  - B. long
  - C. int
  - D. None of the above
19. Given the following code snippet, what is the value of dinner after it is executed?
- ```
int time = 11;
int day = 4;
String dinner = time > 10 ? day ? "Takeout" : "Salad" : "Leftovers";
```
- A. Takeout
  - B. Salad
  - C. The code does not compile but would compile if parentheses were added.
  - D. None of the above
20. What is the output of the following application?
- ```
package recreation;
public class Dancing {
    public static void main(String[] vars) {
        int leaders = 10 * (2 + (1 + 2 / 5));
        int followers = leaders * 2;
        System.out.print(leaders + followers < 10 ? "Too few" : "Too many");
    }
}
```
- A. Too few
  - B. Too many
  - C. The code does not compile.
  - D. The code compiles but throws a division by zero error at runtime.
21. What is the output of the following application?
- ```
package schedule;
public class PrintWeek {
    public static final void main(String[] days) {
        System.out.print(5 + 6 + "7" + 8 + 9);
    }
}
```
- A. 56789
  - B. 11789
  - C. 11717
  - D. The code does not compile.

22. Fill in the blanks: The \_\_\_\_\_ operator is used to find the difference between two numbers, while the \_\_\_\_\_ operator is used to find the remainder when one number is divided by another.

- A. /, %
- B. -, %
- C. %, <
- D. -, ||

23. What is the output of the following application?

```
package transporter;
public class Rematerialize {
    public static void main(String[] input) {
        int dog = 11;
        int cat = 3;
        int partA = dog / cat;
        int partB = dog % cat;
        int newDog = partB + partA * cat;
        System.out.print(newDog);
    }
}
```

- A. 9
- B. 11
- C. 15
- D. The code does not compile.

24. What is the output of the following application?

```
package dessert;
public class IceCream {
    public final static void main(String... args) {
        int flavors = 30;
        int eaten = 0;
        switch(flavors) {
            case 30: eaten++;
            case 40: eaten+=2;
            default: eaten--;
        }
        System.out.print(eaten);
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. The code does not compile.
25. What is the output of the following application?
- ```
package mode;
public class Transportation {
    public static String travel(int distance) {
        return distance<1000 ? "train" : 10;
    }
    public static void main(String[] answer) {
        System.out.print(travel(500));
    }
}
```
- A. train
  - B. 10
  - C. The code does not compile.
  - D. The code compiles but throws an exception at runtime.
26. Fill in the blanks: Given two non-null `String` objects with reference names `apples` and `oranges`, if `apples _____ oranges` evaluates to `true`, then `apples _____ oranges` must also evaluate to `true`.
- A. `==, equals()`
  - B. `!=, equals()`
  - C. `equals(), ==`
  - D. `equals(), !=`
27. For a given non-null `String myTestVariable`, what is the resulting value of executing the statement `myTestVariable.equals(null)`?
- A. `true`
  - B. `false`
  - C. The statement does not compile.
  - D. The statement compiles but will produce an exception when used at runtime.
28. How many 1s are outputted when the following application is compiled and run?
- ```
package city;
public class Road {
    public static void main(String... in) {
        int intersections = 100;
```

```
int streets = 200;
if (intersections < 150) {
    System.out.print("1");
} else if (streets && intersections > 1000) {
    System.out.print("2");
} if (streets < 500)
    System.out.print("1");
else
    System.out.print("2");
}
```

A. None

B. One

C. Two

D. The code does not compile.

**29.** Which statement about the logical operators & and && is true?

- A. The & and && operators are interchangeable, always producing the same results at runtime.
- B. The & operator always evaluates both operands, while the && operator may only evaluate the left operand.
- C. Both expressions evaluate to `true` if either operand is `true`.
- D. The & operator always evaluates both operands, while the && operator may only evaluate the right operand.

**30.** What is the output of the following code snippet?

```
int x = 10, y = 5;
boolean w = true, z = false;
x = w ? y++ : y--;
w = !z;
System.out.print((x+y)+" "+(w ? 5 : 10));
```

A. The code does not compile.

B. 10 10

C. 11 5

D. 12 5

**31.** What is the output of the following application?

```
package bob;
public class AreYouBob {
    public static void main(String[] unused) {
```

```
String bob = new String("bob");
String notBob = bob;
System.out.print((bob==notBob)+" "+(bob.equals(notBob)));
}
}

A. true true
B. true false
C. false true
D. false false
```

32. What is the value of  $12 + 6 * 3 \% (1 + 1)$  in Java?

A. 0  
B. 12  
C. 14  
D. None of the above

33. Given the following truth table, the boolean variables p and q, and the expression  $p \wedge q$ , what are the missing values in the truth table, starting with the first column?

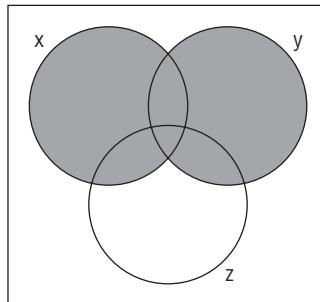
|                  | <b>p = true</b> | <b>p = false</b> |
|------------------|-----------------|------------------|
| <b>q = true</b>  | false           | true             |
| <b>q = false</b> |                 |                  |

A. false and true  
B. false and false  
C. true and true  
D. true and false

34. Which of the following is not a possible result of executing the following application?

```
public class ConditionallyLogical {
    public static void main(String... data) {
        if(data.length>=1
            && (data[0].equals("sound") || data[0].equals ("logic"))
            && data.length<2) {
            System.out.print(data[0]);
        }
    }
}
```

- A. Nothing is printed.  
B. sound is printed.  
C. The application throws an exception at runtime.  
D. logic is printed.
35. Fill in the blanks: The operators +, \_\_\_\_\_, \_\_\_\_\_, and ++ are listed in the same or increasing level of operator precedence.  
A. \*, --, /  
B. %, -, \*  
C. /, \*, %  
D. \*, -, /
36. What statement about the `^` operator is correct?  
A. If one of the operands of `^` is true, then the result is always true.  
B. There is a conditional form of the operator, denoted as `^^`.  
C. If both operands of `^` are true, the result is true.  
D. The `^` operator can only be applied to boolean values.
37. Given the following Venn diagram and the variables, `x`, `y`, and `z`, which Java expression most closely represents the filled-in region of the diagram?



- A. `x || z`  
B. `y || (y && z)`  
C. `x || y`  
D. `y && x`
38. What variable type of red allows the following application to compile?

```
package tornado;
public class Kansas {
    public static void main(String[] args) {
        int colorOfRainbow = 10;
        _____ red = 5;
```

```
switch(colorOfRainbow) {  
    default:  
        System.out.print("Home");  
        break;  
    case red:  
        System.out.print("Away");  
    }  
}  
}
```

- A. long  
B. double  
C. int  
D. None of the above
39. Which two operators would be used to test if a number is equal to or greater than 5.21 but strictly less than 8.1?
- A. > and <=   
B. >= and >   
C. < and >=   
D. < and >
40. What is the output of the following application?
- ```
package transporter;  
public class TurtleVsHare {  
    public static void main(String[] arguments) {  
        int turtle = 10 * (2 + (3 + 2) / 5);  
        int hare = turtle < 5 ? 10 : 25;  
        System.out.print(turtle < hare ? "Hare wins!" : "Turtle wins!");  
    }  
}
```
- A. Hare wins!  
B. Turtle wins!  
C. The code does not compile.  
D. The code compiles but throws a division by zero error at runtime.
41. What is the output of the following application?

```
public class CountEntries {  
    public static int getResult(int threshold) {  
        return threshold > 5 ? 1 : 0;
```

```
}

public static final void main(String[] days) {
    System.out.print(getResult(5)+getResult(1)
        +getResult(0)+getResult(2)+"");
}
}

A. 0
B. 1
C. 0000
D. 1000
```

- 42.** What is the output of the following application?

```
package yoyo;
public class TestGame {
    public String runTest(boolean spinner, boolean roller) {
        if(spinner = roller) return "up";
        else return roller ? "down" : "middle";
    }
    public static final void main(String pieces[]) {
        final TestGame tester = new TestGame();
        System.out.println(tester.runTest(false,true));
    }
}

A. up
B. middle
C. down
D. The code does not compile.
```

- 43.** Fill in the blanks: The \_\_\_\_\_ operator is true if either of the operands are true, while the \_\_\_\_\_ operator flips a boolean value.

**A.** +, -
**B.** &&, !
**C.** |, -
**D.** ||, !

- 44.** Given the following code snippet, what is the value of movieRating after it is executed?

```
int characters = 5;
int story = 3;
double movieRating = characters <= 4 ? 3 : story>1 ? 2 : 1;
```

- A. 2.0  
B. 3.0  
C. The code does not compile but would compile if parentheses were added.  
D. None of the above
45. Fill in the blanks: A switch statement can have \_\_\_\_\_ case statements and \_\_\_\_\_ default statements.
- A. at most one, at least one  
B. any number of, at most one  
C. at least one, any number of  
D. at least one, at most one
46. Which of the following is not a possible result of executing the following application?
- ```
public class OutsideLogic {  
    public static void main(String... weather) {  
        System.out.print(weather[0] != null  
                        && weather[0].equals("sunny")  
                        && !false  
                        ? "Go Outside" : "Stay Inside");  
    }  
}
```
- A. Nothing is printed.  
B. The application throws an exception at runtime.  
C. Go Outside is printed.  
D. Stay Inside is printed.
47. What is the value of  $(5 + (!2 + 8) * 3 - 3 \% 2)/2$  in Java?
- A. 2  
B. 11  
C. 16  
D. None of the above
48. Given the following truth table, the boolean variables w and z, and the expression  $w \mid\mid z$ , what are the missing values in the truth table, starting with the first row?

|           | w = true | w = false |
|-----------|----------|-----------|
| z = true  | true     |           |
| z = false |          | false     |

- A. false and false  
B. true and false  
C. true and true  
D. false and true
49. Fill in the blanks: The operators -, \_\_\_\_\_, \_\_\_\_\_, and % are listed in the same or increasing level of operator precedence.  
A. +, /, \*  
B. --, -, \*  
C. ++, /, \*  
D. \*, ++, %
50. What is the output of the following application?
- ```
public class Baby {  
    public static String play(int toy, int age) {  
        final String game;  
        if(toy<2)  
            game = age > 1 ? 1 : 10; // p1  
        else  
            game = age > 3 ? "Ball" : "Swim"; // p2  
        return game;  
    }  
    public static void main(String[] variables) {  
        System.out.print(play(5,2));  
    }  
}
```
- A. Ball  
B. Swim  
C. The code does not compile due to p1.  
D. The code does not compile due to p2.

# Chapter 4



## Creating and Using Arrays

---

**THE OCA EXAM TOPICS COVERED  
IN THIS PRACTICE TEST INCLUDE  
THE FOLLOWING:**

✓ **Creating and Using Arrays**

- Declare, instantiate, initialize and use a one-dimensional array
- Declare, instantiate, initialize and use multi-dimensional arrays

1. What symbol is used for a varargs method parameter?

A. ..  
B. ...  
C. --  
D. ---

2. Fill in the blank in the following code to get the first element from the varargs parameter.

```
public void toss (Frisbee... f) {  
    Frisbee first = _____;  
}
```

A. f  
B. f[0]  
C. f[1]  
D. None of the above

3. Which of the following are primitives?

```
int[] lowercase = new int[0];  
Integer[] uppercase = new Integer[0];
```

A. Only lowercase  
B. Only uppercase  
C. Both lowercase and uppercase  
D. Neither lowercase nor uppercase

4. How many of the following are legal declarations?

```
[]double lion;  
double[] tiger;  
double bear[];
```

A. None  
B. One  
C. Two  
D. Three

5. Given the following two methods, which method call will not compile?

```
public void printStormName(String... names) {  
    System.out.println(Arrays.toString(names));  
}  
public void printStormNames(String[] names) {  
    System.out.println(Arrays.toString(names));  
}
```

- A. `printStormName("Arlene");`
  - B. `printStormName(new String[] { "Bret" });`
  - C. `printStormNames("Cindy");`
  - D. `printStormNames(new String[] { "Don" });`
6. How do you determine the number of elements in an array?
- A. `buses.length`
  - B. `buses.length()`
  - C. `buses.size`
  - D. `buses.size()`
7. Which of the following create an empty two-dimensional array with dimensions 2x2?
- A. `int[][] blue = new int[2, 2];`
  - B. `int[][] blue = new int[2], [2];`
  - C. `int[][] blue = new int[2][2];`
  - D. `int[][] blue = new int[2 x 2];`
8. How many lines does the following code output?
- ```
String[] days = new String[] { "Sunday", "Monday", "Tuesday",
    "Wednesday", "Thursday", "Friday", "Saturday" };
for (int i = 0; i < days.length; i++)
    System.out.println(days[i]);
```
- A. Six
  - B. Seven
  - C. The code does not compile.
  - D. The code compiles but throws an exception at runtime.
9. What are the names of the methods to do searching and sorting respectively on arrays?
- A. `Arrays.binarySearch()` and `Arrays.linearSort()`
  - B. `Arrays.binarySearch()` and `Arrays.sort()`
  - C. `Arrays.search()` and `Arrays.linearSort()`
  - D. `Arrays.search()` and `Arrays.sort()`
10. What does this code output?
- ```
String[] nums = new String[] { "1", "9", "10" };
Arrays.sort(nums);
System.out.println(Arrays.toString(nums));
```
- A. `[1, 9, 10]`
  - B. `[1, 10, 9]`
  - C. `[10, 1, 9]`
  - D. None of the above

11. Which of the following references the first and last element in a non-empty array?

- A. trains[0] and trains[trains.length]
- B. trains[0] and trains[trains.length - 1]
- C. trains[1] and trains[trains.length]
- D. trains[1] and trains[trains.length - 1]

12. How many of the following are legal declarations?

```
String lion [] = new String[] {"lion"};
String tiger [] = new String[1] {"tiger"};
String bear [] = new String[] {};
String ohMy [] = new String[0] {};
```

- A. None
- B. One
- C. Two
- D. Three

13. How many of the following are legal declarations?

```
float[] lion = new float[];
float[] tiger = new float[1];
float[] bear = new[] float;
float[] ohMy = new[1] float;
```

- A. None
- B. One
- C. Two
- D. Three

14. Which statement most accurately represents the relationship between searching and sorting with respect to the Arrays class?

- A. If the array is not sorted, calling Arrays.binarySearch() will be accurate, but slower than if it were sorted.
- B. The array does not need to be sorted before calling Arrays.binarySearch() to get an accurate result.
- C. The array must be sorted before calling Arrays.binarySearch() to get an accurate result.
- D. None of the above

15. Which is not a true statement about an array?

- A. An array expands automatically when it is full.
- B. An array is allowed to contain duplicate values.
- C. An array understands the concept of ordered elements.
- D. An array uses a zero index to reference the first element.

16. Which line of code causes an `ArrayIndexOutOfBoundsException`?

```
String[][] matrix = new String[1][2];
matrix[0][0] = "Don't think you are, know you are.";           // m1
matrix[0][1] = "I'm trying to free your mind Neo";            // m2
matrix[1][0] = "Is all around you ";                            // m3
matrix[1][1] = "Why oh why didn't I take the BLUE pill?";    // m4
```

- A. m1
- B. m2
- C. m3
- D. m4

17. What does the following output?

```
String[] os = new String[] { "Mac", "Linux", "Windows" };
Arrays.sort(os);
System.out.println(Arrays.binarySearch(os, "Mac"));
```

- A. 0
- B. 1
- C. 2
- D. The output is not defined.

18. Which is the first line to prevent this code from compiling and running without error?

```
char[][] ticTacToe = new char[3,3];                      // r1
ticTacToe[1][3] = 'X';                                    // r2
ticTacToe[2][2] = 'X';
ticTacToe[3][1] = 'X';
System.out.println(ticTacToe.length + " in a row!"); // r3
```

- A. Line r1
- B. Line r2
- C. Line r3
- D. None of the above

19. How many objects are created when running the following code?

```
Integer[] lotto = new Integer[4];
lotto[0] = new Integer(1_000_000);
lotto[1] = new Integer(999_999);
```

- A. Two
- B. Three
- C. Four
- D. Five

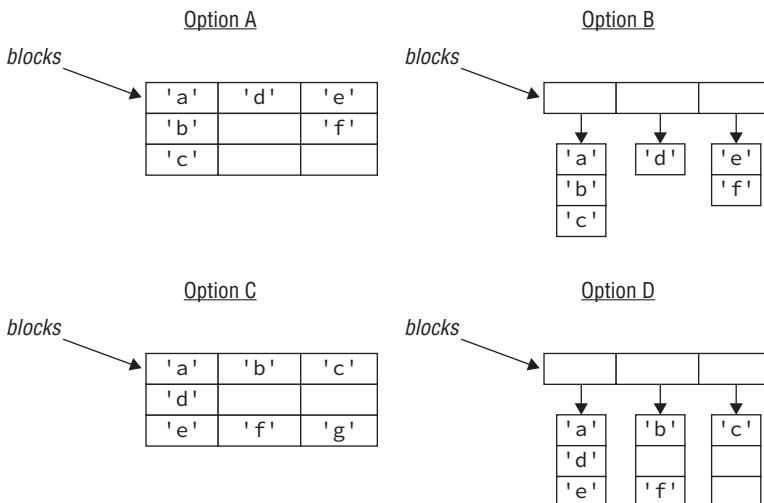
**20.** How many of the following are legal declarations?

```
[] [] String alpha;
[] String beta;
String[][] gamma;
String[] delta[];
String epsilon[][];
```

- A. Two
- B. Three
- C. Four
- D. Five

**21.** Which of the options in the graphic best represent the `blocks` variable?

```
char[][] blocks = new char[][] { { 'a', 'b', 'c' }, { 'd' }, { 'e', 'f' } };
```



- A. Option A
- B. Option B
- C. Option C
- D. Option D

**22.** What happens when calling the following method with a non-null and non-empty array?

```
public static void addStationName(String[] names) {
    names[names.length] = "Times Square";
}
```

- A. It adds an element to the array the value of which is Times Square.
  - B. It replaces the last element in the array with the value Times Square.
  - C. It does not compile.
  - D. It throws an exception.
23. How many lines does the following code output?
- ```
String[] days = new String[] { "Sunday", "Monday", "Tuesday",
    "Wednesday", "Thursday", "Friday", "Saturday" };
for (int i = 0; i < days.size(); i++)
    System.out.println(days[i]);
```
- A. Six
  - B. Seven
  - C. The code does not compile.
  - D. The code compiles but throws an exception at runtime.
24. How many dimensions does the array reference moreBools allow?
- ```
boolean[][][] bools, moreBools;
```
- A. One dimension
  - B. Two dimensions
  - C. Three dimensions
  - D. None of the above
25. What is a possible output of the following code?
- ```
String[] strings = new String[2];
System.out.println(strings);
```
- A. [null, null]
  - B. [, ]
  - C. [Ljava.lang.String;@74a14482
  - D. None of the above
26. Which is the first line to prevent this code from compiling and running without error?

```
char[][] ticTacToe = new char[3][3];           // r1
ticTacToe[1][3] = 'X';                         // r2
ticTacToe[2][2] = 'X';
ticTacToe[3][1] = 'X';
System.out.println(ticTacToe.length + " in a row!"); // r3
```

- A. Line r1
  - B. Line r2
  - C. Line r3
  - D. None of the above
27. What is the result of running the following as java Copier?

```
package duplicate;
public class Copier {
    public static void main(String... original) {
        String... copy = original;
        System.out.println(copy.length + " " + copy[0]);
    }
}
```

- A. 0
  - B. 0 followed by an exception
  - C. 1 followed by an exception
  - D. The code does not compile.
28. What is the result of running the following program?

```
1: package fun;
2: public class Sudoku {
3:     static int[][] game = new int[6][6];
4:
5:     public static void main(String[] args) {
6:         game[3][3] = 6;
7:         Object[] obj = game;
8:         obj[3] = "X";
9:         System.out.println(game[3][3]);
10:    }
11: }
```

- A. X
  - B. The code does not compile.
  - C. The code compiles but throws a NullPointerException at runtime.
  - D. The code compiles but throws a different exception at runtime.
29. What does the following output?

```
String[] os = new String[] { "Mac", "Linux", "Windows" };
Arrays.sort(os);
System.out.println(Arrays.binarySearch(os, "RedHat"));
```

- A. -1
- B. -2
- C. -3
- D. The output is not defined.

30. What is the output of the following when run as java FirstName Wolfie?

```
public class FirstName {  
    public static void main(String... names) {  
        System.out.println(names[0]);  
    }  
}
```

- A. FirstName
- B. Wolfie
- C. The code throws an `ArrayIndexOutOfBoundsException`.
- D. The code throws a `NullPointerException`.

31. What is the output of the following when run as java Count 1 2?

```
public class Count {  
    public static void main(String target[]) {  
        System.out.println(target.length);  
    }  
}
```

- A. 0
- B. 1
- C. 2
- D. The code does not compile.

32. What is the output of the following when run as java unix.EchoFirst seed flower?

```
package unix;  
import java.util.*;  
public class EchoFirst {  
  
    public static void main(String[] args) {  
        String one = args[0];  
        Arrays.sort(args);  
        int result = Arrays.binarySearch(args, one);  
    }  
}
```

```

        System.out.println(result);
    }
}

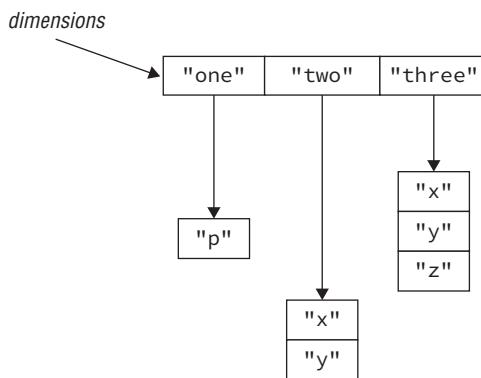
```

- A.** 0  
**B.** 1  
**C.** The code does not compile.  
**D.** The code compiles but throws an exception at runtime.

**33.** Which of these four array declarations produces a different array than the others?

- A.** int[][] nums = new int[2][1];  
**B.** int[] nums[] = new int[2][1];  
**C.** int[] nums[] = new int[][] { { 0 }, { 0 } };  
**D.** int[] nums[] = new int[][] { { 0, 0 } };;

**34.** How do you access the array element with the value of "z"?



- A.** dimensions["three"][2]  
**B.** dimensions["three"][3]  
**C.** dimensions[2][2]  
**D.** dimensions[3][3]

**35.** How many lines does the following code output?

```

String[] days = new String[] { "Sunday", "Monday", "Tuesday",
    "Wednesday", "Thursday", "Friday", "Saturday" };
for (int i = 1; i <= days.length; i++)
    System.out.println(days[i]);

```

- A.** Six  
**B.** Seven  
**C.** The code does not compile.  
**D.** The code compiles but throws an exception at runtime.

36. What is the output of the following when run as java FirstName Wolfie?

```
public class FirstName {  
    public static void main(String... names) {  
        System.out.println(names[1]);  
    }  
}
```

- A. FirstName
- B. Wolfie
- C. The code throws an `ArrayIndexOutOfBoundsException`.
- D. The code throws a `NullPointerException`.

37. Which is the first line to prevent this code from compiling and running without error?

```
char[][] ticTacToe = new char[3][3]; // r1  
ticTacToe[0][0] = 'X'; // r2  
ticTacToe[1][1] = 'X';  
ticTacToe[2][2] = 'X';  
System.out.println(ticTacToe.length + " in a row!"); // r3
```

- A. Line r1
- B. Line r2
- C. Line r3
- D. None of the above

38. What is the output of the following when run as java Count 1 2?

```
public class Count {  
    public static void main(String target[]) {  
        System.out.println(target.length());  
    }  
}
```

- A. 0
- B. 1
- C. 2
- D. The code does not compile.

39. How many dimensions does the array reference `moreBools` allow?

```
boolean[][] bools[], moreBools;
```

- A. One dimension
- B. Two dimensions
- C. Three dimensions
- D. None of the above

40. What is the result of the following when called as `java counting.Binary`?

```
package counting;
import java.util.*;
public class Binary {

    public static void main(String... args) {
        Arrays.sort(args);
        System.out.println(Arrays.toString(args));
    }
}
```

- A. null
  - B. []
  - C. The code does not compile.
  - D. The code compiles but throws an exception at runtime.
41. What does the following output?

```
String[] os = new String[] { "Mac", "Linux", "Windows" };
System.out.println(Arrays.binarySearch(os, "Linux"));
```

- A. 0
  - B. 1
  - C. 2
  - D. The output is not defined.
42. What is the result of running the following program?

```
1: package fun;
2: public class Sudoku {
3:     static int[][] game;
4:
5:     public static void main(String[] args) {
6:         game[3][3] = 6;
7:         Object[] obj = game;
8:         game[3][3] = "X";
9:         System.out.println(game[3][3]);
10:    }
11: }
```

- A. X
- B. The code does not compile.
- C. The code compiles but throws a `NullPointerException` at runtime.
- D. The code compiles but throws a different exception at runtime.

43. What is the output of the following?

```
String[][][] listing = new String[][][] { { "Book" }, { "Game", "29.99" } };
System.out.println(listing.length + " " + listing[0].length);
```

- A. 2 1
- B. 2 2
- C. The code does not compile.
- D. The code compiles but throws an exception at runtime.

44. What is the output of the following when run as java FirstName?

```
public class FirstName {
    public static void main(String[] names) {
        System.out.println(names[0]);
    }
}
```

- A. FirstName
- B. The code does not compile.
- C. The code throws an ArrayIndexOutOfBoundsException.
- D. The code throws a NullPointerException.

45. How many lines does the following code output?

```
String[] days = new String[] { "Sunday", "Monday", "Tuesday",
    "Wednesday", "Thursday", "Friday", "Saturday" };
for (int i = 1; i < days.length; i++)
    System.out.println(days[i]);
```

- A. Six
- B. Seven
- C. The code does not compile.
- D. The code compiles but throws an exception at runtime.

46. What is the output of the following when run as java Count "1 2"?

```
public class Count {
    public static void main(String target[]) {
        System.out.println(target.length);
    }
}
```

- A. 0
- B. 1
- C. 2
- D. The code does not compile.

**47.** What does the following output?

```
String[] os = new String[] { "Linux", "Mac", "Windows" };  
System.out.println(Arrays.binarySearch(os, "Linux"));
```

- A.** 0
- B.** 1
- C.** 2
- D.** The output is not defined.

**48.** Which of the following statements are true?

- I.** You can always change a method signature from `call(String[] arg)` to `call(String... arg)` without causing a compiler error in the calling code.
  - II.** You can always change a method signature from `call(String... arg)` to `call(String[] arg)` without causing a compiler error in the existing code.
- A.** I
  - B.** II
  - C.** Both I and II
  - D.** Neither I nor II

**49.** Which of these four array references can point to an array that is different from the others?

- A.** `int[][][][], nums1a, nums1b;`
- B.** `int[][][], nums2a[], nums2b;`
- C.** `int[][], nums3a[][][], nums3b[][][];`
- D.** `int[] nums4a[][][], nums4b[][][];`

**50.** What is the output of the following when run as `java unix.EchoFirst seed flower?`

```
package unix;  
import java.util.*;  
public class EchoFirst {  
  
    public static void main(String[] args) {  
        Arrays.sort(args);  
        String result = Arrays.binarySearch(args, args[0]);  
        System.out.println(result);  
    }  
}
```

- A.** 0
- B.** 1
- C.** The code does not compile.
- D.** The code compiles but throws an exception at runtime.

# Chapter 5



## Using Loop Constructs

---

**THE OCA EXAM TOPICS COVERED  
IN THIS PRACTICE TEST INCLUDE  
THE FOLLOWING:**

✓ **Using Loop Constructs**

- Create and use while loops
- Create and use for loops including the enhanced for loop
- Create and use do/while loops
- Compare loop constructs
- Use break and continue

1. Which type of loop is best known for its boolean condition that controls entry to the loop?
  - A. do-while loop
  - B. for (traditional)
  - C. for-each
  - D. while
2. Which type of loop is best known for using an index or counter?
  - A. do-while loop
  - B. for (traditional)
  - C. for-each
  - D. while
3. Which type of loop is guaranteed to have the body execute at least once?
  - A. do-while loop
  - B. for (traditional)
  - C. for-each
  - D. while
4. Which of the following can loop through an array without referring to the elements by index?
  - A. do-while loop
  - B. for (traditional)
  - C. for-each
  - D. while
5. What keyword is used to end the current loop iteration and proceed execution with the next iteration of that loop?
  - A. break
  - B. continue
  - C. end
  - D. skip
6. What keyword is used to proceed with execution immediately after a loop?
  - A. break
  - B. continue
  - C. end
  - D. skip
7. Which type of loop has three segments within parentheses?
  - A. do-while loop
  - B. for (traditional)
  - C. for-each
  - D. while

- 8.** Which of the following statements is/are true?
- I. A traditional for loop can iterate through an array starting from index 0.
  - II. A traditional for loop can iterate through an array starting from the end.
  - A. Only I
  - B. Only II
  - C. Both statements
  - D. Neither statement
- 9.** Which of the following statements is/are true?
- I. A for-each loop can iterate through an array starting from index 0.
  - II. A for-each loop can iterate through an array starting from the end.
  - A. Only I
  - B. Only II
  - C. Both statements
  - D. Neither statement
- 10.** Which type of loop has a boolean condition that is first checked after a single iteration through the loop?
- A. do-while loop
  - B. for (traditional)
  - C. for-each
  - D. while
- 11.** What does the following code output?
- ```
int singer = 0;
while (singer)
    System.out.println(singer++);
```
- A. 0
  - B. The code does not compile.
  - C. The loops complete with no output.
  - D. This is an infinite loop.
- 12.** What does the following code output?
- ```
List<String> drinks = Arrays.asList("can", "cup");
for (int container = drinks.size() - 1; container >= 0; container--)
    System.out.print(drinks.get(container) + ",");
```
- A. can,cup,
  - B. cup,can,
  - C. The code does not compile.
  - D. None of the above

**13.** What does the following code output?

```
public static void main(String[] args) {  
    List<String> bottles = Arrays.asList("glass", "plastic");  
    for (int type = 0; type < bottles.size(); ) {  
        System.out.print(bottles.get(type) + ",");  
        break;  
    }  
    System.out.print("end");  
}
```

- A.** glass,end
- B.** glass,plastic,end
- C.** The code does not compile.
- D.** None of the above

**14.** What does the following code output?

```
String letters = "";  
while (letters.length() != 2)  
    letters+="a";  
System.out.println(letters);
```

- A.** aa
- B.** aaa
- C.** The loops complete with no output.
- D.** This is an infinite loop.

**15.** What is the result of the following when run with  
java peregrine.TimeLoop September 3 1940?

```
package peregrine;  
public class TimeLoop {  
    public static void main(String[] args) {  
        for (int i = args.length; i>=0; i++)  
            System.out.println("args");  
    }  
}
```

- A.** args
- B.** argsargs
- C.** The code does not compile.
- D.** None of the above

16. What is the output of the following code?

```
package chicago;
public class Loop {
    private static int count;
    private static String[] stops = new String[] { "Washington",
        "Monroe", "Jackson", "LaSalle" };
    public static void main(String[] args) {
        while (count < stops.length) {
            if (stops[count++].length() < 8) {
                break;
            }
        }
        System.out.println(count);
    }
}
```

- A. 1
- B. 2
- C. 4
- D. The code does not compile.

17. What is the result of the following code?

```
do {
    int count = 0;
    do {
        count++;
    } while (count < 2);
    break;
} while (true);
System.out.println(count);
```

- A. 2
- B. 3
- C. The code does not compile.
- D. This is an infinite loop.

18. Which of the following segments of a `for` loop can be left blank?

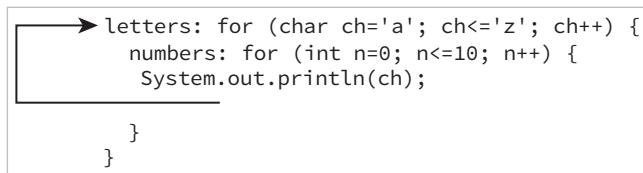
```
for (segmentA; segmentB; segmentC) {
}
```

- A. segmentA
  - B. segmentB
  - C. segmentC
  - D. All of the above
19. How many of the loop types (while, do while, traditional for, and enhanced for) allow you to write code that creates an infinite loop?
- A. One
  - B. Two
  - C. Three
  - D. Four
20. What is the output of the following?
- ```
List<String> drinks = Arrays.asList("can", "cup");
for (int container = 0; container < drinks.size(); container++)
    System.out.print(drinks.get(container) + ",");

A. can,cup,
B. cup,can,
C. The code does not compile.
D. None of the above
```
21. What happens when running the following code?
- ```
do (
    System.out.println("helium");
) while (false);
```
- A. It completes successfully without output.
  - B. It outputs helium once.
  - C. It keeps outputting helium.
  - D. The code does not compile.
22. Which of the following is equivalent to this code snippet given an array of String objects?
- ```
for (int i=0; i<fun.length; i++)
    System.out.println(fun[i]);
```
- A. for (String f = fun) System.out.println(f);
  - B. for (String f : fun) System.out.println(f);
  - C. for (String = fun) System.out.println(it);
  - D. None of the above

23. How many of these statements can be inserted after the `println` to have the code flow follow the arrow in this diagram?

```
break;  
break letters;  
break numbers;
```



- A. None
  - B. One
  - C. Two
  - D. Three
24. Using the diagram in the previous question, how many of these statements can be inserted after the `println` to have the code flow follow the arrow in the diagram?

```
continue;  
continue letters;  
continue numbers;
```

- A. None
- B. One
- C. Two
- D. Three

25. What does the following code output?

```
int singer = 0;  
while (singer > 0)  
    System.out.println(singer++);
```

- A. 0
- B. The code does not compile.
- C. The loops completes with no output.
- D. This is an infinite loop.

26. Which of the following types is `taxis` not allowed to be in order for this code to compile?

```
for (Object obj : taxis) {  
}
```

- A. `ArrayList<Integer>`
- B. `int[]`
- C. `StringBuilder`
- D. All of these are allowed.
27. What is the output of the following?
- ```
boolean balloonInflated = false;
do {
    if (!balloonInflated) {
        balloonInflated = true;
        System.out.print("inflate-");
    }
} while (! balloonInflated);
System.out.println("done");
```
- A. done
- B. inflate-done
- C. The code does not compile.
- D. This is an infinite loop.
28. What does the following code output?
- ```
String letters = "";
while (letters.length() != 3)
    letters+="ab";
System.out.println(letters);
```
- A. ab
- B. abab
- C. The loop completes with no output.
- D. This is an infinite loop.
29. What describes the order in which the three expressions appear in a `for` loop?
- A. boolean conditional, initialization expression, update statement
- B. initialization expression, boolean conditional, update statement
- C. initialization expression, update statement, boolean conditional
- D. None of the above
30. What is the result of the following?
- ```
int count = 10;
List<Character> chars = new ArrayList<>();
do {
```

```
chars.add('a');
for (Character x : chars) count -=1;
} while (count > 0);
System.out.println(chars.size());
```

- A. 3
  - B. 4
  - C. The code does not compile.
  - D. None of the above
31. What is the result of the following?
- ```
int k = 0;
for (int i = 10; i > 0; i--) {
    while (i > 3) i -= 3;
    k += 1;
}
System.out.println(k);
```
- A. 1
  - B. 2
  - C. 3
  - D. 4
32. Which of the following is equivalent to this code snippet given an array of String objects?
- ```
for (int i=fun.length-1; i>=0; i--)
    System.out.println(fun[i]);
```
- A. for (String f = fun) System.out.println(f);
  - B. for (String f : fun) System.out.println(f);
  - C. for (String f fun) System.out.println(it);
  - D. None of the above
33. What does the following code output?

```
public static void main(String[] args) {
    List<String> bottles = Arrays.asList("glass", "plastic");
    for (int type = 0; type < bottles.size();)
        System.out.print(bottles.get(type) + ",");
        break;
    System.out.print("end");
}
```

- A. glass,end
- B. glass,plastic,end
- C. The code does not compile.
- D. None of the above

34. What is the result of the following?

```
String[] nycTourLoops = new String[] { "Downtown", "Uptown", "Brooklyn" };
String[] times = new String[] { "Day", "Night" };
for (int i = 0, j = 0; i < nycTourLoops.length
    && j < times.length; i++; j++)
{
    System.out.print(nycTourLoops[i] + " " + times[j] + "-");
}
```

- A. Downtown Day-
- B. Downtown Day-Uptown Night-
- C. The code does not compile.
- D. The code compiles but throws an exception at runtime.

35. What is the result of the following when run with  
java peregrine.TimeLoop September 3 1940?

```
package peregrine;
public class TimeLoop {
    public static void main(String[] args) {
        for (int i = args.length; i>=0; i--)
            System.out.println(args[i]);
    }
}
```

- A. September
- B. 1940
- C. The code does not compile.
- D. None of the above

36. What is the output of the following?

```
public class Shoelaces {
    public static void main(String[] args) {
        String tie = null;
        while (tie == null)
            tie = "shoelace";
        System.out.print(tie);
    }
}
```

- A. null
- B. shoelace
- C. shoelaceshoelace
- D. None of the above

37. The following code outputs a single letter x. What happens if you remove lines 25 and 28?

```
23: String race = "";
24: loop:
25: do {
26:     race += "x";
27:     break loop;
28: } while (true);
29: System.out.println(race);
```

- A. It prints an empty string.
- B. It still outputs a single letter x.
- C. It no longer compiles.
- D. It becomes an infinite loop.

38. What is the output of the following code?

```
package chicago;
public class Loop {
    private static int count;
    private static String[] stops = new String[] { "Washington",
        "Monroe", "Jackson", "LaSalle" };
    public static void main(String[] args) {
        while (count < stops.length) {
            if (stops[count++].length() < 8) {
                continue;
            }
        }
        System.out.println(count);
    }
}
A. 1
B. 2
C. 4
D. The code does not compile.
```

**39.** What is the output of the following?

```
StringBuilder builder = new StringBuilder();
String str = new String("Leaves growing");
do {
    System.out.println(str);
} while (builder);
System.out.println(builder);
```

- A.** Leaves growing
- B.** This is an infinite loop.
- C.** The code does not compile.
- D.** The code compiles but throws an exception at runtime.

**40.** What is the result of the following code?

```
6: int count = 0;
7: do {
8:     do {
9:         count++;
10:    } while (count < 2);
11:    break;
12: } while (true);
13: System.out.println(count);
```

- A.** 2
- B.** 3
- C.** The code does not compile.
- D.** This is an infinite loop.

**41.** Fill in the blank so this code compiles and does not cause an infinite loop.

```
t: while (true) {
    f: while(true) {
        _____
    }
}
```

- A.** break;
- B.** break f;
- C.** break t;
- D.** None of the above

**42.** What is the result of the following?

```
String[] nycTourLoops = new String[] { "Downtown", "Uptown", "Brooklyn" };
String[] times = new String[] { "Day", "Night" };
for (int i = 0, j = 0; i < nycTourLoops.length
    && j < times.length; i++, j++)
{
    System.out.print(nycTourLoops[i] + " " + times[j] + "-");
}
```

- A.** Downtown Day-
- B.** Downtown Day-Uptown Night-
- C.** The code does not compile.
- D.** The code compiles but throws an exception at runtime.

**43.** How many lines does the following code output?

```
import java.util.*;
public class Exams {
    public static void main(String[] args) {
        List<String> exams = Arrays.asList("OCA", "OCP");
        for (String e1 : exams)
            for (String e2 : exams)
                System.out.println(e1 + " " + e2);
    }
}
```

- A.** One
- B.** Four
- C.** The code does not compile.
- D.** The code compiles but throws an exception at runtime.

**44.** Which of the following best describes the flow of execution in this `for` loop if beta always returns false?

```
for (alpha; beta; gamma) {
    delta;
}
```

- A.** alpha
- B.** alpha, beta
- C.** alpha, beta, gamma
- D.** None of the above

45. Which of the following best describes the flow of execution in this `for` loop if the loop body is run exactly once?

```
for (alpha; beta; gamma) {  
    delta;  
}
```

- A. alpha, delta, gamma, beta
- B. alpha, beta, delta, gamma, beta
- C. alpha, delta, gamma, alpha, beta
- D. alpha, beta, delta, gamma, alpha, beta

46. Which of the following iterates a different number of times than the others?

- A. `for (int k=0; k < 5; k++) {}`
- B. `for (int k=1; k <= 5; k++) {}`
- C. `int k=0; do { } while(k++ < 5)`
- D. `int k=0; while (k++ < 5) {}`

47. What is the output of the following?

```
public class Shoelaces {  
    public static void main(String[] args) {  
        String tie = null;  
        while (tie == null);  
        tie = "shoelace";  
        System.out.print(tie);  
    }  
}
```

- A. null
- B. shoelace
- C. shoelaceshoelace
- D. None of the above

48. What is the output of the following?

```
12: int result = 8;  
13: for: while (result > 7) {  
14:     result++;  
15:     do {  
16:         result--;  
17:     } while (result > 5);  
18:     break for;  
19: }  
20: System.out.println(result);
```

- A. 5
  - B. 8
  - C. The code does not compile.
  - D. The code compiles but throws an exception at runtime.
49. What is the output of the following?
- ```
boolean balloonInflated = false;
do {
    if (!balloonInflated) {
        balloonInflated = true;
        System.out.print("inflate-");
    }
} while (balloonInflated);
System.out.println("done");
```
- A. done
  - B. inflate-done
  - C. The code does not compile.
  - D. This is an infinite loop.

50. Which of the following can fill in the blank to have the code compile successfully?

```
package nyc;
public class TouristBus {
    public static void main(String... args) {
        String[] nycTourLoops = new String[] { "Downtown", "Uptown", "Brooklyn" };
        String[] times = new String[] { "Day", "Night" };
        for (_____ i < 1; i++, j++)
            System.out.println(nycTourLoops[i] + " " + times[j]);
    }
}
```

- A. int i=0; j=0;
- B. int i=0, j=0;
- C. int i=0; int j=0;
- D. int i=0, int j=0;



# Chapter 6



# Working with Methods and Encapsulation

---

**THE OCA EXAM TOPICS COVERED IN  
THIS PRACTICE TEST INCLUDE THE  
FOLLOWING:**

✓ **Working with Methods and Encapsulation**

- Create methods with arguments and return values; including overloaded methods
- Apply the static keyword to methods and fields
- Create and overload constructors; differentiate between default and user defined constructors
- Apply access modifiers
- Apply encapsulation principles to a class
- Determine the effect upon object references and primitive values when they are passed into methods that change the values

1. Fill in the blanks: The \_\_\_\_\_ access modifier allows access to everything the \_\_\_\_\_ access modifier does and more.
  - A. package-private, protected
  - B. protected, public
  - C. protected, package-private
  - D. private, package-private
2. What is the command to call one constructor from another constructor in the same class?
  - A. super()
  - B. this()
  - C. that()
  - D. construct()
3. What is the output of the following application?

```
package stocks;
public class Bond {
    private static int price = 5;
    public boolean sell() {
        if(price<10) {
            price++;
            return true;
        } else if(price>=10) {
            return false;
        }
    }
    public static void main(String[] cash) {
        new Bond().sell();
        new Bond().sell();
        new Bond().sell();
        System.out.print(price);
    }
}
```

- A. 5
- B. 6
- C. 8
- D. The code does not compile.

4. What is true about the following program?

```
package figures;
public class Dolls {
    public void nested() { nested(2,true); } // g1
    public int nested(int level, boolean height) { return nested(level); }
    public int nested(int level) { return level+1; }; // g2

    public static void main(String[] outOfTheBox) {
        System.out.print(new Dolls().nested());
    }
}
```

- A. It compiles successfully and prints 3 at runtime.
  - B. It does not compile because of line g1.
  - C. It does not compile because of line g2.
  - D. It does not compile for some other reason.
5. Fill in the blank: Java uses \_\_\_\_\_ to send data into a method.
- A. pass-by-null
  - B. pass-by-value
  - C. both pass-by-value and pass-by-reference
  - D. pass-by-reference
6. Which of the following is a valid JavaBean method signature?
- A. public void getArrow()
  - B. public void setBow()
  - C. public void setRange(int range)
  - D. public String addTarget(String target)
7. Which of the following statements about calling `this()` in a constructor is not true?
- A. If `this()` is used, it must be the first line of the constructor.
  - B. If `super()` and `this()` are both used in the same constructor, `super()` must appear on the line immediately after `this()`.
  - C. If arguments are provided to `this()`, then there must be a constructor in the class able to take those arguments.
  - D. If the no-argument `this()` is called, then the class must explicitly implement the no-argument constructor.

8. Which of the following can fill in the blank to make the class compile?

```
package ai;
public class Robot {
    _____ compute() { return 10; }
}
```

- A. Public int
- B. Long
- C. void
- D. private String

9. Fill in the blank: A \_\_\_\_\_ variable is always available to all instances of the class.

- A. public
- B. local
- C. static
- D. instance

10. Which line of code, inserted at line p1, causes the application to print 5?

```
package games;
public class Jump {
    private int rope = 1;
    protected boolean outside;
    public Jump() {
        // p1
        outside = true;
    }
    public Jump(int rope) {
        this.rope = outside ? rope : rope+1;
    }
    public static void main(String[] bounce) {
        System.out.print(new Jump().rope);
    }
}
```

- A. this(4);
- B. new Jump(4);
- C. this(5);
- D. rope = 4;

11. Which of the following statements is not true?
- A. An instance of one class may access an instance of another class's attributes if it has a reference to the instance and the attributes are declared `public`.
  - B. An instance of one class may access package-private attributes in a parent class, provided the parent class is not in the same package.
  - C. Two instances of the same class may access each other's `private` attributes.
  - D. An instance of one class may access an instance of another class's attributes if both classes are located in the same package and marked `protected`.
12. Given the following class, what should be inserted into the two blanks to ensure the class data is properly encapsulated?

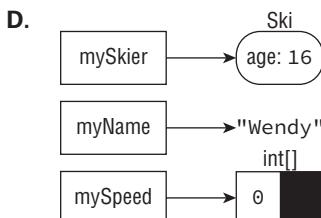
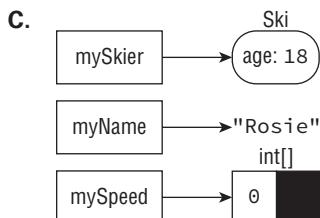
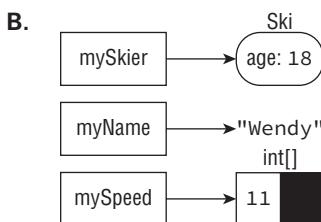
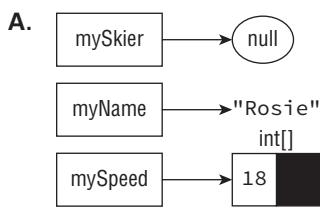
```
package storage;
public class Box {
    public String stuff;
    _____ String _____() {
        return stuff;
    }

    public void setStuff(String stuff) {
        this.stuff = stuff;
    }
}
```

- A. `public` and `getStuff`
  - B. `private` and `isStuff`
  - C. `public` and `setStuff`
  - D. None of the above
13. Which statement about a no-argument constructor is true?
- A. The Java compiler will always insert a default no-argument constructor if you do not define a no-argument constructor in your class.
  - B. In order for a class to call `super()` in one of its constructors, its parent class must explicitly implement a no-argument constructor.
  - C. If a class extends another class that has only one constructor that takes a value, then the child class must explicitly declare at least one constructor.
  - D. A class may contain more than one no-argument constructor.
14. Which of the following method signatures does not contain a compiler error?
- A. `public void sing(String key, String... harmonies)`
  - B. `public void sing(int note, String... sound, int music)`
  - C. `public void sing(String... keys, String... pitches)`
  - D. `public void sing(String... notes, String melodies)`

15. Given the following application, which diagram best represents the state of the mySkier, mySpeed, and myName variables in the main() method after the call to the slalom() method?

```
package slopes;
public class Ski {
    private int age = 18;
    private static void slalom(Ski racer, int[] speed, String name) {
        racer.age = 18;
        name = "Wendy";
        speed = new int[1];
        speed[0] = 11;
        racer = null;
    }
    public static void main(String... mountain) {
        final Ski mySkier = new Ski();
        mySkier.age = 16;
        final int[] mySpeed = new int[1];
        final String myName = "Rosie";
        slalom(mySkier,mySpeed,myName);
    }
}
```



16. Given the class below, which method signature could be successfully added to the class as an overloaded version of the `findAverage()` method?

```
public class Calculations {  
    public Integer findAverage(int sum) { return sum; }  
}
```

- A. `public Long findAverage(int sum)`
  - B. `public Long findAverage(int sum, int divisor)`
  - C. `public Integer average(int sum)`
  - D. `private void findAverage(int sum)`
17. Which of the following is not a reason to use encapsulation when designing a class?
- A. Promote usability by other developers.
  - B. Maintain class data integrity of data elements.
  - C. Prevent users from modifying the internal attributes of a class.
  - D. Increase concurrency and improve performance.
18. Which of the following data types can be modified after they are passed to a method as an argument?
- A. `int[]`
  - B. `String`
  - C. `long`
  - D. `boolean`
19. What is the best way to call the following method from another class in the same package, assuming the class using the method does not have any `static` imports?

```
package useful;  
public class MathHelper {  
    public static int roundValue(double d) {  
        // Implementation omitted  
    }  
}
```

- A. `MathHelper:roundValue(5.92)`
- B. `MathHelper.roundValue(3.1)`
- C. `roundValue(4.1)`
- D. `useful.MathHelper.roundValue(65.3)`

- 20.** Given a method with one of the following return types, which data type prevents the return statement from being used within the method?
- A. byte
  - B. String
  - C. void
  - D. None of the above
- 21.** How many `final` modifiers would need to be removed for this application to compile?
- ```
package end;
public final class Games {
    public final static int finish(final int score) {
        final int win = 3;
        final int result = score++ < 5 ? 2 : win;
        return result+=win;
    }
    public static void main(final String[] v) {
        System.out.print(finish(Integer.parseInt(v[0])));
    }
}
```
- A. None
  - B. One
  - C. Two
  - D. The code will not compile regardless of the number of `final` modifiers that are removed.
- 22.** Fill in the blanks: \_\_\_\_\_ is used to call a constructor in the parent class, while \_\_\_\_\_ is used to reference a member of the parent class.
- A. `super` and `this()`
  - B. `super` and `super()`
  - C. `super()` and `this`
  - D. `super()` and `super`
- 23.** Given the following method signature, which classes can call it?
- ```
void run(String government)
```
- A. Classes in other packages
  - B. Classes in the same package
  - C. Subclasses in a different package
  - D. All classes

- 24.** Which statement(s) about the following class would help to properly encapsulate the data in the class?

```
package shield;
public class Protect {
    private String material;
    protected int strength;

    public int getStrength() {
        return strength;
    }
    public void setStrength(int strength) {
        this.strength = strength;
    }
}
```

- I. Change the access modifier of `strength` to `private`.
  - II. Add a getter method for `material`.
  - III. Add a setter method for `material`.
- A. I  
B. II and III  
C. I, II, and III  
D. None, the data in the class is already encapsulated.
- 25.** Which of the following is a valid method name in Java?
- A. `Go_$Outside$2()`  
B. `have-Fun()`  
C. `new()`  
D. `9enjoyTheWeather()`
- 26.** Which of the following lines of code can be inserted in the line below that would allow the class to compile?

```
package farm;
public class Coop {
    public final int static getNumberOfChickens() {
        // INSERT CODE HERE
    }
}
```

- A. return 3.0;
  - B. return 5L;
  - C. return 10;
  - D. None of the above
27. Which of the following is a true statement about passing data to a method?
- A. A change made to a primitive value passed to a method is reflected in the calling method.
  - B. A change made to the data within an object passed to a method is reflected in the calling method.
  - C. Reassigning an object reference passed to a method is reflected in the calling method.
  - D. A change made to a boolean value passed to a method is reflected in the calling method.
28. What is a possible output of the following application?

```
package wrap;
public class Gift {
    private final Object contents;
    protected Object getContents() {
        return contents;
    }
    protected void setContents(Object contents) {
        this.contents = contents;
    }
    public void showPresent() {
        System.out.print("Your gift: "+contents);
    }
    public static void main(String[] treats) {
        Gift gift = new Gift();
        gift.setContents(gift);
        gift.showPresent();
    }
}
```

- A. Your gift: wrap.Gift@29ca2745
- B. Your gift: Your gift:
- C. It does not compile.
- D. It compiles but throws an exception at runtime.

- 29.** Which of the following is a valid JavaBean method prefix?
- A. is
  - B. gimme
  - C. request
  - D. put
- 30.** Given the following two classes, each in a different package, which line inserted below allows the second class to compile?

```
package clothes;
public class Store {
    public static String getClothes() { return "dress"; }
}
```

```
package wardrobe;
// INSERT CODE HERE
public class Closet {
    public void borrow() {
        System.out.print("Borrowing clothes: "+getClothes());
    }
}
```

- A. static import clothes.Store.getClothes;
  - B. import clothes.Store.\*;
  - C. import static clothes.Store.getClothes;
  - D. import static clothes.Store;
- 31.** What access modifier is used to mark class members package-private?
- A. private
  - B. default
  - C. protected
  - D. None of the above
- 32.** How many lines of the following program contain compilation errors?

```
package sky;
public class Stars {
    private int inThe = 4;
    public void Stars() {
        super();
    }
}
```

```
public Stars(int inThe) {  
    this.inThe = this.inThe;  
}  
public static void main(String[] endless) {  
    System.out.print(new sky.Stars(2).inThe);  
}  
}
```

- A. None
  - B. One
  - C. Two
  - D. Three
33. Which of the following statements is true?
- A. An instance method is allowed to reference a `static` variable.
  - B. A `static` method is allowed to reference an instance variable.
  - C. A `static` initialization block is allowed to reference an instance variable.
  - D. A `final static` variable may be set in a constructor.
34. Given the following method declaration, which line can be inserted to make the code compile?
- ```
public short calculateDistance(double lat1, double lon1,  
    double lat2, double lon2) {  
    // INSERT CODE HERE  
}
```
- A. `return new Integer(3);`
  - B. `return new Byte((byte)6);`
  - C. `return 5L;`
  - D. `return new Short(4).longValue();`
35. Which of the following statements about overloaded methods are true?
- I. Overloaded methods must have the same name.
  - II. Overloaded methods must have the same return type.
  - III. Overloaded methods must have a different list of parameters.
- A. I
  - B. I and II
  - C. I and III
  - D. I, II, and III

36. How many lines of code would need to be removed for the following class to compile?

```
package work;
public class Week {
    private static final String monday;
    String tuesday;
    final static wednesday = 3;
    final protected int thursday = 4;
}
```

- A. One
  - B. Two
  - C. Three
  - D. The code will not compile regardless of the number of lines removed.
37. What is the output of the following application?

```
package pet;
public class Puppy {
    public static int wag = 5; // q1
    public void Puppy(int wag) { // q2
        this.wag = wag;
    }
    public static void main(String[] tail) {
        System.out.print(new Puppy(2).wag); // q3
    }
}
```

- A. 2
  - B. It does not compile because of line q1.
  - C. It does not compile because of line q2.
  - D. It does not compile because of line q3.
38. Fill in the blanks: The \_\_\_\_\_ access modifier allows access to everything the \_\_\_\_\_ access modifier does and more.
- A. public, private
  - B. private, package-private
  - C. package-private, protected
  - D. private, public

- 39.** What is the output of the following application?

```
package ship;
public class Phone {
    private int size;
    public Phone(int size) {this.size=size;

    public static void sendHome(Phone p, int newSize) {
        p = new Phone(newSize);
        p.size = 4;
    }
    public static final void main(String... params) {
        final Phone phone = new Phone(3);
        sendHome(phone,7);
        System.out.print(phone.size);
    }
}
```

- A.** 3
- B.** 4
- C.** 7
- D.** The code does not compile.

- 40.** Given the following class, which line of code when inserted below would prevent the class from compiling?

```
public class Drink {
    public static void water() {}
    public void get() {
        // INSERT CODE HERE
    }
}
```

- A.** water();
- B.** this.Drink.water();
- C.** this.water();
- D.** Drink.water();

- 41.** Given the following method declaration signature, which of the following is a valid call of this method?

```
public void call(int count, String me, String... data)
```

- A.** call(9,"me",10,"Al")
- B.** call(5)

- C. call(2,"home","sweet")
  - D. call("answering","service")
42. Which statement about a **static** variable is true?
- A. The value of a **static** variable must be set when the variable is declared or in a **static** initialization block.
  - B. It is not possible to read **static final** variables outside the class in which they are defined.
  - C. It is not possible to reference **static** methods using **static** imports.
  - D. A **static** variable is always available in all instances of the class.
43. Which of the following is not a true statement?
- A. The first line of every constructor is a call to the parent constructor via the **super()** command.
  - B. A class does not have to have a constructor explicitly defined.
  - C. A constructor may pass arguments to the parent constructor.
  - D. A **final** instance variable whose value is not set when they are declared or in an initialization block should be set by the constructor.
44. How many **final** modifiers would need to be removed for this application to compile?

```
package park;
public class Tree {
    public final static long numberoftrees;
    public final double height;
    static {}
    { final int initheight = 2;
        height = initheight;
    }
    static {
        numberoftrees = 100;
        height = 4;
    }
}
```

- A. None
- B. One
- C. Two
- D. The code will not compile regardless of the number of **final** modifiers removed.

**45.** What is the output of the following application?

```
package jungle;
public class RainForest extends Forest {
    public RainForest(long treeCount) {
        this.treeCount = treeCount+1;
    }
    public static void main(String[] birds) {
        System.out.print(new RainForest(5).treeCount);
    }
}
class Forest {
    public long treeCount;
    public Forest(long treeCount) {
        this.treeCount = treeCount+2;
    }
}
```

- A.** 5
- B.** 6
- C.** 8
- D.** The code does not compile.

**46.** What is the output of the following application?

```
public class ChooseWisely {
    public ChooseWisely() { super(); }
    public int choose(int choice) { return 5; }
    public int choose(short choice) { return 2; }
    public int choose(long choice) { return 11; }
    public static void main(String[] path) {
        System.out.print(new ChooseWisely().choose((byte)2+1));
    }
}
```

- A.** 5
- B.** 2
- C.** 11
- D.** The code does not compile.

47. What is the output of the following application?

```
package sports;
public class Football {
    public static Long getScore(Long timeRemaining) {
        return 2*timeRemaining; // m1
    }
    public static void main(String[] refs) {
        final int startTime = 4;
        System.out.print(getScore(startTime)); // m2
    }
}
```

- A. 8
  - B. The code does not compile because of line m1.
  - C. The code does not compile because of line m2.
  - D. The code compiles but throws an exception at runtime.
48. Which of the following is a valid method name in Java?
- A. \$sprint()
  - B. \jog13()
  - C. walk#()
  - D. %run()
49. Assume there is a class `Bouncer` with a protected variable. Methods in which class can access this variable?
- A. Only subclasses of `Bouncer`
  - B. Any subclass of `Bouncer` or any class in the same package as `Bouncer`
  - C. Only classes in the same package as `Bouncer`
  - D. Any superclass of `Bouncer`
50. Given the following two classes, each in a different package, which line inserted below allows the second class to compile?

```
package commerce;
public class Bank {
    public void withdrawal(int amountInCents) {}
    public void deposit(int amountInCents) {}
}

package employee;
// INSERT CODE HERE
public class Teller {
    public void processAccount(int depositSlip, int withdrawalSlip) {
```

```
    withdrawal(withdrawalSlip);
    deposit(depositSlip);
}
```

- A. import static commerce.Bank.\*;
- B. static import commerce.Bank.\*;
- C. import static commerce.Bank;
- D. None of the above

# Chapter 7



# Working with Inheritance

---

**THE OCA EXAM TOPICS COVERED  
IN THIS PRACTICE TEST INCLUDE  
THE FOLLOWING:**

✓ **Working with Inheritance**

- Describe inheritance and its benefits
- Develop code that makes use of polymorphism; develop code that overrides methods; differentiate between the type of a reference and the type of an object
- Determine when casting is necessary
- Use super and this to access objects and constructors
- Use abstract classes and interfaces

1. How many lines of the following program contain compilation errors?

```
package theater;
class Cinema {
    private String name;
    public Cinema(String name) {this.name = name;}
}
public class Movie extends Cinema {
    public Movie(String movie) {}
    public static void main(String[] showing) {
        System.out.print(new Movie("Another Trilogy").name);
    }
}
```

- A. None
  - B. One
  - C. Two
  - D. Three
2. Which modifier can be applied to an abstract interface method?
- A. protected
  - B. static
  - C. final
  - D. public
3. What is the output of the following application?

```
package radio;
public class Song {
    public void playMusic() {
        System.out.print("Play!");
    }
    private static int playMusic() {
        System.out.print("Music!");
    }
    public static void main(String[] tracks) {
        new Song().playMusic();
    }
}
```

- A. Play!
- B. Music!
- C. The code does not compile.
- D. The code compiles but the answer cannot be determined until runtime.

4. Which of the following statements about inheritance is true?
- A. Inheritance allows objects to access commonly used attributes and methods.
  - B. Inheritance always leads to simpler code.
  - C. All primitives and objects inherit a set of methods.
  - D. Inheritance allows you to write methods that reference themselves.
5. Given the class declaration below, which value cannot be inserted into the blank line that would allow the code to compile?
- ```
package mammal;
interface Pet {}
public class Canine implements Pet {
    public _____ getDoggy() {
        return this;
    }
}
```
- A. Class
  - B. Pet
  - C. Canine
  - D. Object
6. Imagine you are working with another team to build an application. You are developing code that uses a class that the other team has not finished writing yet. Which element of Java would best facilitate this development, allowing easy integration once the other team's code is complete?
- A. An abstract class
  - B. An interface
  - C. static methods
  - D. An access modifier
7. What is the output of the following application?

```
package vehicles;
class Automobile {
    private final String drive() { return "Driving vehicle"; }
}
class Car extends Automobile {
    protected String drive() { return "Driving car"; }
}
public class ElectricCar extends Car {
    public final String drive() { return "Driving electric car"; }
    public static void main(String[] wheels) {
        final Car car = new ElectricCar();
```

```
        System.out.print(car.drive());
    }
}
```

- A. Driving vehicle
  - B. Driving electric car
  - C. Driving car
  - D. The code does not compile.
8. Which of the following statements about inheritance is correct?
- A. Java does not support multiple inheritance.
  - B. Java allows multiple inheritance using abstract classes.
  - C. Java allows multiple inheritance using non-abstract classes.
  - D. Java allows multiple inheritance using interfaces.
9. How many changes need to be made to the classes below to properly override the `watch()` method?
- ```
package entertainment;
class Television {
    protected final void watch() {}
}
public class LCD extends Television {
    Object watch() {}
}
```
- A. One
  - B. Two
  - C. Three
  - D. None; the code compiles as is.
10. Which of the following statements about overriding a method is incorrect?
- A. The return types must be covariant.
  - B. The access modifier of the method in the child class must be the same or broader than the method in the superclass.
  - C. A checked exception thrown by the method in the parent class must be thrown by the method in the child class.
  - D. A checked exception thrown by a method in the child class must be the same or narrower than the exception thrown by the method in the parent class.
11. What is the output of the following application?

```
package machines;
class Computer {
    protected final int process() { return 5; }
```

```
    }
    public class Laptop extends Computer {
        public final int process() { return 3; }
        public static void main(String[] chips) {
            System.out.print(new Laptop().process());
        }
    }
```

- A.** 5
- B.** 3
- C.** The code does not compile.
- D.** The code compiles but throws an exception at runtime.

- 12.** Given that `FileNotFoundException` is a subclass of `IOException`, what is the output of the following application?

```
package edu;
import java.io.*;
class School {
    public int getNumberOfStudentsPerClassroom(String... students)
        throws IOException {
        return 3;
    }
    public int getNumberOfStudentsPerClassroom() throws IOException {
        return 9;
    }
}
public class HighSchool extends School {
    public int getNumberOfStudentsPerClassroom() throws FileNotFoundException {
        return 2;
    }
    public static void main(String[] students) throws IOException {
        School school = new HighSchool();
        System.out.print(school.getNumberOfStudentsPerClassroom());
    }
}
```

- A.** 2
- B.** 3
- C.** 9
- D.** The code does not compile.

**13.** Which modifier can be applied to an interface method?

- A.** protected
- B.** static
- C.** private
- D.** final

**14.** What is the output of the following application?

```
package track;
interface Run {
    default void walk() {
        System.out.print("Walking and running!");
    }
}
interface Jog {
    default void walk() {
        System.out.print("Walking and jogging!");
    }
}
public class Sprint implements Run, Jog {
    public void walk() {
        System.out.print("Sprinting!");
    }
    public static void main() {
        new Sprint().walk();
    }
}
```

- A.** Walking and running!
- B.** Walking and jogging!
- C.** Sprinting!
- D.** The code does not compile.

**15.** Which of the following statements about interfaces is not true?

- A.** An interface can extend another interface.
- B.** An interface can implement another interface.
- C.** A class can implement two interfaces.
- D.** A class can extend another class.

16. What is the output of the following application?

```
package transport;

class Ship {
    protected int weight = 3;
    private int height = 5;
    public int getWeight() { return weight; }
    public int getHeight() { return height; }
}

public class Rocket extends Ship {
    public int weight = 2;
    public int height = 4;
    public void printDetails() {
        System.out.print(super.getWeight()+"."+super.height);
    }
    public static final void main(String[] fuel) {
        new Rocket().printDetails();
    }
}
A. 2,5
B. 3,4
C. 3,5
D. The code does not compile.
```

17. Fill in the blanks: Excluding `default` and `static` methods, a(n) \_\_\_\_\_ can contain both abstract and concrete methods, while a(n) \_\_\_\_\_ contains only abstract methods.

- A. concrete class, abstract class
- B. concrete class, interface
- C. interface, abstract class
- D. abstract class, interface

18. Which statement about the following class is correct?

```
package shapes;
abstract class Triangle {
    abstract String getDescription();
}
```

```
class RightTriangle extends Triangle {  
    protected String getDescription() { return "rt"; } // g1  
}  
public abstract class IsoscelesRightTriangle extends RightTriangle { // g2  
    public String getDescription() { return "irt"; }  
    public static void main(String[] edges) {  
        final Triangle shape = new IsoscelesRightTriangle(); // g3  
        System.out.print(shape.getDescription());  
    }  
}
```

- A. The code does not compile due to line g1.  
B. The code does not compile due to line g2.  
C. The code does not compile due to line g3.  
D. The code compiles and runs without issue.
19. Given that `Short` and `Integer` extend `Number`, what type can be used to fill in the blank in the class below to allow it to compile?

```
package band;  
  
interface Horn { public Integer play(); }  
abstract class Woodwind { public Short play() {return 3;} }  
public final class Saxophone extends Woodwind implements Horn {  
    public _____ play() {  
        return null;  
    }  
}
```

- A. `Integer`  
B. `Short`  
C. `Number`  
D. None of the above
20. Fill in the blanks: A class \_\_\_\_\_ an interface, while a class \_\_\_\_\_ an abstract class.
- A. extends, implements  
B. extends, extends  
C. implements, extends  
D. implements, implements

21. What is the output of the following application?

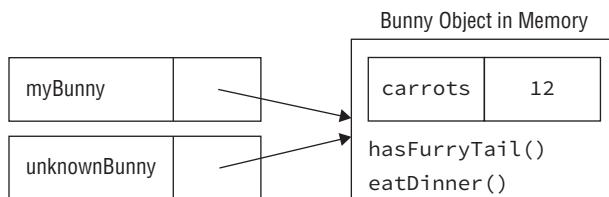
```
package paper;

abstract class Book {
    protected static String material = "papyrus";
    public Book() {}
    public Book(String material) {this.material = material;}
}

public class Encyclopedia extends Book {
    public static String material = "cellulose";
    public Encyclopedia() {super();}
    public String getMaterial() {return super.material;}
    public static void main(String[] pages) {
        System.out.print(new Encyclopedia().getMaterial());
    }
}
```

- A. papyrus
- B. cellulose
- C. The code does not compile.
- D. The code compiles but throws an exception at runtime.

22. The following diagram shows two reference variables pointing to the same Bunny object in memory. The reference variable myBunny is of type Bunny, while unknownBunny is of an unknown data type. Which statement about the reference variables is not true? For this question, assume the instance methods and variables shown in the diagram are marked public.



- A. If the unknownBunny reference does not have access to the same variables and methods that myBunny has access to, it can be explicitly cast to a reference type that does.
- B. The data type of unknownBunny must be Bunny or a subclass of Bunny.
- C. If the data type of unknownBunny is Bunny, it has access to all of the same methods and variables as myBunny.
- D. The data type of unknownBunny could be an interface, class, or abstract class.

**23.** Which of the following modifiers can be applied to an abstract method?

- A. final
- B. private
- C. default
- D. protected

**24.** What is the output of the following application?

```
package space;

interface Sphere {
    default String getName() { return "Unknown"; }
}
abstract class Planet {
    abstract String getName();
}
public class Mars extends Sphere implements Planet {
    public Mars() {
        super();
    }
    public String getName() { return "Mars"; }
    public static void main(final String[] probe) {
        System.out.print(((Planet)new Mars()).getName());
    }
}
```

- A. Mars
- B. Unknown
- C. The code does not compile due to the declaration of Sphere.
- D. The code does not compile for another reason.

**25.** Which of the following statements is correct?

- A. A reference to a class can be assigned to a subclass reference without an explicit cast.
- B. A reference to a class can be assigned to a superclass reference without an explicit cast.
- C. A reference to an interface can be assigned to a reference of a class that implements the interface without an explicit cast.
- D. A reference to a class that implements an interface can be assigned to an interface reference only with an explicit cast.

**26.** Of the following four modifiers, choose the one that is not implicitly applied to all interface variables.

- A. final
- B. abstract
- C. static
- D. public

**27.** What is the output of the following application?

```
package race;
abstract class Car {
    static { System.out.print("1"); }
    public Car(String name) {
        super();
        System.out.print("2");
    }
    { System.out.print("3"); }
}
public class BlueCar extends Car {
    { System.out.print("4"); }
    public BlueCar() {
        super("blue");
        System.out.print("5");
    }
    public static void main(String[] gears) {
        new BlueCar();
    }
}
```

- A.** 23451
- B.** 12354
- C.** 13245
- D.** The code does not compile.

**28.** Fill in the blank: Overloaded and overridden methods always have \_\_\_\_\_.

- A.** the same parameter list
- B.** different return types
- C.** the same method name
- D.** covariant return types

**29.** What is the output of the following application?

```
package sports;
abstract class Ball {
    protected final int size;
    public Ball(int size) {
        this.size = size;
    }
}
```

```

interface Equipment {}
public class SoccerBall extends Ball implements Equipment {
    public SoccerBall() {
        super(5);
    }
    public Ball get() { return this; }
    public static void main(String[] passes) {
        Equipment equipment = (Equipment)(Ball)new SoccerBall().get();
        System.out.print(((SoccerBall)equipment).size());
    }
}

```

- A.** 5
- B.** The code does not compile due an invalid cast.
- C.** The code does not compile for a different reason.
- D.** The code compiles but throws a `ClassCastException` at runtime.
- 30.** Fill in the blanks: A class that defines an instance variable with the same name as a variable in the parent class is referred to as \_\_\_\_\_ a variable, while a class that defines a `static` method with the same signature as a `static` method in a parent class is referred to as \_\_\_\_\_ a method.
- A.** hiding, overriding
- B.** overriding, hiding
- C.** hiding, hiding
- D.** replacing, overriding
- 31.** Which statement about the following class is correct?

```

package shapes;

abstract class Parallelogram {
    private int getEqualSides() {return 0;}
}
abstract class Rectangle extends Parallelogram {
    public static int getEqualSides() {return 2;} // x1
}
public final class Square extends Rectangle {
    public int getEqualSides() {return 4;} // x2
    public static void main(String[] corners) {
        final Square myFigure = new Square(); // x3
        System.out.print(myFigure.getEqualSides());
    }
}

```

- A. The code does not compile due to line x1.
  - B. The code does not compile due to line x2.
  - C. The code does not compile due to line x3.
  - D. The code compiles and runs without issue.
32. What is the output of the following application?

```
package flying;

class Rotorcraft {
    protected final int height = 5;
    abstract int fly();
}

public class Helicopter extends Rotorcraft {
    private int height = 10;
    protected int fly() {
        return super.height;
    }
    public static void main(String[] unused) {
        Helicopter h = (Helicopter) new Rotorcraft();
        System.out.print(h.fly());
    }
}
```

- A. 5
  - B. 10
  - C. The code does not compile.
  - D. The code compiles but produces a ClassCastException at runtime.
33. Fill in the blanks: A class may be assigned to a(n) \_\_\_\_\_ reference variable automatically but requires an explicit cast when assigned to a(n) \_\_\_\_\_ reference variable.
- A. subclass, outer class
  - B. superclass, subclass
  - C. subclass, superclass
  - D. abstract class, concrete class
34. Fill in the blank: A(n) \_\_\_\_\_ is the first non-abstract subclass that is required to implement all of the inherited abstract methods.
- A. abstract class
  - B. abstraction
  - C. concrete class
  - D. interface

**35.** How many compiler errors does the following code contain?

```
package animal;
interface CanFly {
    public void fly() {}
}
final class Bird {
    public int fly(int speed) {}
}
public class Eagle extends Bird implements CanFly {
    public void fly() {}
}
```

- A.** None
- B.** One
- C.** Two
- D.** Three

**36.** Which of the following is not an attribute common to both abstract classes and interfaces?

- A.** They both can contain static variables.
- B.** They both can contain default methods.
- C.** They both can contain static methods.
- D.** They both can contain abstract methods.

**37.** What is the output of the following application?

```
package musical;
interface SpeakDialogue { default int talk() { return 7; } }
interface SingMonologue { default int talk() { return 5; } }
public class Performance implements SpeakDialogue, SingMonologue {
    public int talk(String... x) {
        return x.length;
    }
    public static void main(String[] notes) {
        System.out.print(new Performance().talk(notes));
    }
}
```

- A.** 7
- B.** 5
- C.** The code does not compile.
- D.** The code compiles without issue, but the output cannot be determined until runtime.

- 38.** Which of the following is a virtual method?
- A. protected instance methods
  - B. static methods
  - C. private instance methods
  - D. final instance methods
- 39.** Fill in the blanks: An interface \_\_\_\_\_ another interface, while a class \_\_\_\_\_ another class.
- A. implements, extends
  - B. extends, extends
  - C. implements, implements
  - D. extends, implements
- 40.** What is the output of the following application?
- ```
class Math {  
    public final double secret = 2;  
}  
class ComplexMath extends Math {  
    public final double secret = 4;  
}  
public class InfiniteMath extends ComplexMath {  
    public final double secret = 8;  
    public static void main(String[] numbers) {  
        Math math = new InfiniteMath();  
        System.out.print(math.secret);  
    }  
}
```
- A. 2
  - B. 4
  - C. 8
  - D. The code does not compile.

- 41.** Given the following method and the fact that `FileNotFoundException` is a subclass of `IOException`, which of the following method signatures is a valid override by a subclass?

```
protected void dance() throws FileNotFoundException {}
```

- A. void dance() throws IOException
- B. public void dance() throws IOException
- C. private void dance() throws FileNotFoundException
- D. public final void dance()

- 42.** Given the class definitions below, which value, when inserted into the blank line, does not allow the class to compile?

```
public class Canine {}
public class Dog extends Canine {}
public class Wolf extends Canine {}
public final class Husky extends Dog {}
public class Zoologist {
    Canine animal;
    public final void setAnimal(Dog animal) { this.animal = animal; }
    public static void main(String[] furryFriends) {
        new Zoologist().setAnimal(______);
    }
}
```

- A.** new Husky()
- B.** new Dog()
- C.** new Wolf()
- D.** null

- 43.** Which of the following modifiers cannot be applied to an interface method?

- A.** final
- B.** default
- C.** static
- D.** abstract

- 44.** Which statement about the following application is true?

```
package party;

abstract class House {
    protected abstract Object getSpace();
}
abstract class Room extends House {
    abstract Object getSpace(Object list);
}
abstract public class Ballroom extends House {
    protected abstract Object getSpace();
    public static void main(String[] squareFootage) {
        System.out.print("Let's start the party!");
    }
}
```

- A. It compiles and at runtime prints Let's start the party!
  - B. It does not compile for one reason.
  - C. It does not compile for two reasons.
  - D. It does not compile for three reasons.
45. Fill in the blanks: \_\_\_\_\_ methods must have a different list of parameters, while \_\_\_\_\_ methods must have the exact same return type.
- A. Overloaded, overridden
  - B. Inherited, overridden
  - C. Overridden, overloaded
  - D. None of the above
46. Which of the following statements about no-argument constructors is correct?
- A. If a parent class does not include a no-argument constructor, a child class cannot declare one.
  - B. If a parent class does not include a no-argument constructor (nor a default one inserted by the compiler), a child class must contain at least one constructor definition.
  - C. If a parent class contains a no-argument constructor, a child class must contain a no-argument constructor.
  - D. If a parent class contains a no-argument constructor, a child class must contain at least one constructor.
47. Fill in the blanks: The \_\_\_\_\_ determines which attributes exist in memory, while the \_\_\_\_\_ determines which attributes are accessible by the caller.
- A. reference type, signature
  - B. object type, superclass
  - C. reference type, object type
  - D. object type, reference type
48. Given that Integer and Long are subclasses of Number, what type can be used to fill in the blank in the class below to allow it to compile?

```
package orchestra;
interface MusicCreator { public Number play(); }
abstract class StringInstrument { public Long play() {return 3L;} }
public class Violin extends StringInstrument implements MusicCreator {
    public _____ play() {
        return 12;
    }
}
```

- A. Long
  - B. Integer
  - C. Long or Integer
  - D. Long or Number
49. Which of the following is the best reason for creating a default interface method?
- A. Allow interface methods to be inherited.
  - B. Add backward compatibility to existing interfaces.
  - C. Give an interface the ability to create concrete methods.
  - D. Allow an interface to define a method at the class level.
50. Given that EOFException is a subclass of IOException, what is the output of the following application?

```
package ai;
import java.io.*;
class Machine {
    public boolean turnOn() throws EOFException {return true;}
}
public class Robot extends Machine {
    public boolean turnOn() throws IOException {return false;}
    public static void main(String[] doesNotCompute) throws Exception {
        Machine m = new Robot();
        System.out.print(m.turnOn());
    }
}


- A. true
- B. false
- C. The code does not compile.
- D. The code compiles but produces an exception at runtime.

```

# Chapter 8



# Handling Exceptions

---

**THE OCA EXAM TOPICS COVERED  
IN THIS PRACTICE TEST INCLUDE  
THE FOLLOWING:**

✓ **Handling Exceptions**

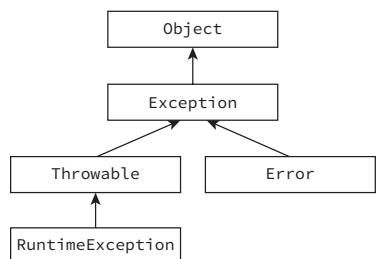
- Differentiate among checked exceptions, unchecked exceptions, and Errors
- Create a try-catch block and determine how exceptions alter normal program flow
- Describe the advantages of Exception handling
- Create and invoke a method that throws an exception
- Recognize common exception classes (such as NullPointerException, ArithmeticException, ArrayIndexOutOfBoundsException, ClassCastException)

1. What is the result of compiling and executing the following application?

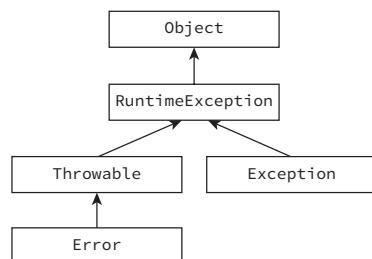
```
package mind;
public class Remember {
    public static void think() throws Exception { // k1
        try {
            throw new Exception();
        }
    }
    public static void main(String... ideas) throws Exception {
        think();
    }
}
```

- A. The code compiles and runs without printing anything.
  - B. The code compiles but a stack trace is printed at runtime.
  - C. The code does not compile because of line k1.
  - D. The code does not compile for another reason.
2. Choose the answer that lists the keywords in the order that they would be used together.
- A. catch, try, finally
  - B. try, catch, finally
  - C. finally, catch, try
  - D. try, finally, catch
3. Which of the following diagrams of java.lang classes shows the inheritance model properly?

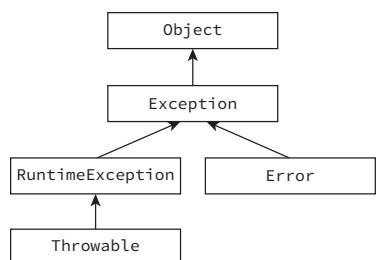
A.



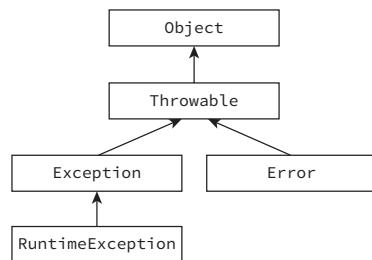
B.



C.



D.



4. Which of the following `Throwable` types is it recommended not to catch in a Java application?

- A. `Error`
- B. `CheckedException`
- C. `Exception`
- D. `RuntimeException`

5. What is the output of the following application?

```
package game;
public class Baseball {
    public static void main(String... teams) {
        try {
            int score = 1;
            System.out.print(score++);
        } catch (Throwable t) {
            System.out.print(score++);
        } finally {
            System.out.print(score++);
        }
        System.out.print(score++);
    }
}
```

- A. 123
- B. 124
- C. 12
- D. None of the above

6. Which of the following is a checked exception?

- A. `ClassCastException`
- B. `IOException`
- C. `ArrayIndexOutOfBoundsException`
- D. `IllegalArgumentException`

7. Fill in the blanks: The \_\_\_\_\_ keyword is used in method declarations, while the \_\_\_\_\_ keyword is used to throw an exception to the surrounding process.

- A. `throws, throw`
- B. `catch, throw`
- C. `throw, throws`
- D. `throws, catch`

8. If a try statement has catch blocks for both Exception and IOException, then which of the following statements is correct?
- A. The catch block for Exception must appear before the catch block for IOException.
  - B. The catch block for IOException must appear before the catch block for Exception.
  - C. The catch blocks for these two exception types can be declared in any order.
  - D. A try statement cannot be declared with these two catch block types because they are incompatible.

9. What is the output of the following application?

```
package game;
public class Football {
    public static void main(String officials[]) {
        try {
            System.out.print('A');
            throw new RuntimeException("Out of bounds!");
        } catch (ArrayIndexOutOfBoundsException aioobe) {
            System.out.print('B');
            throw t;
        } finally {
            System.out.print('C');
        }
    }
}
```

- A. ABC
- B. ABC, followed by a stack trace for a RuntimeException
- C. AC, followed by a stack trace for a RuntimeException
- D. None of the above

10. What is the result of compiling and running the following application?

```
package castles;
public class Fortress {
    public void openDrawbridge() throws Exception { // p1
        try {
            throw new Exception("Circle");
        } catch (Exception e) {
            System.out.print("Opening!");
        } finally {
            System.out.print("Walls"); // p2
        }
    }
}
```

- ```
public static void main(String[] moat) {  
    new Fortress().openDrawbridge(); // p3  
}  
}  
  
A. The code does not compile because of line p1.  
B. The code does not compile because of line p2.  
C. The code does not compile because of line p3.  
D. The code compiles, but a stack trace is printed at runtime.  
  
11. Which of the following exception types must be handled or declared by the method in which they are thrown?  
A. NullPointerException  
B. Exception  
C. RuntimeException  
D. ArithmeticException  
  
12. What is the output of the following application?  
  
package game;  
public class BasketBall {  
    public static void main(String[] dribble) {  
        try {  
            System.out.print(1);  
            throw new ClassCastException();  
        } catch (ArrayIndexOutOfBoundsException ex) {  
            System.out.print(2);  
        } catch (Throwable ex) {  
            System.out.print(3);  
        } finally {  
            System.out.print(4);  
        }  
        System.out.print(5);  
    }  
}
```
- A. 1345  
B. 1235  
C. The code does not compile.  
D. The code compiles but throws an exception at runtime.

13. Which of the following statements about a `finally` block is true?
- A. Every line of the `finally` block is guaranteed to be executed.
  - B. The `finally` block is executed only if the related `catch` block is also executed.
  - C. The `finally` statement requires brackets {}.
  - D. The `finally` block cannot throw an exception.
14. Given that `FileNotFoundException` is a subclass of `IOException`, what is the output of the following application?

```
package office;
import java.io.*;
public class Printer {
    public void print() {
        try {
            throw new FileNotFoundException();
        } catch (IOException exception) {
            System.out.print("Z");
        } catch (FileNotFoundException enfe) {
            System.out.print("X");
        } finally {
            System.out.print("Y");
        }
    }
    public static void main(String... ink) {
        new Printer().print();
    }
}
```

- A. XY
  - B. ZY
  - C. The code does not compile.
  - D. The code compiles but a stack trace is printed at runtime.
15. Which keywords are required with a `try` statement?
- I. `catch`
  - II. `finalize`
  - III. `finally`
- A. I only
  - B. II only
  - C. I or III, or both
  - D. None of these statements are required with a `try` statement.

16. Which statement about the role of exceptions in Java is incorrect?
- A. Exceptions are often used when things “go wrong” or deviate from the expected path.
  - B. An application that throws an exception will terminate.
  - C. Some exceptions can be avoided programmatically.
  - D. An application that can properly handle its exception may recover from unexpected problems.

17. What is the output of the following application?

```
package harbor;
class CapsizedException extends Exception {}
class Transport {
    public int travel() throws CapsizedException { return 2; }
}
public class Boat {
    public int travel() throws Exception { return 4; } // j1
    public static void main(String... distance) throws Exception{
        try {
            System.out.print(new Boat().travel());
        } catch (Exception e) {
            System.out.print(8);
        }
    }
}
```

- A. 4
- B. 8
- C. The code does not compile due to line j1.
- D. The code does not compile for another reason.

18. Which of following method signatures would not be allowed in a class implementing the Printer interface?

```
class PrintException extends Exception {}
class PaperPrintException extends PrintException {}
public interface Printer {
    abstract int printData() throws PrintException;
}


- A. public int printData() throws PaperPrintException
- B. public int printData() throws Exception
- C. public int printData()
- D. None of the above

```

- 19.** Which import statement is required to be declared in order to use the Exception, RuntimeException, and Throwable classes in an application?

- A. import java.exception.\*;
- B. import java.util.exception.\*;
- C. import java.lang.\*;
- D. None of the above

- 20.** Which statement about the following classes is correct?

```
class GasException extends Exception {}  
class Element {  
    public int getSymbol() throws GasException { return -1; } // g1  
}  
public class Oxygen extends Element {  
    public int getSymbol() { return 8; } // g2  
    public void printData() {  
        try {  
            System.out.print(getSymbol());  
        } catch { // g3  
            System.out.print("Unable to read data");  
        }  
    }  
}
```

- A. The code does not compile because of line g1.
- B. The code does not compile because of line g2.
- C. The code does not compile because of line g3.
- D. None of the above

- 21.** Fill in the blanks: A program must handle or declare \_\_\_\_\_ but should never handle \_\_\_\_\_.

- A. java.lang.Error, unchecked exceptions
- B. checked exceptions, java.lang.Error
- C. java.lang.Throwable, java.lang.Error
- D. unchecked exceptions, java.lang.Exception

- 22.** What is the result of compiling and running the following application?

```
package castles;  
class CastleUnderSiegeException extends Exception {}  
class KnightAttackingException extends CastleUnderSiegeException {}  
public class Citadel {  
    public void openDrawbridge() throws RuntimeException { // q1
```

```
try {
    throw new KnightAttackingException();
} catch (Exception e) {
    throw new ClassCastException();
} finally {
    throw new CastleUnderSiegeException(); // q2
}
}

public static void main(String[] moat) {
    new Citadel().openDrawbridge(); // q3
}
```

- A. The code does not compile because of line q1.
  - B. The code does not compile because of line q2.
  - C. The code does not compile because of line q3.
  - D. The code compiles, but a stack trace is printed at runtime.
23. If an exception matches two or more catch blocks, which catch block is executed?
- A. The first one that matches is executed.
  - B. The last one that matches is executed.
  - C. All matched blocks are executed.
  - D. It is not possible to write code like this.
24. What is the output of the following application?

```
package system;
public class Computer {
    public void compute() throws Exception {
        throw new RuntimeException("Error processing request");
    }
    public static void main(String[] bits) {
        try {
            new Computer().compute();
            System.out.print("Ping");
        } catch (NullPointerException e) {
            System.out.print("Pong");
            throw e;
        }
    }
}
```

- A. Ping
- B. Pong
- C. The code does not compile.
- D. The code compiles but throws an exception at runtime.

25. In the following application, the value of `list` has been omitted. Assuming the code compiles without issue, which one of the following is not a possible output of executing this class?

```
package checkboard;

public class Attendance {
    private Boolean[] list = // value omitted
    public int printTodaysCount() {
        int count=0;
        for(int i=0; i<10; i++) {
            if(list[i]) ++count;
        }
        return count;
    }
    public static void main(String[] roster) {
        new Attendance().printTodaysCount();
    }
}
```

- A. A stack trace for `NullPointerException` is printed.
  - B. A stack trace for `ArrayIndexOutOfBoundsException` is printed.
  - C. A stack trace for `ClassCastException` is printed.
  - D. None of the above
26. Fill in the blanks: A \_\_\_\_\_ occurs when a program recurses too deeply into an infinite loop, while a(n) \_\_\_\_\_ occurs when a reference to a nonexistent object is acted upon.
- A. `NoClassDefFoundError`, `StackOverflowError`
  - B. `StackOverflowError`, `NullPointerException`
  - C. `ClassCastException`, `IllegalArgumentException`
  - D. `StackOverflowError`, `IllegalArgumentException`
27. Which of the following is not a reason to add checked exceptions to a method signature?
- A. To force a caller to handle or declare its exceptions
  - B. To notify the caller of potential types of problems
  - C. To ensure that exceptions never cause the application to terminate
  - D. To give the caller a chance to recover from a problem

28. What is the output of the following application?

```
package peculiar;
public class Stranger {
    public static String getFullName(String firstName, String lastName) {
        try {
            return firstName.toString() + " " + lastName.toString();
        } finally {
            System.out.print("Finished!");
        } catch (NullPointerException npe) {
            System.out.print("Problem?");
        }
        return null;
    }
    public static void main(String[] things) {
        System.out.print(getFullName("Joyce", "Hopper"));
    }
}
```

- A. Joyce Hopper
- B. Finished!Joyce Hopper
- C. Problem?Finished!null
- D. None of the above

29. Fill in the blanks: A try statement has \_\_\_\_\_ finally block(s) and \_\_\_\_\_ catch blocks.

- A. zero or one, zero or more
- B. one, one or more
- C. zero or one, zero or one
- D. one or more, zero or one

30. What is the output of the following application?

```
package pond;
abstract class Duck {
    protected int count;
    public abstract int getDuckies();
}
public class Ducklings extends Duck {
    private int age;
    public Ducklings(int age) { this.age = age; }
    public int getDuckies() { return this.age/count; }
    public static void main(String[] pondInfo) {
```

```
        Duck itQuacks = new Ducklings(5);
        System.out.print(itQuacks.getDuckies());
    }
}
```

- A. 0
- B. 5
- C. The code does not compile.
- D. The code compiles but throws an exception at runtime.

31. Given a `try` statement, if both the `catch` block and the `finally` block each throw an exception, what does the caller see?
- A. The exception from the `catch` block
  - B. The exception from the `finally` block
  - C. Both the exception from the `catch` block and the exception from the `finally` block
  - D. None of the above
32. What is the output of the following application?

```
package zoo;
class BigCat {
    void roar(int level) throw RuntimeException { // m1
        if(level<3) throw new IllegalArgumentException("Incomplete");
        System.out.print("Roar!");
    }
}
public class Lion extends BigCat {
    public void roar() { // m2
        System.out.print("Roar!!!");
    }
}

public static void main(String[] cubs) {
    final BigCat kitty = new Lion(); // m3
    kitty.roar(2);
}
```

- A. The code does not compile because of line m1.
- B. The code does not compile because of line m2.
- C. The code does not compile because of line m3.
- D. The code compiles but a stack trace is printed at runtime.

33. Given the following code snippet, which specific exception will be thrown?

```
final Object exception = new Exception();
final Exception data = (RuntimeException)exception;
System.out.print(data);
```

- A. ClassCastException
- B. RuntimeException
- C. NullPointerException
- D. None of the above

34. Which of the following classes will handle all types in a catch block?

- A. Exception
- B. Error
- C. Throwable
- D. RuntimeException

35. In the following application, the values of street and city have been omitted. Which one of the following is a possible output of executing this class?

- I. 350 5th Ave - New York
- II. Posted:350 5th Ave - New York

```
package registration;
public class Address {
    public String getAddress(String street, String city) {
        try {
            return street.toString() + " : " + city.toString();
        } finally {
            System.out.print("Posted:");
        }
    }
    public static void main(String[] form) {
        String street = // value omitted
        String city = // value omitted
        System.out.print(new Address().getAddress(street,city));
    }
}
```

- A. I only
- B. II only
- C. I and II
- D. None of the above

- 36.** If a try statement has catch blocks for both `ClassCastException` and `RuntimeException`, then which of the following statements is correct?
- A.** The catch block for `ClassCastException` must appear before the catch block for `RuntimeException`.
  - B.** The catch block for `RuntimeException` must appear before the catch block for `ClassCastException`.
  - C.** The catch blocks for these two exception types can be declared in any order.
  - D.** A try statement cannot be declared with these two catch block types because they are incompatible.
- 37.** Which of the following is the best scenario to use an exception?
- A.** The computer caught fire.
  - B.** The code does not compile.
  - C.** A caller passes invalid data to a method.
  - D.** A method finishes sooner than expected.
- 38.** What is the output of the following application?

```
package body;
class Organ {
    public void operate() throws RuntimeException {
        throw new RuntimeException("Not supported");
    }
}
public class Heart extends Organ {
    public void operate() throws Exception {
        System.out.print("beat");
    }
    public static void main(String... cholesterol) throws Exception {
        try {
            new Heart().operate();
        } finally {
        }
    }
}
A. beat
B. Not supported
C. The code does not compile.
D. The code compiles but a stack trace is printed at runtime.
```

39. Which statement about the following exception statement is correct?

```
throw new NullPointerException();
```

- A. The code where this is called must include a try-catch block that handles this exception.
- B. The method where this is called must declare a compatible exception.
- C. This exception cannot be handled.
- D. This exception can be handled with a try-catch block or ignored altogether by the surrounding method.

40. What is the output of the following application?

```
package clothing;
public class Coat {
    public Long zipper() throws Exception {
        try {
            String checkZipper = (String) new Object();
        } catch (Exception e) {
            throw RuntimeException("Broken!");
        }
        return null;
    }
    public static void main(String... warmth) {
        try {
            new Coat().zipper();
            System.out.print("Finished!");
        } catch (Throwable t) {}
    }
}
```

- A. Finished!
- B. Finished!, followed by a stack trace
- C. The application does not produce any output at runtime.
- D. The code does not compile.

41. Given the following application, which type of exception will be printed in the stack trace at runtime?

```
package carnival;
public class WhackAnException {
    public static void main(String... hammer) {
        try {
            throw new ClassCastException();
        }
```

```

        } catch (IllegalArgumentException e) {
            throw new IllegalArgumentException();
        } catch (RuntimeException e) {
            throw new NullPointerException();
        } finally {
            throw new RuntimeException();
        }
    }
}

```

- A.** `IllegalArgumentException`
- B.** `NullPointerException`
- C.** `RuntimeException`
- D.** The code does not compile.
- 42.** Which of these method signatures is allowed in a class implementing the `Outfielder` interface?

```

class OutOfBoundsException extends BadCatchException {}
class BadCatchException extends Exception {}

```

```

public interface Outfielder {
    public void catchBall() throws OutOfBoundsException;
}

```

- A.** `public int catchBall() throws OutOfBoundsException`
- B.** `public int catchBall() throws BadCatchException`
- C.** `public int catchBall() throws Exception`
- D.** None of the above

- 43.** What is the output of the following application?

```

package city;
public class Street {
    public static void dancing() throws RuntimeException {
        try {
            throw new IllegalArgumentException();
        } catch (Error) {
            System.out.print("Unable!");
        }
    }
    public static void main(String... count) throws RuntimeException {
        dancing();
    }
}

```

- A. Unable!
  - B. The application does not produce any output.
  - C. The application compiles but produces a stack trace at runtime.
  - D. The code does not compile.
44. What is the result of compiling and running the following application?
- ```
package castles;
class DragonException extends Exception {}
public class Lair {
    public void openDrawbridge() throws Exception { // r1
        try {
            throw new Exception("This Exception");
        } catch (RuntimeException e) {
            throw new DragonException(); // r2
        } finally {
            throw new RuntimeException("Or maybe this one");
        }
    }
    public static void main(String[] moat) throws Exception {
        new Lair().openDrawbridge(); // r3
    }
}
```
- A. The code does not compile because of line r1.
  - B. The code does not compile because of line r2.
  - C. The code does not compile because of line r3.
  - D. The code compiles, but a stack trace is printed at runtime.
45. If a try statement has catch blocks for both `IllegalArgumentException` and `ClassCastException`, then which of the following statements is correct?
- A. The catch block for `IllegalArgumentException` must appear before the catch block for `ClassCastException`.
  - B. The catch block for `ClassCastException` must appear before the catch block for `IllegalArgumentException`.
  - C. The catch blocks for these two exception types can be declared in any order.
  - D. A try statement cannot be declared with these two catch block types because they are incompatible.

**46.** What is the output of the following application?

```
package broken;
class Problem implements RuntimeException {}
public class BiggerProblem extends Problem {
    public static void main(String uhOh[]) {
        try {
            throw new BiggerProblem();
        } catch (BiggerProblem re) {
            System.out.print("Problem?");
        } catch (Problem e) {
            System.out.print("Handled");
        } finally {
            System.out.print("Fixed!");
        }
    }
}
```

- A.** Problem?Fixed!
- B.** Handled.Fixed!
- C.** Problem?Handled.Fixed!
- D.** The code does not compile.

**47.** What is the output of the following application?

```
package lighting;
interface Source {
    void flipSwitch() throws Exception;
}
public class LightBulb implements Source {
    public void flipSwitch() {
        try {
            throws new RuntimeException("Circuit Break!");
        } finally {
            System.out.print("Flipped!");
        }
    }
    public static void main(String... electricity) throws Throwable {
        final Source bulb = new LightBulb();
        bulb.flipSwitch();
    }
}
```

- A. A stack trace for a `RuntimeException`
  - B. Flipped!, followed by a stack trace for a `RuntimeException`
  - C. The code does not compile because `flipSwitch()` is an invalid method override.
  - D. The code does not compile for another reason.
48. Given an application that hosts a website, which of the following would most likely result in a `java.lang.Error` being thrown?
- A. Two users try to register an account at the same time.
  - B. The application temporarily loses connection to the network.
  - C. A user enters their password incorrectly.
  - D. The application runs out of memory.
49. Given that `FileNotFoundException` is a subclass of `IOException`, what is the output of the following application?

```
package storage;
import java.io.*;
public class Backup {
    public void performBackup() {
        try {
            throw new IOException("Disk not found");
        } catch (Exception e) {
            try {
                throw new FileNotFoundException("File not found");
            } catch (FileNotFoundException e) { // z1
                System.out.print("Failed");
            }
        }
    }
    public static void main(String... files) {
        new Backup().performBackup(); // z2
    }
}
```

- A. Failed
- B. The application compiles but a stack trace is printed at runtime.
- C. The code does not compile because of line z1.
- D. The code does not compile because of line z2.

50. What is the output of the following application?

```
package bed;
public class Sleep {
    public static void snore() {
        try {
            String sheep[] = new String[3];
            System.out.print(sheep[3]);
        } catch (RuntimeException e) {
            System.out.print("Awake!");
        } finally {
            throw new Exception(); // x1
        }
    }
    public static void main(String... sheep) { // x2
        new Sleep().snore(); // x3
    }
}
```

- A. Awake!, followed by a stack trace
- B. The code does not compile because of line x1.
- C. The code does not compile because of line x2.
- D. The code does not compile because of line x3.

# Chapter 9



# Working with Selected Classes from the Java API

---

**THE OCA EXAM TOPICS COVERED  
IN THIS PRACTICE TEST INCLUDE  
THE FOLLOWING:**

✓ **Working with Selected classes from the Java API**

- Manipulate data using the `StringBuilder` class and its methods
- Create and manipulate `String`s
- Create and manipulate calendar data using classes from `java.time.LocalDateTime`, `java.time.LocalDate`, `java.time.LocalTime`, `java.time.format.DateTimeFormatter`, `java.time.Period`
- Declare and use an `ArrayList` of a given type
- Write a simple Lambda expression that consumes a Lambda Predicate expression

1. What is the best reason for using `StringBuilder` instead of `String`?
  - A. `StringBuilder` adds support for multiple threads.
  - B. `StringBuilder` can use `==` to compare values.
  - C. `StringBuilder` saves memory by reducing the number of objects created.
  - D. `StringBuilder` supports different languages and encodings.
2. What is not true about a `String`?
  - A. It can be created without coding a call to a constructor.
  - B. It can be reused via the string pool.
  - C. It is final.
  - D. It is mutable.
3. Which of the following creates a `StringBuilder` with a different value than the other options?
  - A. `new StringBuilder().append("clown")`
  - B. `new StringBuilder("clown")`
  - C. `new StringBuilder("cl").insert(2, "own")`
  - D. All of them create the same value.
4. What is the output of the following?

```
StringBuilder teams = new StringBuilder("333");
teams.append(" 806");
teams.append(" 1601");
System.out.print(teams);
```

  - A. 333
  - B. 333 806 1601
  - C. The code compiles but outputs something else.
  - D. The code does not compile.
5. How many of the types `ArrayList`, `List`, and `Object` can fill in the blank to produce code that compiles?

```
List frisbees = new _____();
```

  - A. None
  - B. One
  - C. Two
  - D. Three

6. What is the output of the following?

```
List<String> tools = new ArrayList<>();  
tools.add("hammer");  
tools.add("nail");  
tools.add("hex key");  
System.out.println(tools.get(1));
```

- A. hammer
- B. hex key
- C. nail
- D. None of the above

7. What is the result of the following code?

```
StringBuilder sb = new StringBuilder("radical")  
    .insert(sb.length(), "robots");  
System.out.println(sb);
```

- A. radicarobots
- B. radicalrobots
- C. The code does not compile.
- D. The code compiles but throws an exception at runtime.

8. What is the output of the following?

```
List<String> museums = new ArrayList<>(1);  
museums.add("Natural History");  
museums.add("Science");  
museums.add("Art");  
museums.remove(2);  
System.out.println(museums);
```

- A. [Natural History, Science]
- B. [Natural History, Art, Science]
- C. The code does not compile.
- D. The code compiles but throws an exception at runtime.

9. What is the output of the following?

```
12: StringBuilder b = new StringBuilder("12");  
13: b = b.append("3");  
14: b.reverse();  
15: System.out.println(b.toString());
```

- A. 12  
B. 123  
C. 321  
D. The code does not compile.
10. What is the main benefit of a lambda expression?  
A. It allows you to convert a primitive to a wrapper class.  
B. It allows you to change the bytecode while the application is running.  
C. It allows you to inherit from multiple classes.  
D. It allows you to write code that has the execution deferred.
11. What is the output of the following?
- ```
5: StringBuilder line = new StringBuilder("-");  
6: StringBuilder anotherLine = line.append("-");  
7: System.out.print(line == anotherLine);  
8: System.out.print(" ");  
9: System.out.print(line.length());
```
- A. false 1  
B. false 2  
C. true 1  
D. true 2
12. The author of this method forgot to include the data type. Which of the following reference types can fill in the blank to complete this method?
- ```
public static void secret(____ mystery) {  
    mystery.add("metal");  
    String str = mystery.get(0);  
    int num = mystery.length();  
}
```
- A. ArrayList  
B. ArrayList<String>  
C. StringBuilder  
D. None of the above
13. Which portion of code can be removed so that this line of code continues to compile?
- ```
Predicate<StringBuilder> p = (StringBuilder b) -> {return true;};
```
- A. Remove StringBuilder b  
B. Remove ->  
C. Remove { and ;}  
D. Remove { return and ;}

14. What is the output of the following?

```
20: List<Character> chars = new ArrayList<>();  
21: chars.add('a');  
22: chars.add('b');  
23: chars.set(1, 'c');  
24: chars.remove(0);  
25: System.out.print(chars.size() + " " + chars.contains('b'));  
  
A. 1 false  
B. 1 true  
C. 2 false  
D. 2 true
```

15. What is the output of the following?

```
12: String b = "12";  
13: b += "3";  
14: b.reverse();  
15: System.out.println(b.toString());  
  
A. 12  
B. 123  
C. 321  
D. The code does not compile.
```

16. How many of these lines fail to compile?

```
Predicate<String> pred1 = s -> false;  
Predicate<String> pred2 = (s) -> false;  
Predicate<String> pred3 = String s -> false;  
Predicate<String> pred4 = (String s) -> false;
```

- A. One
- B. Two
- C. Three
- D. Four

17. What does the following do?

```
public class Shoot {  
    interface Target {  
        boolean needToAim(double angle);  
    }  
    static void prepare(double angle, Target t) {  
        boolean ready = t.needToAim(angle); // k1
```

```
        System.out.println(ready);
    }
    public static void main(String[] args) {
        prepare(45, d -> d > 5 || d < -5); // k2
    }
}
```

- A. It prints true.  
B. It prints false.  
C. It doesn't compile due to line k1.  
D. It doesn't compile due to line k2.
18. What is the output of the following?
- ```
String teams = new String("694");
teams.concat(" 1155");
teams.concat(" 2265");
teams.concat(" 2869");
System.out.println(teams);
```
- A. 694  
B. 694 1155 2265 2869  
C. The code compiles but outputs something else.  
D. The code does not compile.
19. Which of these classes are in the `java.util` package?
- I. `ArrayList`  
II. `LocalDate`  
III. `String`
- A. I only  
B. II only  
C. I and II  
D. I, II, and III
20. Which of the answer choices results in a different value being output than the other three choices?

```
StringBuilder sb = new StringBuilder("radical ");
sb = _____;
System.out.print(sb);
```

- A. new StringBuilder("radical ")  
.append("robots")
  - B. new StringBuilder("radical ")  
.delete(1, 100)  
.append("obots")  
.insert(1, "adical r")
  - C. new StringBuilder("radical ")  
.insert(7, "robots")
  - D. new StringBuilder("radical ")  
.insert(sb.length(), "robots")
21. What is the output of the following?
- ```
String[] array = {"Natural History", "Science"};  
List<String> museums = Arrays.asList(array);  
museums.set(0, "Art");  
System.out.println(museums.contains("Art"));
```
- A. true
  - B. false
  - C. The code does not compile.
  - D. The code compiles but throws an exception at runtime.

22. Which is a true statement?
- A. If s.contains("abc") is true, then s.equals("abc") is also true.
  - B. If s.contains("abc") is true, then s.startsWith("abc") is also true.
  - C. If s.startsWith("abc") is true, then s.equals("abc") is also true.
  - D. If s.startsWith("abc") is true, then s.contains("abc") is also true.

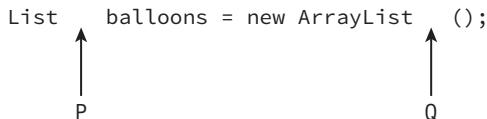
23. What is the output of the following?
- ```
20: List<Character> chars = new ArrayList<>();  
21: chars.add('a');  
22: chars.add('b');  
23: chars.set(1, 'c');  
24: chars.remove(0);  
25: System.out.print(chars.length());
```
- A. 0
  - B. 1
  - C. 2
  - D. None of the above

- 24.** The author of this method forgot to include the data type. Which of the following reference types can fill in the blank to complete this method?

```
public static void secret(____ mystery) {
    mystery = mystery.replace("1", "8");
    mystery.startsWith("paper");
    String s = mystery.toString();
}
```

- A.** `ArrayList`
- B.** `String`
- C.** `StringBuilder`
- D.** None of the above

- 25.** Which statement is true about the following figure while ensuring the code continues to compile?



- A.** `<>` can be inserted at position P without making any other changes.
  - B.** `<>` can be inserted at position Q without making any other changes.
  - C.** `<>` can be inserted at both positions P and Q.
  - D.** None of the above
- 26.** Which of the following can fill in the blank to make the code compile?

```
import java.util.function.*;
public class Card {
    public static void main(String[] s) {
        Predicate<String> pred = _____ -> true;
    }
}
```

- A.** `(Integer i)`
- B.** `(Object o)`
- C.** `(String s)`
- D.** None of the above

- 27.** What is the output of the following?

```
5: String line = new String("-");
6: String anotherLine = line.concat("-");
7: System.out.print(line == anotherLine);
8: System.out.print(" ");
9: System.out.print(line.length());
```

- A. false 1  
B. false 2  
C. true 1  
D. true 2
28. What does the following output?
- ```
Predicate dash = c -> c.startsWith("-");  
System.out.println(dash.test("-"));
```
- A. true  
B. false  
C. The code does not compile.  
D. The code compiles but throws an exception at runtime.
29. Of the classes LocalDate, LocalDateTime, LocalTime, and LocalTimeStamp, how many include hours, minutes, and seconds?
- A. One  
B. Two  
C. Three  
D. Four
30. What is the output of the following class?
- ```
1: package rocket;  
2: public class Countdown {  
3:     public static void main(String[] args) {  
4:         String builder = "54321";  
5:         builder = builder.substring(4);  
6:         System.out.println(builder.charAt(2));  
7:     }  
8: }
```
- A. 2  
B. 3  
C. 4  
D. None of the above
31. Which equivalent code can replace `i -> i != 0` in the following line?
- ```
Predicate<Integer> ip = i -> i != 0;
```
- A. `i -> { i != 0 }`  
B. `i -> { i != 0; }`  
C. `i -> { return i != 0 }`  
D. `i -> { return i != 0; }`

**32.** What is the output of the following?

```
LocalDate xmas = LocalDate.of(2016, 12, 25);
xmas.plusDays(-1);
System.out.println(xmas.getDayOfMonth());
```

- A.** 24
- B.** 25
- C.** 26
- D.** None of the above

**33.** What is the output of the following?

```
1: public class Legos {
2:     public static void main(String[] args) {
3:         StringBuilder sb = new StringBuilder();
4:         sb.append("red");
5:         sb.deleteCharAt(0);
6:         sb.delete(1, 2);
7:         System.out.println(sb);
8:     }
9: }
```

- A.** e
- B.** d
- C.** ed
- D.** None of the above

**34.** What does the following output?

```
Predicate clear = c -> c.equals("clear");
System.out.println(clear.test("pink"));
```

- A.** true
- B.** false
- C.** The code does not compile.
- D.** The code compiles but throws an exception at runtime.

**35.** Which starts counting from one rather than zero?

- A.** Array indexes
- B.** The index used by charAt in a String
- C.** The months in a LocalDateTime
- D.** The months in a LocalTime

**36.** Which statement is not true of Predicate?

- A. A boolean is returned from the method it declares.
- B. It is an interface.
- C. The method it declares accepts two parameters.
- D. The method it declares is named test.

**37.** Which of these periods represents a larger amount of time?

```
Period period1 = Period.ofWeeks(1).ofDays(3);
Period period2 = Period.ofDays(10);
```

- A. period1
- B. period2
- C. They represent the same length of time.
- D. None of the above. This code does not compile.

**38.** What is the result of the following?

```
import java.time.*;
import java.time.format.*;

public class HowLong {
    public static void main(String[] args) {
        LocalDate newYears = LocalDate.of(2017, 1, 1);
        Period period = Period.ofDays(1);
        DateTimeFormatter format = DateTimeFormatter.ofPattern("MM-dd-yyyy");
        System.out.print(format.format(newYears.minus(period)));
    }
}
```

- A. 01-01-2017
- B. 12-31-2016
- C. The code does not compile.
- D. The code compiles but throws an exception at runtime.

**39.** Which of the following can fill in the blank so the following code prints true?

```
String happy = " :) - (: ";
String really = happy.trim();
String question = _____;
System.out.println(really.equals(question));
```

- A. `happy.substring(0, happy.length() - 1)`
- B. `happy.substring(0, happy.length())`
- C. `happy.substring(1, happy.length() - 1)`
- D. `happy.substring(1, happy.length())`
- 40.** Which is not a true statement about the `Period` class?
- A. A `Period` is immutable.
- B. A `Period` is typically used for adding or subtracting time from dates.
- C. You can create a `Period` representing 2 minutes.
- D. You can create a `Period` representing 5 years.
- 41.** What is the output of the following class?
- ```
1: package rocket;
2: public class Countdown {
3:     public static void main(String[] args) {
4:         StringBuilder builder = new StringBuilder("54321");
5:         builder.substring(2);
6:         System.out.println(builder.charAt(1));
7:     }
8: }
```
- A. 1
- B. 2
- C. 3
- D. 4
- 42.** What does the following output?
- ```
List<Integer> pennies = new ArrayList<>();
pennies.add(3);
pennies.add(2);
pennies.add(1);
pennies.remove(2);
System.out.println(pennies);
```
- A. [3, 1]
- B. [3, 2]
- C. The code does not compile.
- D. The code compiles but throws an exception at runtime.

- 43.** The author of this method forgot to include the data type. Which of the following reference types can best fill in the blank to complete this method?

```
public static void secret(____ mystery) {  
    char ch = mystery.charAt(3);  
    mystery = mystery.insert(1, "more");  
    int num = mystery.length();  
}
```

- A.** ArrayList
- B.** String
- C.** StringBuilder
- D.** None of the above

- 44.** What is the smallest unit you can add to a LocalTime object?

- A.** Second
- B.** Millisecond
- C.** Nanosecond
- D.** Picosecond

- 45.** What is the result of the following?

```
import java.time.*;  
import java.time.format.*;  
  
public class HowLong {  
    public static void main(String[] args) {  
        LocalDate newYears = LocalDate.of(2017, 1, 1);  
        Period period = Period.ofDays(1);  
        DateTimeFormatter format = DateTimeFormatter.ofPattern("mm-dd-yyyy");  
        System.out.print(format.format(newYears.minus(period)));  
    }  
}
```

- A.** 01-01-2017
- B.** 12-31-2016
- C.** The code does not compile.
- D.** The code compiles but throws an exception at runtime.

**46.** Which of the following types can you pass as a parameter to the `replace()` method on the `String` class?

- I. `char`
  - II. `String`
  - III. `StringBuilder`
- A. I
  - B. I and II
  - C. II and III
  - D. I, II, and III

**47.** How many lines does this code output?

```
import java.util.*;
import java.util.function.*;

public class PrintNegative {
    public static void main(String[] args) {
        List<String> list = new ArrayList<>();
        list.add("-5");
        list.add("0");
        list.add("5");
        print(list, e -> e < 0);
    }
    public static void print(List<String> list, Predicate<Integer> p) {
        for (String num : list)
            if (p.test(Integer.parseInt(num)))
                System.out.println(num);
    }
}
A. One
B. Two
C. None. The code does not compile.
D. None. The code throws an exception at runtime.
```

**48.** What is the output of the following?

```
12: List<String> magazines = new ArrayList();
13: magazines.add("Readers Digest");
14: magazines.add("People");
15: magazines.clear();
16: magazines.add("The Economist");
17: magazines.remove(1);
18: System.out.println(magazines.size());
```

- A. 0
  - B. 1
  - C. The code does not compile.
  - D. The code compiles but throws an exception at runtime.
49. What is the output of the following?
- ```
public class Costume {  
    public static void main(String[] black) {  
        String witch = 'b';  
        String tail = "lack";  
        witch = witch.concat(tail);  
        System.out.println(witch);  
    }  
}
```
- A. b
  - B. black
  - C. The code does not compile.
  - D. The code compiles but throws an exception at runtime.
50. What is the result of the following?
- ```
LocalDate xmas = LocalDate.of(2016, 12, 25);  
xmas.setYear(2017);  
System.out.println(xmas.getYear());
```
- A. 2016
  - B. 2017
  - C. The code does not compile.
  - D. The code compiles but throws an exception at runtime.



# Chapter 10



## OCA Practice Exam

---

This chapter contains 80 questions and is designed to simulate a real OCA exam. While previous chapters were focused on a specific set of objectives, this chapter covers all of the objectives on the exam. We recommend you take this exam only after you score well on the questions in the individual chapters.

For this chapter, you should try to simulate the real exam experience as much as possible. This means setting aside 150 minutes of uninterrupted time to complete the test, as well as not looking at any reference material while taking the exam. If you don't know an answer to a question, complete it as best you can and move on to the next question, just as you would on a real exam.

Remember, the exam permits writing material, such as a whiteboard. If you do not have a whiteboard handy, you can just use blank sheets of paper and a pencil. If you do well on this test, then you are hopefully ready to take the real exam. With that said, good luck!

1. What is the output if this class is run with `java Indexing cars carts`?

```
public class Indexing {  
    public static void main(String... books) {  
        StringBuilder sb = new StringBuilder();  
        for (String book : books)  
            sb.insert(sb.indexOf("c"), book);  
        System.out.println(sb);  
    }  
}
```

- A. cars
- B. cars carts
- C. ccars arts
- D. The code does not compile.
- E. The code compiles but throws an exception at runtime.

2. Fill in the blanks: The operators `+=`, `—`, `*`, `/`, `==`, and `++` are listed in increasing or the same level of operator precedence. (Choose two.)

- A. `—, +, =, --`
- B. `%, *, /, +`
- C. `=, +, /, *`
- D. `^, *, -, ==`
- E. `*, /, %, --`

3. Which of the following are valid JavaBean signatures? (Choose three.)

- A. `public byte getNose(String nose)`
- B. `public void setHead(int head)`
- C. `public String getShoulders()`
- D. `public long isMouth()`
- E. `public void gimmeEars()`
- F. `public boolean isToes()`

4. Which of the following are true? (Choose two.)

```
20: int[] crossword [] = new int[10][20];  
21: for (int i = 0; i < crossword.length; i++)  
22:     for (int j = 0; j < crossword.length; j++)  
23:         crossword[i][j] = 'x';  
24: System.out.println(crossword.size());
```

- A. One line needs to be changed for this code to compile.
  - B. Two lines need to be changed for this code to compile.
  - C. Three lines need to be changed for this code to compile.
  - D. If the code is fixed to compile, none of the cells in the 2D array have a value of 0.
  - E. If the code is fixed to compile, half of the cells in the 2D array have a value of 0.
  - F. If the code is fixed to compile, all of the cells in the 2D array have a value of 0.
5. Which of the following statements about `java.lang.Error` are most accurate? (Choose two.)
- A. An `Error` should be thrown if a file system resource becomes temporarily unavailable.
  - B. An application should never catch an `Error`.
  - C. `Error` is a subclass of `Exception`, making it a checked exception.
  - D. It is possible to catch and handle an `Error` thrown in an application.
  - E. An `Error` should be thrown if a user enters invalid input.
6. Given a class that uses the following `import` statements, which class would be automatically accessible without using its full package name? (Choose three.)
- ```
import forest.Bird;
import jungle.tree.*;
import savana.*;
```
- A. `forest.Bird`
  - B. `savana.sand.Wave`
  - C. `jungle.tree.Huicungo`
  - D. `java.lang.Object`
  - E. `forest.Sloth`
  - F. `forest.ape.bonobo`
7. How many of the following variables represent immutable objects?
- ```
ArrayList l = new ArrayList();
String s = new String();
StringBuilder sb = new StringBuilder();
LocalDateTime t = LocalDateTime.now();
```
- A. None
  - B. One
  - C. Two
  - D. Three
  - E. Four
  - F. None of the above—this code doesn't compile.

8. What is the output of the following?

```
StringBuilder builder = new StringBuilder("Leaves growing");
do {
    builder.delete(0, 5);
} while (builder.length() > 5);
System.out.println(builder);
```

- A. Leaves growing
- B. ing
- C. wing
- D. The code does not compile.
- E. The code compiles but throws an exception at runtime.

9. What is the output of the following application?

```
package reality;
public class Equivalency {
    public static void main(String[] edges) {
        final String ceiling = "up";
        String floor = new String("up");
        final String wall = new String(floor);
        System.out.print((ceiling==wall)
            +" "+(floor==wall)
            +" "+ceiling.equals(wall));
    }
}
```

- A. false false false
- B. true true true
- C. false true true
- D. false false true
- E. It does not compile.

10. How many times does the following code print true?

```
1:  public class Giggles {
2:      public static void main(String[] args) {
3:          String lol = "lol";
4:          System.out.println(lol.toUpperCase() == lol);
5:          System.out.println(lol.toUpperCase() == lol.toUpperCase());
6:          System.out.println(lol.toUpperCase().equals(lol));
7:          System.out.println(lol.toUpperCase().equals(lol.toUpperCase())));
8:          System.out.println(lol.toUpperCase().equalsIgnoreCase(lol));
```

```
9:         System.out.println(lol.toUpperCase())
10:            .equalsIgnoreCase(lol.toUpperCase()));
11:    } }
```

- A. One
- B. Two
- C. Three
- D. Four
- E. Five
- F. None. The code does not compile.

11. Which lines can be removed together without stopping the code from compiling and while printing the same output? (Choose three.)

```
14: String race = "";
15: outer:
16: do {
17: inner:
18:   do {
19:     race += "x";
20:   } while (race.length() <= 4);
21: } while (race.length() < 4);
22: System.out.println(race);
```

- A. Lines 15 and 17
- B. Lines 15, 16, and 21
- C. Line 17
- D. Lines 17, 18, and 20
- E. Line 20
- F. Line 21

12. Which of the following do not compile when filling in the blank? (Choose two.)

```
long bigNum = _____;
```

- A. 1234
- B. 1234.0
- C. 1234.0L
- D. 1234l
- E. 1234L
- F. 1\_234

13. How many lines does this program print?

```
import java.time.*;
public class OnePlusOne {
    public static void main(String... nums) {
        LocalTime time = LocalTime.of(1, 11);
        while (time.getHour() < 1) {
            time.plusHours(1);
            System.out.println("in loop");
        }
    }
}
```

- A. None
- B. One
- C. Two
- D. This is an infinite loop.
- E. The code does not compile.

14. What is the result of running the following program?

```
1: package fun;
2: public class Sudoku {
3:     static int[][] game;
4:
5:     public static void main(String args[]) {
6:         game[3][3] = 6;
7:         Object[] obj = game;
8:         obj[3] = 'X';
9:         System.out.println(game[3][3]);
10:    }
11: }
```

- A. 6
- B. X
- C. The code does not compile.
- D. The code compiles but throws a `NullPointerException` at runtime.
- E. The code compiles but throws a different exception at runtime.
- F. The output is not guaranteed.

15. Which of the following use generics and compile without warnings? (Choose two.)

- A. List<String> a = new ArrayList();
- B. List<> b = new ArrayList();
- C. List<String> c = new ArrayList<>();
- D. List<> d = new ArrayList<>();
- E. List<String> e = new ArrayList<String>();
- F. List<> f = new ArrayList<String>();

16. Which of the following are true right before the main() method ends? (Choose two.)

```
public static void main(String[] args) {  
    String shoe1 = new String("sandal");  
    String shoe2 = new String("flip flop");  
    String shoe3 = new String("croc");  
  
    shoe1 = shoe2;  
    shoe2 = shoe3;  
    shoe3 = shoe1;  
}
```

- A. No objects are eligible for garbage collection.
- B. One object is eligible for garbage collection.
- C. Two objects are eligible for garbage collection.
- D. No objects are guaranteed to be garbage collected.
- E. One object is guaranteed to be garbage collected.
- F. Two objects are guaranteed to be garbage collected.

17. How many lines of the following application do not compile?

```
package ocean;  
class BubbleException extends Exception {}  
class Fish {  
    Fish getFish() throws BubbleException {  
        throw new RuntimeException("fish!");  
    }  
}  
public final class Clownfish extends Fish {  
    public final Clownfish getFish() {  
        throw new RuntimeException("clown!");  
    }  
}  
public static void main(String[] bubbles) {  
    final Fish f = new Clownfish();
```

```
        f.getFish();
        System.out.println("swim!");
    }
}
```

- A. None. The code compiles and prints swim!.
- B. None. The code compiles and prints a stack trace.
- C. One
- D. Two
- E. Three
- 18.** How many lines does this code output?

```
import java.util.*;
import java.util.function.*;

public class PrintNegative {

    public static void main(String[] args) {
        List<Integer> list= new ArrayList<>();
        list.add(-5);
        list.add(0);
        list.add(5);
        print(list, e -> e < 0);
    }

    public static void print(List<Integer> list, Predicate<Integer> p) {
        for (Integer num : list)
            if (p.test(num))
                System.out.println(num);
    }
}
```

- A. One
- B. Two
- C. Three
- D. None. It doesn't compile.
- E. None. It throws an exception at runtime.
- 19.** Which keywords are required with a `try` statement?
- I. `finalize`
- II. `catch`
- III. `throws`
- IV. `finally`

- A. I only
  - B. II only
  - C. III only
  - D. IV only
  - E. I or II, or both
  - F. None of the above
20. What is the output of the following?
- ```
12: int result = 8;
13: loop: while (result > 7) {
14:     result++;
15:     do {
16:         result--;
17:     } while (result > 5);
18:     break loop;
19: }
20: System.out.println(result);
```
- A. 5
  - B. 7
  - C. 8
  - D. The code does not compile.
  - E. The code compiles but throws an exception at runtime.
21. What is the result of compiling and executing the following application?

```
package reptile;
public class Alligator {
    static int teeth;
    double scaleToughness;
    public Alligator() {
        teeth++;
    }
    public void snap(int teeth) {
        System.out.print(teeth+" ");
        teeth--;
    }
    public static void main(String[] unused) {
        new Alligator().snap(teeth);
        new Alligator().snap(teeth);
    }
}
```

- A. 0 1
  - B. 1 1
  - C. 1 2
  - D. 2 2
  - E. The code does not compile.
  - F. The code compiles but produces an exception at runtime.
22. What is the output of the following?
- ```
public class Costume {  
    public static void main(String[] black) {  
        String witch = "b";  
        String tail = "lack";  
        witch.concat(tail);  
        System.out.println(witch);  
    }  
}
```
- A. b
  - B. black
  - C. lack
  - D. The code does not compile.
  - E. The code compiles but throws an exception at runtime.
23. Which modifiers can be independently applied to an interface method? (Choose three.)
- A. default
  - B. protected
  - C. static
  - D. private
  - E. final
  - F. abstract
24. What is the output of the following?

```
public class Shoelaces {  
    public static void main(String[] args) {  
        String tie = null;  
        while (tie = null)  
            tie = "shoelace";  
        System.out.print(tie);  
    }  
}
```

- A. null
  - B. shoelace
  - C. shoelaceshoelace
  - D. The code does not compile.
  - E. This is an infinite loop.
  - F. The code compiles but throws an exception at runtime.
25. What statements are true about compiling a Java class file? (Choose two.)
- A. If the file does not contain a package statement, then the compiler considers the class part of the `java.lang` package.
  - B. The compiler assumes every class implicitly imports the `java.lang.*` package.
  - C. The compiler assumes every class implicitly imports the `java.util.*` package.
  - D. Java requires every file to declare a package statement.
  - E. Java requires every file to declare at least one `import` statement.
  - F. If the class declaration does not extend another class, then it implicitly extends the `java.lang.Object` class.
26. What is the output of the following application?

```
package woods;
interface Plant {
    default String grow() { return "Grow!"; }
}
interface Living {
    public default String grow() { return "Growing!"; }
}
public class Tree implements Plant, Living { // m1
    public String grow(int height) { return "Super Growing!"; }
    public static void main(String[] leaves) {
        Plant p = new Tree(); // m2
        System.out.print(((Living)p).grow()); // m3
    }
}
```

- A. Grow!
- B. Growing!
- C. Super Growing!
- D. It does not compile because of line m1.
- E. It does not compile because of line m2.
- F. It does not compile because of line m3.

27. What is the result of the following?

```
public static void main(String... args) {  
    String name = "Desiree";  
    int _number = 694;  
    boolean profit$$$;  
    System.out.println(name + " won. "  
        + _number + " profit? " + profit$$$);  
}
```

- A. The declaration of name does not compile.
  - B. The declaration of \_number does not compile.
  - C. The declaration of profit\$\$\$ does not compile.
  - D. The println statement does not compile.
  - E. The code compiles and runs successfully.
  - F. The code compiles and throws an exception at runtime.
28. Fill in the blanks: Given a variable x, \_\_\_\_\_ decreases the value of x by 1 and returns the original value, while \_\_\_\_\_ increases the value of x by 1 and returns the new value.
- A. x--, ++x
  - B. x--, x++
  - C. --x, x++
  - D. --x, ++x

29. Given the following two classes in the same package, which constructors contain compiler errors? (Choose three.)

```
public class Big {  
    public Big(boolean stillIn) {  
        super();  
    }  
}  
  
public class Trouble extends Big {  
    public Trouble() {}  
    public Trouble(int deep) {  
        super(false);  
        this();  
    }  
    public Trouble(String now, int... deep) {  
        this(3);  
    }  
}
```

```
    }
    public Trouble(long deep) {
        this("check",deep);
    }
    public Trouble(double test) {
        super(test>5 ? true : false);
    }
}
```

- A. public Big(boolean stillIn)
  - B. public Trouble()
  - C. public Trouble(int deep)
  - D. public Trouble(String now, int... deep)
  - E. public Trouble(long deep)
  - F. public Trouble(double test)
30. Which of the following can replace the comment so this code outputs 100? (Choose two.)
- ```
public class Stats {
    // INSERT CODE
    public static void main(String[] math) {
        System.out.println(max - min);
    }
}
```
- A. final int min, max = 100;
  - B. final int min = 0, max = 100;
  - C. int min, max = 100;
  - D. int min = 0, max = 100;
  - E. static int min, max = 100;
  - F. static int min = 0, max = 100;
31. Which of the following statements are true about Java operators and statements? (Choose two.)
- A. Both right-hand sides of the ternary expression will be evaluated at runtime.
  - B. A switch statement may contain at most one default statement.
  - C. A single if-then statement can have multiple else statements.
  - D. The | and || operator are interchangeable, always producing the same results at runtime.
  - E. The ! operator may not be applied to numeric expressions.

**32.** What is the output of the following?

```
1:  public class Legos {  
2:      public static void main(String[] args) {  
3:          StringBuilder sb = new StringBuilder();  
4:          sb.append("red");  
5:          sb.deleteCharAt(0);  
6:          sb.delete(1, 1);  
7:          System.out.println(sb);  
8:      }  
9:  }
```

- A.** r
- B.** e
- C.** ed
- D.** red
- E.** The code does not compile.
- F.** The code compiles but throws an exception at runtime.

**33.** Which of the following is a valid method name in Java? (Choose two.)

- A.** \_\_\_\_\_()
- B.** %run()
- C.** check-Activity()
- D.** \$Hum2()
- E.** sing\\3()
- F.** po#ut ()

**34.** Which of the following statements about inheritance are true? (Choose two.)

- A.** Inheritance is better than using `static` methods for accessing data in other classes.
- B.** Inheritance allows a method to be overridden in a subclass, possibly changing the expected behavior of other methods in a superclass.
- C.** Inheritance allows objects to inherit commonly used attributes and methods.
- D.** It is possible to create a Java class that does not inherit from any other.
- E.** Inheritance tends to make applications more complicated.

**35.** Which of the following statements about Java are true?

- I.** The `java` command uses `.` to separate packages.
- II.** Java supports functional programming.
- III.** Java is object oriented.
- IV.** Java supports polymorphism.

- A. I only
  - B. II only
  - C. II and III
  - D. I, III, and IV
  - E. I, II, III, and IV
  - F. None are true.
- 36.** What is the output of the following?
- ```
String[][] listing = new String[][] { { "Book", "34.99" },  
    { "Game", "29.99" }, { "Pen", ".99" } };  
System.out.println(listing.length + " " + listing[0].length);
```
- A. 2 2
  - B. 2 3
  - C. 3 2
  - D. 3 3
  - E. The code does not compile.
  - F. The code compiles but throws an exception at runtime.

- 37.** Which of the following variable types is permitted in a switch statement? (Choose three.)
- A. Character
  - B. Byte
  - C. Double
  - D. long
  - E. String
  - F. Object

- 38.** What does the following do?

```
public class Shoot {  
    interface Target {  
        boolean needToAim(double angle);  
    }  
    static void prepare(double angle, Target t) {  
        boolean ready = t.needToAim(angle); // k1  
        System.out.println(ready);  
    }  
    public static void main(String[] args) {  
        prepare(45, d => d > 5 || d < -5); // k2  
    }  
}
```

- A. It prints true.
  - B. It prints false.
  - C. It doesn't compile due to line k1.
  - D. It doesn't compile due to line k2.
  - E. It doesn't compile due to another line.
39. Which of the following is a valid code comment in Java? (Choose three.)
- A. `/** Insert */ in next method **/`
  - B. `/***** Find the kitty cat */`
  - C. `// Is this a bug?`
  - D. `/ Begin method - performStart() /`
  - E. `/** TODO: Call grandma **/`
  - F. `# Updated code by Patti`
40. Given the following two classes, each in a different package, which lines allow the second class to compile when inserted independently? (Choose two.)

```
package food;
public class Grass {
    public static int seeds = 10;
    public static Grass getGrass() {return new Grass();}
}
```

```
package woods;
```

```
// INSERT CODE HERE
public class Deer {
    public void eat() {
        getGrass();
        System.out.print(seeds);
    }
}
```

- A. `import static food.Grass.getGrass;`  
`import static food.Grass.seeds;`
- B. `import static food.*;`
- C. `static import food.Grass.*;`
- D. `import food.Grass.*;`
- E. `static import food.Grass.getGrass;`  
`static import food.Grass.seeds;`
- F. `import static food.Grass.*;`

41. What is the result of the following?

```
import java.util.*;
public class Museums {
    public static void main(String[] args) {
        String[] array = {"Natural History", "Science", "Art"};
        List<String> museums = Arrays.asList(array);
        museums.remove(2);
        System.out.println(museums);
    }
}
```

- A. [Natural History, Science]
- B. [Natural History, Science, Art]
- C. The code does not compile.
- D. The code compiles but throws an exception at runtime.

42. Which of the following substitutions will compile? (Choose two.)

```
public class Underscores {
    public String name = "Sherrin";
    public void massage() {
        int zip = 10017;
    }
}
```

- A. Change name to \_name
- B. Change 10017 to \_10017
- C. Change 10017 to 10017\_
- D. Change 10017 to 10\_0\_17
- E. Change int to \_int

43. What is the result of the following when called as java counting.Binary?

```
package counting;
import java.util.*;
public class Binary {

    public static void main(String[] args) {
        args = new String[] {"0", "1", "01", "10" };

        Arrays.sort(args);
        System.out.println(Arrays.toString(args));
    }
}
```

- A. []  
B. [0, 01, 1, 10]  
C. [0, 01, 10, 1]  
D. [0, 1, 01, 10]  
E. The code does not compile.  
F. The code compiles but throws an exception at runtime.
44. Fill in the blanks: Using the \_\_\_\_\_ and \_\_\_\_\_ modifiers together allows a variable to be accessed from any class, without requiring an instance variable.
- A. final, package-private  
B. class, static  
C. protected, instance  
D. public, static  
E. default, public
45. How many lines does the following code output?
- ```
import java.util.*;  
public class Exams {  
    public static void main(String[] args) {  
        List<String> exams = Arrays.asList("OCA", "OCP");  
        for (String e1 : exams)  
            for (String e2 : exams)  
                System.out.print(e1 + " " + e2);  
                System.out.println();  
    }  
}
```
- A. One  
B. Four  
C. Five  
D. The code does not compile.  
E. The code compiles but throws an exception at runtime.
46. Which of the following are true statements? (Choose two.)
- A. The javac command compiles a source text file into a set of machine instructions.  
B. The java command compiles a .class file into a .java file.  
C. The javac command compiles a .java file into a .class file.  
D. The javac command compiles a source text file into a bytecode file.  
E. The java command compiles a .java file into a .class file.  
F. The javac command compiles a .class file into a .java file.

47. How many of the following lines of code compile?

```
char one = Integer.parseInt("1");
Character two = Integer.parseInt("2");
int three = Integer.parseInt("3");
Integer four = Integer.parseInt("4");
short five = Integer.parseInt("5");
Short six = Integer.parseInt("6");
```

- A. None
- B. One
- C. Two
- D. Three
- E. Four
- F. Five

48. Given the application below, what data types can be inserted into the blank that would allow the code to print 3? (Choose three.)

```
public class Highway {
    public int drive(long car) { return 2; }
    public int drive(double car) { return 3; }
    public int drive(int car) { return 5; }
    public int drive(short car) { return 3; }
    public static void main(String[] gears) {
        _____ value = 5;
        System.out.print(new Highway().drive(value));
    }
}
```

- A. boolean
- B. short
- C. int
- D. byte
- E. long
- F. float

49. How many times does this code print true?

```
import java.time.*;
public class Equality {
    public void main(String[] args) {
        System.out.println(new StringBuilder("zelda")
            == new StringBuilder("zelda"));
```

```
        System.out.println(3 == 3);
        System.out.println("bart" == "bart");
        System.out.println(new int[0] == new int[0]);
        System.out.println(LocalTime.now() == LocalTime.now());
    }
}
```

- A. None
- B. One
- C. Two
- D. Three
- E. The code does not compile.

50. What is the output of the following application?

```
package ballroom;
public class Dance {
    public static void swing(int... beats) throws ClassCastException {
        try {
            System.out.print("1"+beats[2]); // p1
        } catch (RuntimeException e) {
            System.out.print("2");
        } catch (Exception e) {
            System.out.print("3");
        } finally {
            System.out.print("4");
        }
    }
    public static void main(String... music) {
        new Dance().swing(0,0); // p2
        System.out.print("5");
    }
}
```

- A. 145
- B. 1045
- C. 24, followed by a stack trace
- D. 245
- E. The code does not compile because of line p1.
- F. The code does not compile because of line p2.

51. What is the output of the following?

```
List<String> drinks = Arrays.asList("can", "cup");
for (int container = drinks.size(); container > 0; container++) {
    System.out.print(drinks.get(container-1) + ",");
}
```

- A. can,cup,
- B. cup,can,
- C. The code does not compile.
- D. This is an infinite loop.
- E. The code compiles but throws an exception at runtime.

52. Which of the following method signatures are valid declarations of an entry point in a Java application? (Choose three.)

- A. public static void main(String... widgets)
- B. public static void main(String sprockets)
- C. protected static void main(String[] args)
- D. public static int void main(String[] arg)
- E. public static final void main(String []a)
- F. public static void main(String[] data)

53. Given the application below and the choices available, which lines must all be removed to allow the code to compile? (Choose three.)

```
1: package year;
2: public class Seasons {
3:     public static void main(String[] time) {
4:         final long winter = 10;
5:         final byte season = 2;
6:         int fall = 4;
7:         final short summer = 3;
8:         switch(season) {
9:             case 1:
10:                 case winter: System.out.print("winter");
11:                 default:
12:                     case fall: System.out.print("fall");
13:                     case summer: System.out.print("summer");
14:                     default:
15:                 }
16:     }
17: }
```

- A. Line 8
  - B. Line 9
  - C. Line 10
  - D. Line 11
  - E. Line 12
  - F. Line 13
54. Given the application below, which lines do not compile? (Choose three.)

```
package furryfriends;
interface Friend {
    protected String getName(); // h1
}
class Cat implements Friend {
    String getName() { // h2
        return "Kitty";
    }
}
public class Dog implements Friend {
    String getName() throws RuntimeException { // h3
        return "Doggy";
    }
    public static void main(String[] adoption) {
        Friend friend = new Dog(); // h4
        System.out.print(((Cat)friend).getName()); // h5
        System.out.print(((Dog)null).getName()); // h6
    }
}
```

- A. Line h1
  - B. Line h2
  - C. Line h3
  - D. Line h4
  - E. Line h5
  - F. Line h6
55. Which of the following are unchecked exceptions? (Choose three.)
- A. FileNotFoundException
  - B. ArithmeticException
  - C. IOException
  - D. Exception
  - E. IllegalArgumentException
  - F. RuntimeException

56. What is the result of compiling and executing the following application?

```
package ranch;
public class Cowboy {
    private int space = 5;
    private double ship = space < 2 ? 1 : 10; // g1
    public void printMessage() {
        if(ship>1) {
            System.out.println("Goodbye");
        } if(ship<10 && space>=2) System.out.println("Hello"); // g2
        else System.out.println("See you again");
    }
    public static final void main(String... stars) {
        new Cowboy().printMessage();
    }
}
```

- A. It only prints Hello.
  - B. It only prints Goodbye.
  - C. It only prints See you again.
  - D. It does not compile because of line g1.
  - E. It does not compile because of line g2.
  - F. None of the above
57. Given the following three class declarations, which sets of access modifiers can be inserted, in order, into the blank lines below that would allow all of the classes to compile? (Choose three.)

```
package wake;
public class Alarm {
    _____ static int clock;
    _____ long getTime() {return clock;}
}

package wake;
public class Coffee {
    private boolean bringCoffee() { return new Alarm().clock<10;}
}

package sleep;
public class Snooze extends wake.Alarm {
    private boolean checkTime() { return getTime()>10;}
}
```

- A. protected and package-private (blank)
  - B. public and public
  - C. package-private (blank) and protected
  - D. protected and protected
  - E. private and public
  - F. package-private (blank) and package-private (blank)
58. Given that `FileNotFoundException` is a subclass of `IOException` and `Long` is a subclass of `Number`, what is the output of the following application?

```
package materials;

import java.io.*;

class CarbonStructure {
    protected long count;
    public abstract Number getCount() throws IOException; // q1
    public CarbonStructure(int count) { this.count = count; }
}

public class Diamond extends CarbonStructure {
    public Diamond() { super(15); }
    public Long getCount() throws FileNotFoundException { // q2
        return count;
    }
    public static void main(String[] cost) {
        try {
            final CarbonStructure ring = new Diamond(); // q3
            System.out.print(ring.getCount()); // q4
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

- A. 15
- B. It does not compile because of line q1.
- C. It does not compile because of line q2.
- D. It does not compile because of line q3.
- E. It does not compile because of line q4.
- F. It compiles but throws an exception at runtime.

59. How many lines contain a compile error?

```
1: import java.time.*;
2: import java.time.format.*;
3:
4: public class HowLong {
5:     public void main(String h) {
6:         LocalDate newYears = new LocalDate(2017, 1, 1);
7:         Period period = Period.ofYears(1).ofDays(1);
8:         DateTimeFormat format = DateTimeFormat.ofPattern("MM-dd-yyyy");
9:         System.out.print(format.format(newYears.minus(period)));
10:    }
11: }
```

- A. None
- B. One
- C. Two
- D. Three
- E. Four
- F. Five

60. Which of the following statements about try-catch blocks are correct? (Choose two.)

- A. A catch block can never appear after a finally block.
- B. A try block must be followed by a catch block.
- C. A finally block can never appear after a catch block.
- D. A try block must be followed by a finally block.
- E. A try block can have zero or more catch blocks.
- F. A try block can have zero or more finally blocks.

61. What is printed by the following code snippet?

```
int fish = 1 + 2 * 5>=2 ? 4 : 2;
int mammals = 3 < 3 ? 1 : 5>=5 ? 9 : 7;
System.out.print(fish+mammals+"");
```

- A. 49
- B. 13
- C. 18
- D. 99
- E. It does not compile.

**62.** Which of the following statements about objects, reference types, and casting are correct? (Choose three.)

- A. An object can be assigned to an inherited interface reference variable without an explicit cast.
- B. The compiler can prevent all explicit casts that lead to an exception at runtime.
- C. Casting an object to a reference variable does not modify the object in memory.
- D. An object can be assigned to a subclass reference variable without an explicit cast.
- E. An object can be assigned to a superclass reference variable without an explicit cast.
- F. An implicit cast of an object to one of its inherited types can sometimes lead to a `ClassCastException` at runtime.

**63.** What is the output of the following when run as `java EchoFirst seed flower plant?`

```
package unix;

import java.util.*;

public class EchoFirst {

    public static void main(String[] args) {
        int result = Arrays.binarySearch(args, args[0]);
        System.out.println(result);
    }
}
```

- A. 0
- B. 1
- C. 2
- D. The code does not compile.
- E. The code compiles but throws an exception at runtime.
- F. The output is not guaranteed.

**64.** How many objects are eligible for garbage collection at the end of the `main()` method?

```
package store;
public class Shoes {

    static String shoe1 = new String("sandal");
    static String shoe2 = new String("flip flop");

    public static void shopping() {
        String shoe3 = new String("croc");
    }
}
```

```
        shoe2 = shoe1;
        shoe1 = shoe3;
    }

    public static void main(String... args) {
        shopping();
    }
}
```

- A. None  
B. One  
C. Two  
D. Three  
E. The code does not compile.
65. Fill in the blanks: The \_\_\_\_\_ keyword is used in method declarations, the \_\_\_\_\_ keyword is used to guarantee a statement will execute even if an exception is thrown, and the \_\_\_\_\_ keyword is used to throw an exception to the surrounding process.
- A. throw, finally, throws  
B. throws, catch, throw  
C. catch, finally, throw  
D. finally, catch, throw  
E. throws, finally, throw
66. Which statements best describe the result of this code? (Choose two.)

```
package nyc;
public class TouristBus {

    public static void main(String... args) {
        String[] nycTourLoops = new String[] { "Downtown", "Uptown", "Brooklyn" };
        String[] times = new String[] { "Day", "Night" };
        for (int i = 0, j = 0; i < nycTourLoops.length; i++, j++)
            System.out.println(nycTourLoops[i] + " " + times[j]);
    }
}
```

- A. The println causes one line of output.  
B. The println causes two lines of output.  
C. The println causes three lines of output.  
D. The code terminates successfully.  
E. The code throws an exception at runtime.

**67.** Fill in the blanks: Because of \_\_\_\_\_, it is possible to \_\_\_\_\_ a method, which allows Java to support \_\_\_\_\_.

- A. abstract methods, override, inheritance
- B. concrete methods, overload, inheritance
- C. virtual methods, overload, interfaces
- D. inheritance, abstract, polymorphism
- E. virtual methods, override, polymorphism.

**68.** What is the result of the following?

```
package calendar;
public class Seasons {

    public static void seasons(String... names) {
        int l = names[1].length();           // s1
        System.out.println(names[l]);      // s2
    }

    public static void main(String[] args) {
        seasons("Summer", "Fall", "Winter", "Spring");
    }
}
```

- A. Fall
- B. Spring
- C. The code does not compile.
- D. The code throws an exception on line s1.
- E. The code throws an exception on line s2.

**69.** How many lines of the following application contain compilation errors?

```
1: package percussion;
2:
3: interface MakesNoise {}
4: abstract class Instrument implements MakesNoise {
5:     public Instrument(int beats) {}
6:     public void play() {}
7: }
8: public class Drum extends Instrument {
9:     public void play(int count) {}
10:    public void concert() {
11:        super.play(5);
```

```
12:      }
13:      public static void main(String[] beats) {
14:          MakesNoise mn = new Drum();
15:          mn.concert();
16:      }
17: }
```

- A. None. The code compiles and runs without issue.  
B. One  
C. Two  
D. Three  
E. Four
70. What is the output of the following application?

```
package fly;
public class Helicopter {
    public int adjustPropellers(int length, String[] type) {
        length++;
        type[0] = "LONG";
        return length;
    }
    public static void main(String[] climb) {
        final Helicopter h = new Helicopter();
        int length = 5;
        String[] type = new String[1];
        length = h.adjustPropellers(length, type);
        System.out.print(length+", "+type[0]);
    }
}
```

- A. 5, LONG  
B. 6, LONG  
C. 5, null  
D. 6, null  
E. The code does not compile.  
F. The code compiles but throws an exception at runtime.
71. How many lines of the following application do not compile?

```
package castles;
class OpenDoorException extends Exception {}
class CableSnapException extends OpenDoorException {}
```

```
public class Palace {  
    public void openDrawbridge() throws Exception {  
        try {  
            throw new Exception("Problem");  
        } catch (OpenDoorException e) {  
            throw new OpenDoorException();  
        } catch (CableSnapException ex) {  
            try {  
                throw new OpenDoorException();  
            } catch (Exception ex) {  
            } finally {  
                System.out.println("Almost done");  
            }  
        } finally {  
            throw new RuntimeException("Unending problem");  
        }  
    }  
    public static void main(String[] moat) throws IllegalArgumentException {  
        new Palace().openDrawbridge();  
    }  
}
```

- A. None. The code compiles and produces a stack trace at runtime.
- B. One
- C. Two
- D. Three
- E. Four
- F. Five
72. Choose the best answer: \_\_\_\_\_ and \_\_\_\_\_ are two properties that go hand in hand to improve class design by structuring a class with related attributes and actions while protecting the underlying data from access by other classes.
- A. Optimization and platform independence
- B. Platform independence and encapsulation
- C. Platform independence and inheritance
- D. Object orientation and encapsulation
- E. Inheritance and polymorphism
73. What is the output of the following?

```
string bike1 = "speedy";  
string bike2 = new String("speedy");
```

```
boolean test1 = bike1 == bike2;
boolean test2 = bike1.equals(bike2);
System.out.println(test1 + " " + test2);
```

- A. false false
  - B. false true
  - C. true false
  - D. true true
  - E. The code does not compile.
  - F. The code compiles but throws an exception at runtime.
74. What is the output of the following when run as java EchoFirst seed flower plant?

```
package unix;

import java.util.*;

public class EchoFirst {

    public static void main(String[] args) {
        Arrays.sort(args);
        int result = Arrays.binarySearch(args, args[0]);
        System.out.println(result);
    }
}
```

- A. 0
  - B. 1
  - C. 2
  - D. The code does not compile.
  - E. The code compiles but throws an exception at runtime.
  - F. The output is not guaranteed.
75. Which are true statements? (Choose three.)
- A. Every do-while loop can be rewritten as a for-each loop.
  - B. Every for-each loop can be rewritten as a do-while loop.
  - C. Every for-each loop can be rewritten as a traditional for loop.
  - D. Every for-each loop can be rewritten as a while loop.
  - E. Every traditional for loop can be rewritten as a for-each loop.
  - F. Every while loop can be rewritten as a for-each loop.

**76.** How many lines does this program print?

```
import java.time.*;
public class OnePlusOne {
    public static void main(String... nums) {
        LocalDate time = LocalDate.of(1, 11);
        while (time.getHour() < 1) {
            time.plusHours(1);
            System.out.println("in loop");
        }
    }
}
```

- A.** None
- B.** One
- C.** Two
- D.** This is an infinite loop.
- E.** The code does not compile.

**77.** How many objects are eligible for garbage collection immediately before the end of the `main()` method?

```
public class Tennis {
    public static void main(String[] game) {
        String[] balls = new String[1];
        int[] scores = new int[1];
        balls = null;
        scores = null;
    }
}
```

- A.** None
- B.** One
- C.** Two
- D.** Three
- E.** Four

**78.** What is the output of the following?

```
14: int count = 0;
15: LocalDate date = LocalDate.of(2017, Month.JANUARY, 1);
16: while (date.getMonth() != Month.APRIL)
17:     date = date.minusMonths(1);
18:     count++;
19: System.out.println(count);
```

- A. 0
- B. 1
- C. 3
- D. 9
- E. This is an infinite loop.
- F. The code does not compile.

79. How many lines of the following class do not compile?

```
1: package arctic;
2: abstract class Bear {
3:     protected int sing;
4:     protected abstract int grunt();
5:     int sing() {
6:         return sing;
7:     }
8: }
9: public class PolarBear extends Bear {
10:    int grunt() {
11:        sing() += 10;
12:        return super.grunt()+1;
13:        return 10;
14:    }
15: }
```

- A. None, the class compiles without issue.
- B. One
- C. Two
- D. Three
- E. Four
- F. Five

80. In which places is the `default` keyword permitted to be used? (Choose two.)

- A. Access modifier in a class
- B. Execution path in a `switch` statement
- C. Method name
- D. Modifier in an `abstract` interface method
- E. Modifier in an interface method with a body
- F. Variable name