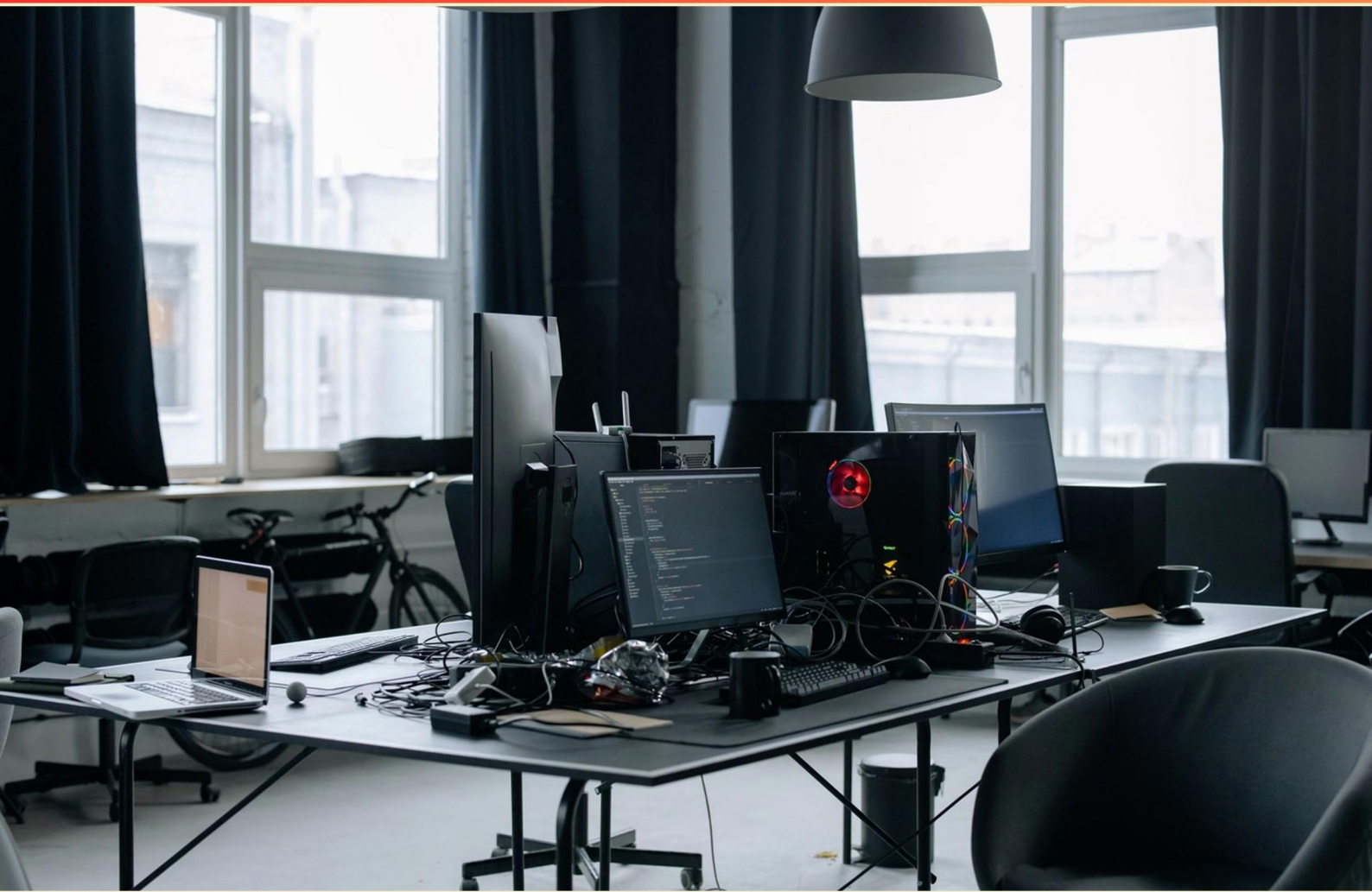


JAVA FPT SOFTWARE (FSOFT) ACADEMY PRACTICE TEST (SAMPLE)

STUDY GUIDE



SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

VISIT US ONLINE FOR
THE FULL VERSION STUDY GUIDE

<https://javasoftacademy.examzify.com>



Copyright © 2026 by Examzify - A Kaluba Technologies Inc. product.

ALL RIGHTS RESERVED.

No part of this book may be reproduced or transferred in any form or by any means, graphic, electronic, or mechanical, including photocopying, recording, web distribution, taping, or by any information storage retrieval system, without the written permission of the author.

Notice: Examzify makes every reasonable effort to obtain accurate, complete, and timely information about this product from reliable sources.

SAMPLE

Table of Contents

Copyright	1
Table of Contents	2
Introduction	3
How to Use This Guide	4
Questions	5
Answers	8
Explanations	10
Next Steps	15

SAMPLE

Introduction

Preparing for a certification exam can feel overwhelming, but with the right tools, it becomes an opportunity to build confidence, sharpen your skills, and move one step closer to your goals. At Examzify, we believe that effective exam preparation isn't just about memorization, it's about understanding the material, identifying knowledge gaps, and building the test-taking strategies that lead to success.

This guide was designed to help you do exactly that.

Whether you're preparing for a licensing exam, professional certification, or entry-level qualification, this book offers structured practice to reinforce key concepts. You'll find a wide range of multiple-choice questions, each followed by clear explanations to help you understand not just the right answer, but why it's correct.

The content in this guide is based on real-world exam objectives and aligned with the types of questions and topics commonly found on official tests. It's ideal for learners who want to:

- Practice answering questions under realistic conditions,
- Improve accuracy and speed,
- Review explanations to strengthen weak areas, and
- Approach the exam with greater confidence.

We recommend using this book not as a stand-alone study tool, but alongside other resources like flashcards, textbooks, or hands-on training. For best results, we recommend working through each question, reflecting on the explanation provided, and revisiting the topics that challenge you most.

Remember: successful test preparation isn't about getting every question right the first time, it's about learning from your mistakes and improving over time. Stay focused, trust the process, and know that every page you turn brings you closer to success.

Let's begin.

How to Use This Guide

This guide is designed to help you study more effectively and approach your exam with confidence. Whether you're reviewing for the first time or doing a final refresh, here's how to get the most out of your Examzify study guide:

1. Start with a Diagnostic Review

Skim through the questions to get a sense of what you know and what you need to focus on. Your goal is to identify knowledge gaps early.

2. Study in Short, Focused Sessions

Break your study time into manageable blocks (e.g. 30 – 45 minutes). Review a handful of questions, reflect on the explanations.

3. Learn from the Explanations

After answering a question, always read the explanation, even if you got it right. It reinforces key points, corrects misunderstandings, and teaches subtle distinctions between similar answers.

4. Track Your Progress

Use bookmarks or notes (if reading digitally) to mark difficult questions. Revisit these regularly and track improvements over time.

5. Simulate the Real Exam

Once you're comfortable, try taking a full set of questions without pausing. Set a timer and simulate test-day conditions to build confidence and time management skills.

6. Repeat and Review

Don't just study once, repetition builds retention. Re-attempt questions after a few days and revisit explanations to reinforce learning. Pair this guide with other Examzify tools like flashcards, and digital practice tests to strengthen your preparation across formats.

There's no single right way to study, but consistent, thoughtful effort always wins. Use this guide flexibly, adapt the tips above to fit your pace and learning style. You've got this!

Questions

SAMPLE

1. Assume a Java program declares `int size = 10; int[] arr = new int[size];` and prints each element in sequence with no separators. Without any assignment to `arr`, what is printed?
 - A. The code will fail to compile
 - B. 0000000000
 - C. 00000000000
 - D. 0
2. Which class is used to write objects to a file?
 - A. `BufferedOutputStream`
 - B. `FileOutputStream`
 - C. `ObjectOutputStream`
 - D. `PrintWriter`
3. Which statement accurately reflects the `equals/hashCode` contract?
 - A. If two objects are equal according to `equals()`, their `hashCode` must be equal
 - B. If two objects are equal, their `hashCode` can be different
 - C. `hashCode` is used only for sorting
 - D. You must override `hashCode` only if objects are mutable
4. Which statement about volatile variables is false?
 - A. They guarantee visibility of writes across threads.
 - B. They establish a happens-before relationship.
 - C. They guarantee atomicity for compound actions.
 - D. They help with ordering of reads and writes.
5. What does `System.out.println(5 + 3);` print?
 - A. 8
 - B. 53
 - C. 58
 - D. 35

- 6. Which description best captures the Java Collections Framework?**
- A. Core interfaces: List, Set, Map, Queue; Implementations: ArrayList, LinkedList, HashSet, TreeSet, LinkedHashSet, HashMap, TreeMap, LinkedHashMap, ArrayDeque.
 - B. Core interfaces: Collection and Iterator; Implementations: Vector and Stack.
 - C. Core interfaces: Map and Collection; Implementations: ArrayList and HashMap.
 - D. Core interfaces: List and Map; Implementations: LinkedList and TreeMap.
- 7. Which command prints Hello! when run with the arguments open and someone?**
- A. Hello!
 - B. Go away someone
 - C. Hello and then error
 - D. Nothing
- 8. Assuming the declarations `int i = 10, j = 12, k = 1; k += i++ - --j;` what is the value printed for k?**
- A. -2
 - B. -1
 - C. 0
 - D. 1
- 9. In a nested loop that builds 11 rows (i from 0 to 10) each row containing 11 elements (j from 0 to 10), how many elements are in each inner row?**
- A. 11
 - B. 10
 - C. 121
 - D. 12
- 10. What will the following program print? The variable c starts at 0 and a flag is true; within a loop the counter increments until certain conditions deactivate the flag. What is printed?**
- A. 3
 - B. 4
 - C. 6
 - D. 7

Answers

SAMPLE

1. C
2. C
3. A
4. C
5. A
6. A
7. A
8. C
9. A
10. C

SAMPLE

Explanations

SAMPLE

1. Assume a Java program declares `int size = 10; int[] arr = new int[size];` and prints each element in sequence with no separators. Without any assignment to `arr`, what is printed?

- A. The code will fail to compile
- B. 0000000000
- C. 0000000000**
- D. 0

Java initializes arrays of primitive types with default values, and for `int` that default is 0. Creating an array of length 10 gives you ten elements, all starting as 0. If you print each element in order with no separators, you get ten zeros in a row: 0000000000. The code compiles cleanly because array creation with a size is valid and no assignments change those values before printing. So the observed output is ten zeros concatenated together.

2. Which class is used to write objects to a file?

- A. `BufferedOutputStream`
- B. `FileOutputStream`
- C. `ObjectOutputStream`**
- D. `PrintWriter`

Serialization is the process of turning a Java object into a sequence of bytes so it can be saved to a file and later reconstructed. `ObjectOutputStream` is designed for this purpose: it writes the state of a `Serializable` object to an `OutputStream`, using methods like `writeObject`. When you pair it with a `FileOutputStream`, you can save the serialized object to a file. Other streams shown in the options write raw bytes (`FileOutputStream`, `BufferedOutputStream`) or human-readable text (`PrintWriter`), not the object's full state, so they aren't suitable for persisting objects. Remember that the class of the object must implement `Serializable`; otherwise an exception is raised during writing. In practice, you'd do something like `new ObjectOutputStream(new FileOutputStream("data.ser"))` and call `writeObject(yourObject)` before closing.

3. Which statement accurately reflects the `equals/hashCode` contract?

- A. If two objects are equal according to `equals()`, their `hashCode` must be equal**
- B. If two objects are equal, their `hashCode` can be different
- C. `hashCode` is used only for sorting
- D. You must override `hashCode` only if objects are mutable

In Java, objects that are considered equal by the `equals` method must produce the same value from `hashCode`. This alignment is essential for hash-based collections like `HashMap` and `HashSet` to find and retrieve elements reliably. If two equal objects could have different hash codes, they would end up in different buckets, making lookups unpredictable or incorrect. Hash codes are used for hashing, not for sorting. Sorting relies on comparisons (like `compareTo` or a `Comparator`), while hashing relies on the `hashCode` value to bucket objects. If you override `equals`, you should also override `hashCode` to maintain this contract. It doesn't matter whether the objects are mutable; what matters is consistency: objects that are equal must always yield the same hash code. If the fields involved in determining equality change after an object is placed in a hash-based collection, it can become unreachable or misplaced unless you carefully manage mutations.

4. Which statement about volatile variables is false?

- A. They guarantee visibility of writes across threads.
- B. They establish a happens-before relationship.
- C. They guarantee atomicity for compound actions.**
- D. They help with ordering of reads and writes.

Volatile variables are about visibility across threads and the ordering of memory operations. When a thread writes to a volatile field, that write becomes visible to other threads that read that same field, and the JVM establishes a happens-before relationship: a write to a volatile happens-before every subsequent read of that volatile. This also helps prevent certain reordering around that access, so the sequence of reads and writes appears in a consistent order relative to the volatile access. But volatile does not guarantee atomicity for compound actions. If you perform a read-modify-write sequence (like counter++), those steps can interleave with other threads and lead to lost updates even with the variable marked volatile. For true atomicity, you need synchronization or atomic classes such as AtomicInteger. So the statement claiming atomicity for compound actions is the false one; the others correctly describe visibility, happens-before, and ordering guarantees provided by volatile.

5. What does System.out.println(5 + 3); print?

- A. 8**
- B. 53
- C. 58
- D. 35

When you add two integers in Java, the plus operator performs arithmetic addition. System.out.println prints the resulting value followed by a newline. Here, both operands are integers, so 5 + 3 equals 8, and that 8 is what gets printed. If one of the operands were a string, the plus operator would do string concatenation, producing something like "53" instead of a numeric 8. The other options come from those different possibilities or mistaken operations, but with two integers this expression yields 8.

6. Which description best captures the Java Collections Framework?

- A. Core interfaces: List, Set, Map, Queue; Implementations: ArrayList, LinkedList, HashSet, TreeSet, LinkedHashMap, HashMap, TreeMap, LinkedHashMap, ArrayDeque.**
- B. Core interfaces: Collection and Iterator; Implementations: Vector and Stack.
- C. Core interfaces: Map and Collection; Implementations: ArrayList and HashMap.
- D. Core interfaces: List and Map; Implementations: LinkedList and TreeMap.

The question is testing knowledge of how the Java Collections Framework is organized: it groups common behavior into a set of core interfaces and then provides typical classes that implement those interfaces. The best description lists the main interfaces such as List, Set, Map, and Queue, and pairs them with representative implementations like ArrayList and LinkedList for lists, HashSet and TreeSet for sets, HashMap and TreeMap for maps, and ArrayDeque for a queue/deque. This captures both the essential categories and the common concrete classes you'd encounter when using the framework. Why this fits best: It accurately reflects the core building blocks of the framework and shows a balanced set of implementations that cover different needs—array-backed versus linked structures for lists; hash-based, tree-based, and insertion-order maps and sets; and a practical queue/deque option. The other options mix in incorrect or incomplete ideas, such as treating Iterator as a core interface, or pairing Map and Collection as a single core group, or suggesting mismatched concrete classes (like Vector/Stack with the wrong concepts).

7. Which command prints Hello! when run with the arguments open and someone?

- A. Hello!
- B. Go away someone
- C. Hello and then error
- D. Nothing

When a program uses command-line arguments to decide its output, it typically checks the provided values and takes a specific path for those matches. With the two arguments open and someone, the code has a branch that prints the literal string Hello! and then stops. So the exact output for these inputs is Hello!, which is why that option is the correct one. The other possibilities would require different input patterns or a different code path (such as assembling a sentence, signaling an error, or producing no output).

8. Assuming the declarations `int i = 10, j = 12, k = 1; k += i++ - --j;` what is the value printed for k?

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

Understanding how a compound assignment interacts with post-increment and pre-decrement, and how Java evaluates expressions from left to right. Start with `i = 10, j = 12, k = 1`. In the expression `k += i++ - --j`, the left part of the subtraction is `i++`, which yields the current value of `i` (10) and then increments `i` to 11. The right part is `--j`, which decrements `j` first (from 12 to 11) and yields the new value (11). So the subtraction becomes `10 - 11 = -1`. The compound assignment `k += (-1)` then becomes `k = k + (-1)`, giving `k = 1 + (-1) = 0`. Therefore, the value printed for `k` is 0.

9. In a nested loop that builds 11 rows (`i` from 0 to 10) each row containing 11 elements (`j` from 0 to 10), how many elements are in each inner row?

- A. 11
- B. 10
- C. 121
- D. 12

When you see a loop that has the inner index go from 0 to 10, inclusive, that means the inner loop runs once for each value in 0, 1, 2, ..., 10. That's 11 iterations. Each iteration adds one element to the current row, so the inner row ends up with 11 elements. If you multiplied for all rows, you'd get 11 rows $\times$ 11 elements per row = 121 elements in total, but the question is asking about a single inner row, which is 11.

10. What will the following program print? The variable c starts at 0 and a flag is true; within a loop the counter increments until certain conditions deactivate the flag. What is printed?

- A. 3
- B. 4
- C. 6**
- D. 7

A loop controlled by a boolean flag and a counter shows how many times the loop body runs before termination. The printed value is the counter's value at the moment the flag becomes false and the loop exits. Since the counter starts at 0 and is incremented each iteration, it takes six increments to reach 6. If the flag is turned off exactly after that sixth increment, the loop ends and 6 is printed. So the output reflects how many iterations occurred before termination, which in this case is six.

SAMPLE

Next Steps

Congratulations on reaching the final section of this guide. You've taken a meaningful step toward passing your certification exam and advancing your career.

As you continue preparing, remember that consistent practice, review, and self-reflection are key to success. Make time to revisit difficult topics, simulate exam conditions, and track your progress along the way.

If you need help, have suggestions, or want to share feedback, we'd love to hear from you. Reach out to our team at hello@examzify.com.

Or visit your dedicated course page for more study tools and resources:

<https://javafsoftacademy.examzify.com>

We wish you the very best on your exam journey. You've got this!

SAMPLE