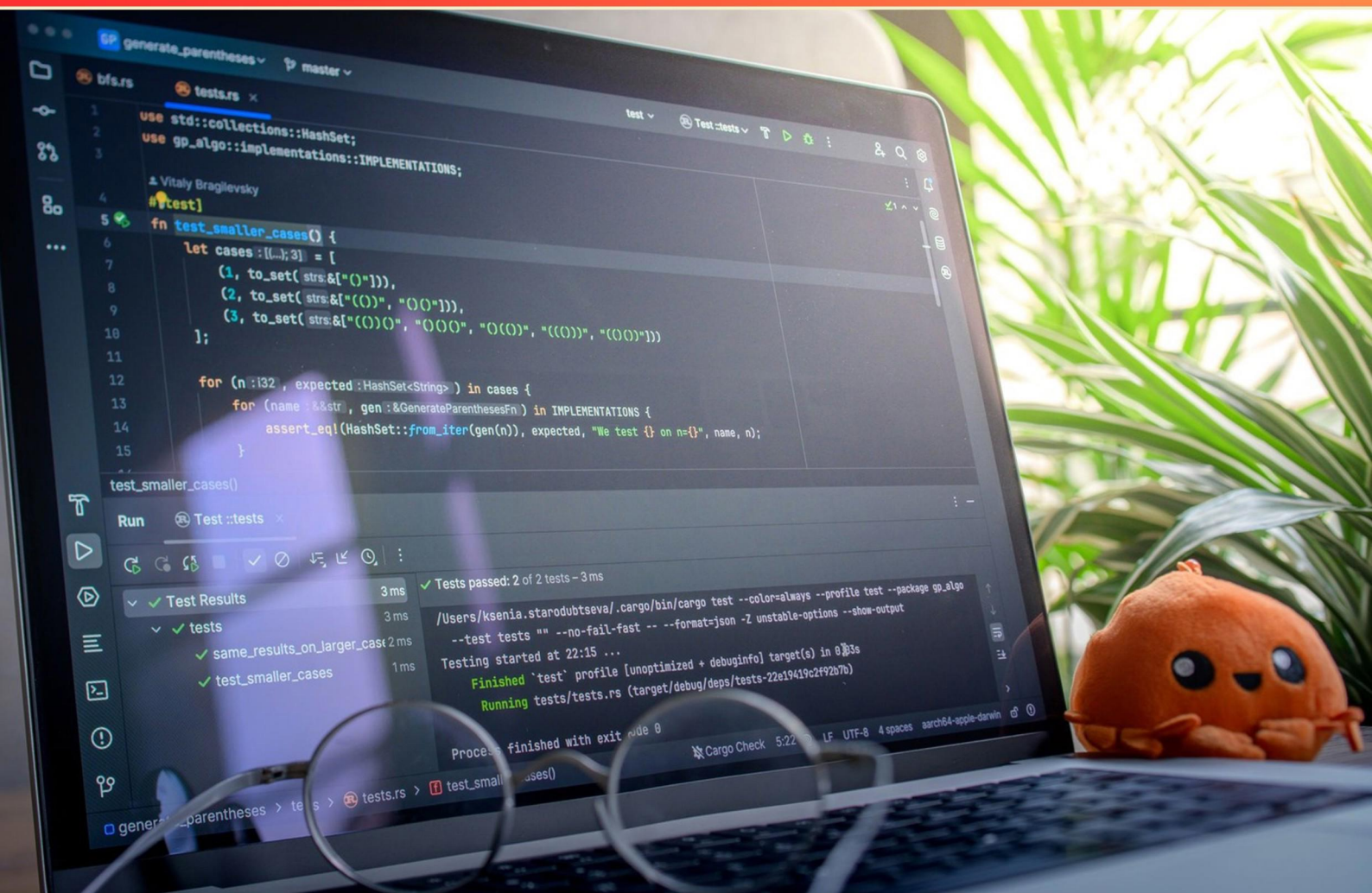


COMPUTER PROGRAMMING (COPROG) PRACTICE TEST (SAMPLE)

STUDY GUIDE



SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

VISIT US ONLINE FOR
THE FULL VERSION STUDY GUIDE

<https://coprog.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. Which primitive data type stores integer values?

- A. Int
- B. Byte
- C. Short
- D. Long

2. In a constant declaration, which part is the data type?

- A. CONSTANT_NAME
- B. dataType
- C. value
- D. access specifier

3. Which of the following is true about String in Java?

- A. It is a primitive type
- B. It is created by default
- C. It is immutable like a primitive
- D. It is a class in java.lang and a non-primitive type

4. What are unary operators?

- A. Operators that require two operands
- B. Operators that need only one operand
- C. Operators that create new values
- D. Operators that require no operands

5. Which operation is an explicit narrowing cast?

- A. `int i = (int) 3.99;`
- B. `double d = i;`
- C. `long l = i;`
- D. `float f = i;`

6. Can the JRE compile Java programs?

- A. Yes.
- B. No.
- C. Only with additional tools.
- D. Only if the JDK is installed.

7. What does void mean?

- A. It indicates that a method returns a value.
- B. It specifies that a method does not return a value.
- C. It indicates that a method is static.
- D. It indicates the method mutates the object.

8. Which part of the syntax assigns the constant's value?

- A. CONSTANT_NAME
- B. value
- C. dataType
- D. access specifier

9. What is $10 / 3$ when both operands are integers?

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

10. Which statement prints text without advancing to a new line?

- A. `System.out.Print()`
- B. `System.out.println()`
- C. `System.out.printl()`
- D. `System.out.print()`

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. Which primitive data type stores integer values?

- A. Int
- B. Byte
- C. Short
- D. Long

Integer values are handled by primitive numeric types that represent whole numbers. Among them, the standard integer type is used for general purposes because it provides a broad range without extra memory overhead, making it the typical choice for arithmetic, counting, and indexing. In many languages, a byte uses 8 bits for very small values, a short uses 16 bits for moderate ranges, and a long uses 64 bits for very large values; the standard integer type provides a balanced default when you don't need to optimize for extreme sizes. So the primitive that stores integer values in everyday use is the standard integer type.

2. In a constant declaration, which part is the data type?

- A. CONSTANT_NAME
- B. dataType
- C. value
- D. access specifier

The data type is the type specifier that tells what kind of value the constant holds. It appears before the constant's name, such as in a declaration like `const int MAX = 100`, where `int` is the data type. The name is the identifier you use to reference the constant, and the value is the literal assigned to initialize it. The access specifier, if present, controls visibility but is not the data type. So the data type is the part that specifies the kind of data stored by the constant.

3. Which of the following is true about String in Java?

- A. It is a primitive type
- B. It is created by default
- C. It is immutable like a primitive
- D. It is a class in java.lang and a non-primitive type

Strings in Java are reference types, not primitive values. They are implemented as objects of the String class, which lives in the java.lang package. Because it's a class, String is a non-primitive type. Strings are immutable: once a String object is created, its character sequence cannot be changed, and operations like concatenation create new String objects. That combination—being a class in java.lang and a non-primitive type—is why this statement is correct. Other options miss or misframe the reality: primitive types are not objects and don't have methods; string literals aren't about being created by default in any special sense; and while Strings are immutable, that property comes from the class design, not from being primitive.

4. What are unary operators?

- A. Operators that require two operands
- B. Operators that need only one operand**
- C. Operators that create new values
- D. Operators that require no operands

Unary operators are the ones that act on a single value. They take one operand and produce a result. Examples include the unary minus that changes the sign of a number ($-x$), the logical NOT that inverts a boolean value ($!x$), and even unary plus or bitwise NOT ($\sim x$). The key idea is how many inputs the operator needs: one input value. This sits in contrast to binary operators, which require two operands (like $a + b$), and to ternary expressions, which involve three operands. The other choices describe what the operator does in terms of producing values or having zero operands, rather than describing how many inputs the operator requires.

5. Which operation is an explicit narrowing cast?

- A. `int i = (int) 3.99;`**
- B. `double d = i;`
- C. `long l = i;`
- D. `float f = i;`

Explicit narrowing cast means you force a value into a type with a smaller range or precision using a cast operator. In the first line, `(int) 3.99` explicitly converts the floating-point literal to an `int`, which trims the fractional part and can lose information. That exact use of `(int)` is what makes it a narrowing cast. The other lines rely on implicit conversions during assignment. Moving from `int` to `double`, `long`, or `float` expands the representable range (a widening conversion) and doesn't use an explicit cast, so they don't demonstrate a narrowing cast.

6. Can the JRE compile Java programs?

- A. Yes.
- B. No.**
- C. Only with additional tools.
- D. Only if the JDK is installed.

The JRE cannot compile Java programs. It is designed to run Java applications, not to develop them. The JRE provides the JVM and the runtime libraries needed to execute already-compiled bytecode. To turn Java source code into bytecode, you need a Java compiler, which is part of the JDK (Java Development Kit). So you write code in `.java` files, use the compiler from the JDK to produce `.class` files, and then run those with the JRE/JVM. If you only have the JRE, you won't have the tools to compile.

7. What does void mean?

- A. It indicates that a method returns a value.
- B. It specifies that a method does not return a value.**
- C. It indicates that a method is static.
- D. It indicates the method mutates the object.

In many languages, the return type of a function or method tells you what value, if any, comes back to the caller. void means no value is returned. The method can still do work, print output, or modify objects, but it doesn't provide a result to the caller. For example, a method declared as void might log a message or update fields without returning anything. This is different from methods that return a value, where the return type would specify something like int, String, or another type. Static status is independent of void—a method can be either static or instance and still be declared as void. Likewise, whether the method mutates the object is not determined by void; a void method may or may not change state.

8. Which part of the syntax assigns the constant's value?

- A. CONSTANT_NAME
- B. value**
- C. dataType
- D. access specifier

The part that actually sets what the constant will hold is the initializer—the value written after the equals sign in the declaration. For example, in a constant declaration like `const int MAX = 100;`, MAX is the name used to refer to the constant, int specifies the type, and 100 is the value assigned to it. The data type only describes the kind of data, not the value, and the access specifier governs who can see or use the constant. So the value on the right-hand side is what determines what the constant stores. If the language requires it, the initializer must be provided to give the constant a concrete value.

9. What is 10 / 3 when both operands are integers?

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

When both operands are integers, the division operation returns the integer quotient, dropping any fractional part. So 10 divided by 3 is 3.333..., and the integer result is 3 (the remainder is 1, ignored in integer division). That's why 3 is the correct answer. The other numbers would reflect different interpretations of the operation, not the integer quotient.

10. Which statement prints text without advancing to a new line?

- A. `System.out.Print()`
- B. `System.out.println()`
- C. `System.out.printl()`
- D. `System.out.print()`**

In Java, printing without moving to a new line is done with the `print` method on the standard output. It writes whatever you pass to it and stays on the same line, so subsequent output continues on the same line. For example, `System.out.print("Hello"); System.out.print(" world");` prints `Hello world` on one line. By contrast, `println` adds a newline after printing, so the next output appears on the next line. A misspelled or incorrectly cased version like `System.out.printl()` or `System.out.Print()` won't compile because the method name must be exactly `print` in lowercase. Therefore, the statement that prints text without advancing to a new line is `System.out.print()`.

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://coprog.examzify.com>

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

SAMPLE