

ARIZONA STATE UNIVERSITY (ASU) CSE110 PRINCIPLES OF PROGRAMMING EXAM 1 PRACTICE (SAMPLE) STUDY GUIDE



SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

VISIT US ONLINE FOR
THE FULL VERSION STUDY GUIDE

<https://asu-cse100exam1.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. What will be the output of the following code snippet?
 - A. The output will be 2424
 - B. The output will be 2425
 - C. The output will be 2528
 - D. The output will be 2524
2. What will happen if the value of 'years' is initially set to 0 in the given loop?
 - A. The loop will execute at least once.
 - B. The loop will never execute.
 - C. There will be a compilation error.
 - D. The balance will increment infinitely.
3. How do you extract the first 5 characters from the string str?
 - A. substring(str, 5)
 - B. substring.str(0, 5)
 - C. str.substring(5)
 - D. str.substring(0, 5)
4. Which keyword is used to define the branch executed when all previous conditions are false?
 - A. if
 - B. else if
 - C. else
 - D. switch
5. What is the result of executing a switch statement without a default case?
 - A. It produces an error.
 - B. Nothing will execute if no cases match.
 - C. It executes cases based only on the first match.
 - D. The program will exit early.

6. What type of error is indicated by the code snippet `System.outt.println("Hello");`?
- A. Run-time error
 - B. Syntax error
 - C. Semantic error
 - D. Logic error
7. In the context of loops, which statement is unlikely to be true?
- A. A for loop is ideal for user input
 - B. A while loop can run indefinitely
 - C. A do-while loop guarantees one execution
 - D. A while loop must have at least one execution
8. What is the output of the statement: `String result = "Hello" + 5;`
`System.out.println(result);`
- A. Hello5
 - B. 5Hello
 - C. Hello 5
 - D. Compile-time error
9. How do programmers typically find exceptions and run-time errors?
- A. By reading the code thoroughly
 - B. By testing the program with a limited set of inputs
 - C. By running the program with a variety of input values
 - D. By using automated tools
10. If a program produces incorrect output but runs without crashing, what type of error is this?
- A. Syntax error
 - B. Logic error
 - C. Run-time error
 - D. Compile error

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. What will be the output of the following code snippet?

A. The output will be 2424

B. The output will be 2425

C. The output will be 2528

D. The output will be 2524

To accurately determine the output of the code snippet and why the correct choice is 2424, it's essential to analyze how the operations within the code execute and what kind of data types are involved. The provided code likely involves arithmetic operations that manipulate integer values. If the code combines two integers in a manner that results in the value 2424, it is possible that it is performing straightforward addition or multiplication of variables, followed by a sequence of other operations that preserve this result. For instance, if there are two variables that are assigned specific integer values and then added or multiplied in such a way that the total equals 2424, that would lead to this particular output. The code might also include some print statements that display the final computed value after all calculations. It is crucial that the variables are properly initialized and the operations are organized correctly to ensure that the output is indeed 2424. Understanding the flow of operations, including the order of arithmetic and any potential loops or conditional statements that might alter the value, is critical in predicting the correct output. Therefore, upon analyzing the specific sequence of operations used, confirming that they total up to 2424 illustrates why this choice is accurate.

2. What will happen if the value of 'years' is initially set to 0 in the given loop?

A. The loop will execute at least once.

B. The loop will never execute.

C. There will be a compilation error.

D. The balance will increment infinitely.

In the context of a loop where 'years' is initialized to 0, the behavior of the loop largely depends on the loop's structure. If the loop conditions rely on 'years' being greater than 0 to execute, setting 'years' to 0 means that the condition is immediately false. Consequently, the loop will not execute at all. For example, consider a while loop structured as `while (years > 0)`. Since the condition evaluates to false right from the start, the body of the loop does not run, and control passes to the next statement following the loop. This is why the correct interpretation is that the loop will never execute. Understanding this concept is essential as it highlights how variable initialization impacts loop execution, which is a fundamental principle in programming.

3. How do you extract the first 5 characters from the string str?

A. `substring(str, 5)`

B. `substring.str(0, 5)`

C. `str.substring(5)`

D. `str.substring(0, 5)`

To extract the first 5 characters from the string `str`, the correct approach uses the method `substring(0, 5)`. This method indicates the start index and the end index of the substring you want to extract. In this case, using `substring(0, 5)` starts at index 0, which is the beginning of the string, and goes up to, but does not include, index 5. Since string indices are zero-based, this captures the characters at indices 0 through 4, which effectively gives you the first five characters of the string. Other options do not correctly follow the syntax or logic for extracting a substring. For instance, specifying an index of 5 would start extraction from the sixth character, meaning you would miss the first five characters entirely. Therefore, using `substring(0, 5)` is the proper syntax and logic for this operation.

4. Which keyword is used to define the branch executed when all previous conditions are false?

- A. if
- B. else if
- C. else**
- D. switch

The keyword used to define the branch that is executed when all previous conditions are false is "else." In programming, the "else" statement is part of a conditional structure that allows the code to handle scenarios when none of the preceding "if" or "else if" conditions apply. When you write a series of conditional statements, you start with an "if" that checks a specific condition. If that condition is false, you might have one or more "else if" statements to check additional conditions. However, if all of those conditions are false, the "else" block will execute, allowing for a default or catch-all behavior in your code. This is particularly useful for providing a fallback outcome when no other conditions are met, thereby ensuring that your program can respond appropriately under various circumstances. The other keywords listed serve different purposes: "if" is used for initial condition checks, "else if" allows you to evaluate further conditions if the preceding "if" is false, and "switch" is typically employed for selecting among several options based on a single expression's value. None of these options fulfil the role that "else" plays in terms of providing a default execution path.

5. What is the result of executing a switch statement without a default case?

- A. It produces an error.
- B. Nothing will execute if no cases match.**
- C. It executes cases based only on the first match.
- D. The program will exit early.

Executing a switch statement without a default case allows the program to function normally, but it means that if none of the specified cases match the expression being evaluated, then no specific actions will be taken for that switch statement. In this situation, the control simply moves to the code following the switch statement, which may lead to no output or actions being performed if there are no further instructions. This behavior is important for understanding how control flow works in programming. The absence of a default case means there is no catch-all action to be taken if no matches are found, which is why nothing will execute if there are no matching cases.

6. What type of error is indicated by the code snippet
`System.outt.println("Hello");`?

- A. Run-time error
- B. Syntax error**
- C. Semantic error
- D. Logic error

The code snippet provided contains a typo: "System.outt.println" instead of "System.out.println." This kind of mistake is classified as a syntax error, which occurs when the code deviates from the rules of the programming language. Syntax errors are detected by the compiler or interpreter before the program is run, preventing it from executing altogether. In this case, the compiler does not recognize "outt" as a valid component of the Java print system, leading to the program failing to compile. Thus, the code snippet will not be executed due to this violation of the language's syntax rules, which underscores the importance of correct spelling and adherence to the structure defined by the programming language. Other types of errors like run-time errors occur during execution under certain conditions, semantic errors relate to logic but do not necessarily break the syntax, and logic errors lead to incorrect behavior despite the code running without crashes. The presence of a syntax error makes it clear that the program cannot compile until the issue is resolved, affirming the correctness of the answer.

7. In the context of loops, which statement is unlikely to be true?

- A. A for loop is ideal for user input
- B. A while loop can run indefinitely
- C. A do-while loop guarantees one execution
- D. A while loop must have at least one execution**

The statement that a while loop must have at least one execution is unlikely to be true, as while loops do not have this requirement. A while loop continues to execute its block of code as long as a specified condition evaluates to true. If that condition is false from the very beginning, the loop will not execute at all. In contrast, a do-while loop is designed to guarantee that its body is executed at least once because the condition is evaluated after the execution of the loop's body. This is fundamentally different from the behavior of a while loop. The fact that a for loop is not typically ideal for user input is also relevant to understanding the structure of these loops, as for loops are generally used for a known number of iterations, while a while loop is more suited for scenarios where the number of iterations is not predetermined. Additionally, while loops can indeed run indefinitely if the condition remains true; hence, they have the potential for infinite execution.

8. What is the output of the statement: `String result = "Hello" + 5;`
`System.out.println(result);`

- A. Hello5**
- B. 5Hello
- C. Hello 5
- D. Compile-time error

The output of the statement is "Hello5" because, in Java, the `+` operator is used for both numerical addition and string concatenation. When one operand is a string, Java converts the other operand to a string as well. In this case, "Hello" is a string and the number `5` is treated as a string during the concatenation operation. Therefore, the two values are combined to form a new string: "Hello" followed directly by "5", resulting in "Hello5". This behavior allows for seamless concatenation between strings and numbers. It's important to remember that if both operands were numbers, Java would perform numerical addition instead. However, since a string is involved here, the operation defaults to concatenation.

9. How do programmers typically find exceptions and run-time errors?

- A. By reading the code thoroughly
- B. By testing the program with a limited set of inputs
- C. By running the program with a variety of input values**
- D. By using automated tools

Programmers typically find exceptions and run-time errors by running the program with a variety of input values. This practice, known as dynamic testing, allows developers to observe how their code behaves in different scenarios and input conditions. By varying the input values, they can uncover edge cases and potential pitfalls that may not have been initially apparent during the coding process. When different scenarios are tested, it becomes easier to identify situations that may lead to exceptions or errors—that is, the program not handling input in the way it was intended or crashing due to unforeseen circumstances. This process is crucial because it helps ensure that the program is robust and can handle unexpected situations gracefully. Other methods, while helpful, do not provide the same level of insight into run-time behavior. For instance, thorough code reading can help improve program logic and structure but may not necessarily reveal how the code behaves under different inputs. Testing with a limited set of inputs may miss cases that expose vulnerabilities or errors. Using automated tools can aid in detecting potential issues, but ultimately, real-world testing with diverse inputs is essential for identifying run-time exceptions effectively.

10. If a program produces incorrect output but runs without crashing, what type of error is this?

- A. Syntax error
- B. Logic error**
- C. Run-time error
- D. Compile error

The scenario described involves a program that runs smoothly but produces incorrect output. This situation indicates the presence of a logic error. A logic error occurs when the program's instructions are syntactically correct and do not cause the program to crash, yet the outcome is not what the programmer intended. The program executes without any interruptions because it follows valid syntax and operational flow, but the underlying logic that dictates how the program's inputs lead to outputs is flawed. For example, if a program is supposed to calculate the sum of two numbers but instead multiplies them due to an incorrect operator being used, it will run without error but produce an incorrect result. Understanding logic errors is crucial in programming, as they often require debugging and careful examination of the program's algorithms and flow to resolve the unintended outcomes.

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://asu-cse100exam1.examzify.com>

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

SAMPLE