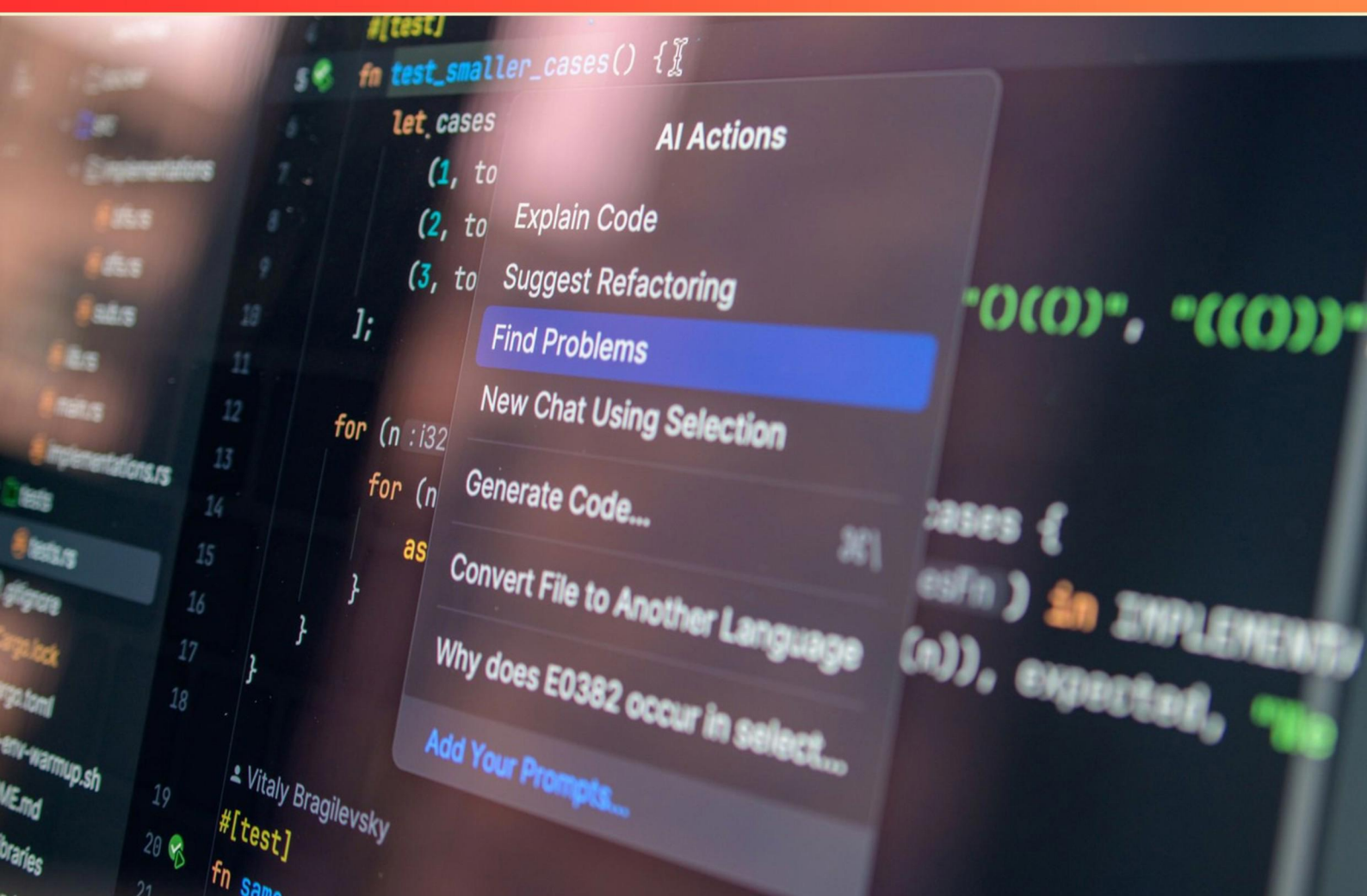


CODEHS AP COMPUTER SCIENCE PRINCIPLES (CSP) PRACTICE EXAM (SAMPLE) STUDY GUIDE



SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

VISIT US ONLINE FOR
THE FULL VERSION STUDY GUIDE

<https://codehsapcsp.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 is the purpose of a library in programming?**
 - A. A runtime environment for executing code.
 - B. A compiler for translating code to machine language.
 - C. A collection of reusable code that provides functions and features to be reused.
 - D. A data store for persisting application state.
- 2. What term describes an incremental or small improvement to an item?**
 - A. Abstraction
 - B. Iteration
 - C. Iterative innovation
 - D. Net neutrality
- 3. Which statement about a library in programming is true?**
 - A. It is the language's compiler.
 - B. It provides a collection of reusable code and features to be reused.
 - C. It directly controls hardware devices.
 - D. It stores data persistently across sessions.
- 4. One Gigabyte is equal to how many Megabytes?**
 - A. 256
 - B. 1024
 - C. 2048
 - D. 512
- 5. What term refers to the gap between people with ready access to computers and the Internet and those without?**
 - A. Digital Divide
 - B. Connectivity Gap
 - C. Technology Gap
 - D. Information Divide

- 6. Which protocol ensures reliable, ordered delivery of a data stream between hosts?**
- A. TCP
 - B. UDP
 - C. IP
 - D. ICMP
- 7. Which practice improves modularity in software design?**
- A. Using cohesive functions with a single responsibility.
 - B. Writing all logic in a single monolithic function.
 - C. Avoiding code reuse.
 - D. Globally shared mutable state.
- 8. Which term best describes a preliminary sketch or model used before building the final product?**
- A. Prototype
 - B. Diagram
 - C. Specification
 - D. Deployment
- 9. Which practice involves finding and fixing problems in your algorithm or program?**
- A. Variable
 - B. Debugging
 - C. Selection
 - D. Boolean
- 10. Which number system uses 16 distinct symbols: 0-9 and A-F?**
- A. Lossless
 - B. RGB
 - C. Image
 - D. Hexadecimal Number System

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. What is the purpose of a library in programming?

- A. A runtime environment for executing code.
- B. A compiler for translating code to machine language.
- C. A collection of reusable code that provides functions and features to be reused.**
- D. A data store for persisting application state.

A library is a collection of reusable code that provides functions and features you can call from your programs. The idea is to save time and reduce mistakes by using prewritten, tested routines instead of writing everything from scratch. Think of a library as a toolbox with a well-defined interface you can use to perform common tasks—math operations, file I/O, networking, or other functionality—without having to reinvent the wheel each time. Libraries are brought into a program by linking or importing, so you can call their functions as part of your code. They aren't the runtime environment where code executes, nor a compiler that translates code, nor a place to store data persistently. They exist to provide ready-to-use features that you can rely on across projects.

2. What term describes an incremental or small improvement to an item?

- A. Abstraction
- B. Iteration
- C. Iterative innovation**
- D. Net neutrality

Incremental improvement happens when you refine something in successive small steps. Iterative innovation captures this idea because each cycle adds a small enhancement to the item, and the process is repeated to accumulate improvements. Abstraction is about hiding details, not improving the item; net neutrality is unrelated policy; an iteration is just repeating a process, while iterative innovation specifically emphasizes ongoing, small upgrades across cycles.

3. Which statement about a library in programming is true?

- A. It is the language's compiler.
- B. It provides a collection of reusable code and features to be reused.**
- C. It directly controls hardware devices.
- D. It stores data persistently across sessions.

A library in programming is a collection of reusable code and features you can call from your program to perform common tasks without writing from scratch. This lets you build software faster and more reliably, since you can reuse well-tested functions, classes, and utilities implemented by others. This idea isn't about the compiler, which is the tool that translates your code into executable instructions. It also isn't about directly controlling hardware devices—libraries may offer APIs that interact with hardware, but their primary role is to provide reusable code, not to manage hardware at a low level. And storing data across sessions isn't something a library inherently guarantees; persistence involves saving data to storage like files or databases, which libraries can help with but do not by themselves ensure.

4. One Gigabyte is equal to how many Megabytes?

- A. 256
- B. 1024**
- C. 2048
- D. 512

Understanding how computer storage scales helps here. In memory sizes, each step up multiplies by 1024, because we use powers of two: a Kilobyte is 1024 bytes, a Megabyte is 1024 Kilobytes, and a Gigabyte is 1024 Megabytes. So a Gigabyte is 2^{30} bytes, and since a Megabyte is 2^{20} bytes, dividing 2^{30} by 2^{20} gives 2^{10} , which is 1024. Therefore, 1 Gigabyte equals 1024 Megabytes. The other numbers would correspond to a quarter of a gigabyte (256 MB), half a gigabyte (512 MB), or two gigabytes (2048 MB), which are not the full amount of one gigabyte.

5. What term refers to the gap between people with ready access to computers and the Internet and those without?

- A. Digital Divide**
- B. Connectivity Gap
- C. Technology Gap
- D. Information Divide

The term for the gap between people who have ready access to computers and the Internet and those who don't is the digital divide. This concept captures how differences in access to devices, broadband, and digital skills create unequal opportunities in education, work, and everyday life. In practice, students without home Internet or a computer may struggle with online homework, applying for jobs, or accessing important services, which widens social and economic disparities. The digital divide is the standard label used in education, policy, and technology discussions to describe this split. The other phrases are less precise: they might hint at related ideas, but they aren't the established term for the broad gap in technology access and capability.

6. Which protocol ensures reliable, ordered delivery of a data stream between hosts?

- A. TCP**
- B. UDP
- C. IP
- D. ICMP

Reliable, ordered data delivery is provided by a connection-oriented transport protocol. TCP establishes a connection, uses sequence numbers for each byte, sends acknowledgments, and automatically retransmits any lost data. This combination guarantees that data arrives intact and in the order it was sent, which is essential for a continuous data stream. It also uses flow control and congestion control to regulate the rate. In contrast, UDP is a connectionless, best-effort protocol with no guarantees about delivery, ordering, or duplicate protection. IP handles routing at the network layer but doesn't guarantee delivery or ordering. ICMP is used for network diagnostics and error reporting, not for delivering application data streams. So the protocol that ensures reliable, ordered delivery is TCP.

7. Which practice improves modularity in software design?

- A. Using cohesive functions with a single responsibility.
- B. Writing all logic in a single monolithic function.**
- C. Avoiding code reuse.
- D. Globally shared mutable state.

Modularity is best achieved when you break a program into small, cohesive units where each piece has a single, well-defined task. Using cohesive functions with a single responsibility fits this idea: each function does one thing, so it's easier to understand, test, and reuse, and you can change one piece without affecting others. A monolithic function that handles many tasks mixes responsibilities and creates tight links between parts of the code, making maintenance and extension difficult. Avoiding code reuse leads to duplicated logic and more places to update, which harms modular design. Globally shared mutable state ties different parts of the program together in ways that can cause changes to have unforeseen effects elsewhere, reducing the clarity of module boundaries. Focusing on cohesive, single-purpose units builds modular and maintainable software.

8. Which term best describes a preliminary sketch or model used before building the final product?

- A. Prototype**
- B. Diagram
- C. Specification
- D. Deployment

This question is about using an early, tangible representation to explore and test ideas before building the final product. That is a prototype. A prototype is created to experiment with how something might work, look, or feel, and to gather feedback so you can refine requirements and design choices before committing to full development. It helps reveal design flaws, clarify user needs, and guide decisions on features, cost, and feasibility in a concrete way. A diagram is a visual plan or flow of how something should function, but it isn't necessarily a working or interactive model. A specification is a detailed description of requirements and constraints for the final product. Deployment is the act of releasing the finished product to users. While diagrams and specifications are important, they don't provide the early, testable model that a prototype offers.

9. Which practice involves finding and fixing problems in your algorithm or program?

- A. Variable
- B. Debugging**
- C. Selection
- D. Boolean

Debugging is the process of finding and fixing problems in your algorithm or program. When code doesn't behave as expected, you identify where things go wrong, inspect the logic, and correct mistakes so the program produces the intended results. This includes using error messages, running test cases, stepping through code, and observing variable values with prints or a debugger. The other terms refer to different ideas: a variable stores data, a selection chooses between different paths in the code, and a boolean represents true/false. Debugging targets the issues that prevent correct behavior, improving the program's reliability and accuracy.

10. Which number system uses 16 distinct symbols: 0-9 and A-F?

A. Lossless

B. RGB

C. Image

D. Hexadecimal Number System

Hexadecimal is the numeral system that uses 16 distinct symbols, 0 through 9 and A through F. Each digit represents a value from 0 to 15, with A=10, B=11, C=12, D=13, E=14, and F=15. Because computers operate on binary, hex is convenient since each hex digit maps to four binary bits, making it easy to read long binary values. For example, the hex number 2A3F corresponds to binary 0010 1010 0011 1111 and equals decimal 10815. The other terms shown aren't number systems: one is a color model used to describe digital colors (RGB), another is a type of image file, and another describes a data compression property.

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://codehsapcsp.examzify.com>

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

SAMPLE