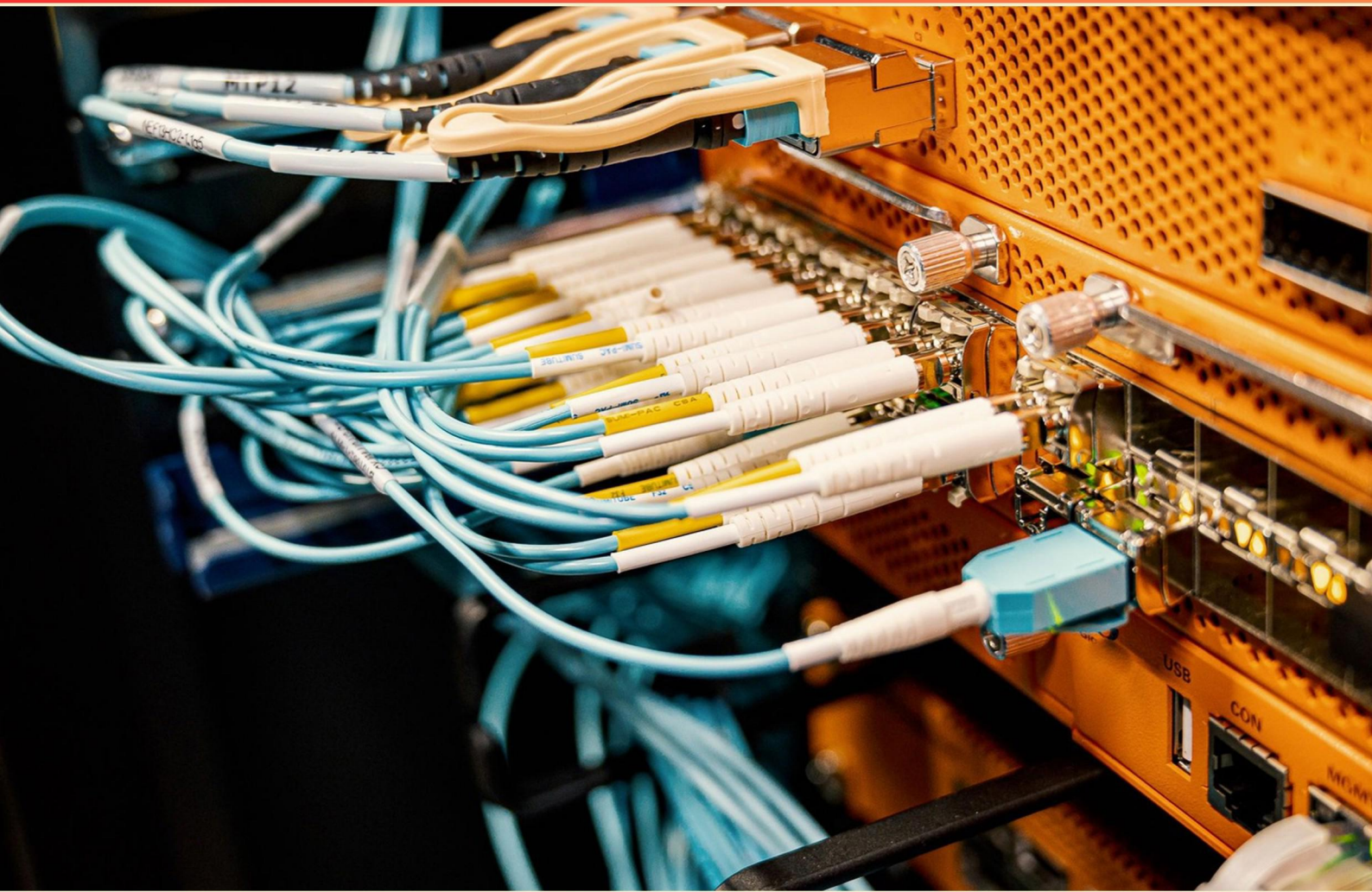


COMPUTER SCIENCE PATHWAY EOPA PRACTICE TEST (SAMPLE)

STUDY GUIDE



**SAMPLE STUDY GUIDE
WITH LIMITED QUESTIONS**

**VISIT US ONLINE FOR
THE FULL VERSION STUDY GUIDE**

<https://cspathwayeopa.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 HTML element is semantic and commonly used to denote the footer of a document or section?**
 - A. Footer
 - B. Header
 - C. Main
 - D. Aside
- 2. Which data type stores a sequence of characters?**
 - A. Array
 - B. Constant
 - C. String
 - D. Field
- 3. Which term describes an object of any type that represents a memory location for storing a value?**
 - A. Variable
 - B. Field
 - C. Pointer
 - D. Register
- 4. Which item is the primary document used to record data collected in a clinical trial?**
 - A. Input Screen Design
 - B. Output Screen Design
 - C. Data Type
 - D. Source Document
- 5. What error refers to a loss of arithmetic precision due to rounding or numerical instability?**
 - A. Truncation error
 - B. Logical error
 - C. Semantic error
 - D. Arithmetic error

- 6. What is written in a program as comments?**
- A. External documentation
 - B. Secondary documentation
 - C. Primary documentation
 - D. Internal documentation
- 7. What term refers to values that do not change and are often defined for reuse in code?**
- A. Variable
 - B. Constant
 - C. Parameter
 - D. Attribute
- 8. What are the three common MFA factors?**
- A. Knowledge, Possession, Inherence
 - B. Knowledge, Exposure, Location
 - C. Possession, Identity, Access
 - D. Biometrics, Tokens, Passwords
- 9. Which type of error occurs when a program attempts to divide by zero?**
- A. Syntax error
 - B. Run time error
 - C. Logic error
 - D. Overflow error
- 10. Differentiate SQL and NoSQL databases and give a typical use case for each.**
- A. SQL databases are non-relational with flexible schemas.
 - B. NoSQL databases use structured schema with SQL queries.
 - C. SQL databases use a structured schema and SQL queries; NoSQL databases are schema-flexible.
 - D. NoSQL always stores data as tables.

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. Which HTML element is semantic and commonly used to denote the footer of a document or section?

- A. Footer**
- B. Header
- C. Main
- D. Aside

Using semantic HTML means naming parts of a page by what they represent. The footer element is the semantic tag used to denote the footer of a document or a section. It typically contains information about the page such as author, publication date, copyright, related links, or small site-wide notes. By signaling a footer, assistive technologies and search engines understand the structure, which improves accessibility and SEO, and it also helps with consistent styling. The other common tags serve different roles: a header marks the top area, main designates the central content, and aside holds content related but separate from the main flow. Wrapping up a page or section with a footer tag communicates a distinct, meaningful region that represents closing or supplementary information.

2. Which data type stores a sequence of characters?

- A. Array
- B. Constant
- C. String**
- D. Field

A string stores a sequence of characters. In programming, strings are the data type designed specifically for text, holding characters in order—think of values like "Hello" or "abc123." They're treated as text data and come with operations such as getting length, concatenating with another string, or extracting substrings. An array can hold a sequence of elements, and a character array is a specific form of that, but the data type used to represent text itself is the string. A constant is simply a value that doesn't change once set, not a container for text. A field is a storage location inside a structure or object, which may hold text, but it's not the data type that represents a sequence of characters. So the string is the right choice for storing a sequence of characters.

3. Which term describes an object of any type that represents a memory location for storing a value?

- A. Variable
- B. Field**
- C. Pointer
- D. Register

A field is a named storage location inside a larger data structure that holds a value of a given type. In many languages, a struct, record, or class is composed of fields, each representing a distinct place in memory where data of a specific type is stored. Because a field is part of a containing object, it has a memory location dedicated to storing that piece of data whenever the object exists. For example, a person object might have fields for name, age, and height, each occupying its own spot inside the object's memory layout. This is what makes a field the right term: it denotes a specific slot within an aggregate that stores a value of its type. By contrast, a variable is a standalone storage location, a pointer holds an address, and a register is CPU-level storage, so they don't fit the description as precisely.

4. Which item is the primary document used to record data collected in a clinical trial?

- A. Input Screen Design
- B. Output Screen Design
- C. Data Type
- D. Source Document**

In a clinical trial, data first exist in source documents—the original records where measurements, observations, and events are documented, such as patient charts, lab reports, or clinician notes. This makes the source document the primary record of data collected, since it preserves the original information before it is transferred into a case report form or an electronic data capture system for analysis. The other options focus on how data are entered or displayed or on data classification, not on where the data originated. Understanding this helps ensure data integrity and traceability, because auditors verify that the data in the study records match what's in the source documents.

5. What error refers to a loss of arithmetic precision due to rounding or numerical instability?

- A. Truncation error
- B. Logical error
- C. Semantic error
- D. Arithmetic error**

Loss of arithmetic precision due to rounding and numerical instability is called arithmetic error. In floating-point computations, numbers can only be stored with finite precision, so many real numbers cannot be represented exactly. Each arithmetic operation may introduce a small rounding error, and when many operations are chained, these errors can accumulate or be amplified, leading to results that differ from the exact mathematical value. This is different from truncation error, which comes from approximating a process by a finite number of terms or steps, such as using a truncated series or a discretized algorithm; truncation error is about the inherent approximation in the method rather than rounding after each operation. Logical error and semantic error relate to the correctness of the algorithm's logic and meaning, not to numeric precision. Therefore, the described error aligns with arithmetic error.

6. What is written in a program as comments?

- A. External documentation
- B. Secondary documentation
- C. Primary documentation
- D. Internal documentation**

Comments written inside a program are internal documentation. They're notes by the programmer that explain what the code does, why a certain approach was taken, or how to reuse and modify it. These comments live in the source file and are ignored by the compiler or interpreter at runtime, so they don't change how the program runs. External documentation resides outside the codebase—like READMEs or API guides—and is intended for users or developers who aren't reading the source directly.

7. What term refers to values that do not change and are often defined for reuse in code?

- A. Variable
- B. Constant**
- C. Parameter
- D. Attribute

Values that stay the same across the program are constants. A constant is defined once and reused wherever needed, and it isn't allowed to be changed later, which helps keep behavior predictable. This makes constants ideal for values like mathematical constants (pi), configuration limits, or other numbers and strings that should remain fixed throughout execution. By using a named constant, you avoid sprinkling the same literal value in many places, making the code easier to read and update if the value needs to change. A variable, by contrast, can be reassigned and changed as the program runs. A parameter is a value passed into a function, serving as input rather than a fixed value. An attribute is a property of an object, describing its state. The concept described here—unchanging values defined for reuse—best matches the idea of a constant.

8. What are the three common MFA factors?

- A. Knowledge, Possession, Inherence**
- B. Knowledge, Exposure, Location
- C. Possession, Identity, Access
- D. Biometrics, Tokens, Passwords

MFA relies on using credentials from distinct kinds of factors to prove who you are. The three common factors are knowledge (something you know, like a password), possession (something you have, such as a hardware token or a phone with an authenticator), and inherence (something you are, such as a fingerprint or other biometric). This combination strengthens security because even if an attacker learns your password, they'd still need the physical token or your biometric to gain access. The other options mix non-factors or mix concepts with the factors. One option lists exposure and location, which aren't authentication factors by themselves. Another combines concepts like identity and access, which are broader security ideas rather than the credential categories. The remaining option lists biometrics, tokens, and passwords, which maps to the same ideas but is just a different way to name the modalities rather than the standard three-factor categories.

9. Which type of error occurs when a program attempts to divide by zero?

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

Dividing by zero triggers a runtime error. The code runs and reaches the division operation, but the divisor is zero, which can't produce a valid numeric result, so the runtime environment raises an exception or halts execution. This is different from a syntax error, which would stop the program from even being parsed or compiled because the code doesn't follow the language's grammar. It's also different from a logic error, where the algorithm runs but produces an incorrect result due to flawed reasoning rather than an illegal operation. Overflow would occur if a computed value exceeded the representable range, not if the divisor is zero. In practice, many languages report a runtime exception (like `ZeroDivisionError` in Python or `ArithmeticException` in Java) when this happens.

10. Differentiate SQL and NoSQL databases and give a typical use case for each.

A. SQL databases are non-relational with flexible schemas.

B. NoSQL databases use structured schema with SQL queries.

C. SQL databases use a structured schema and SQL queries; NoSQL databases are schema-flexible.

D. NoSQL always stores data as tables.

The main idea here is how data organization and querying differ between relational SQL databases and NoSQL systems, and what that means for real-world use. SQL databases use a fixed, structured schema that defines tables, columns, and data types ahead of time, and they query data with SQL. This setup supports strong consistency and powerful relational operations, making it ideal for applications where data integrity and complex joins are essential—think financial systems, inventory or order management, and ERP. NoSQL databases offer schema flexibility and come in several data models (document, key-value, wide-column, graph). They don't require a fixed schema, which lets you store varied or evolving data and often enables easier horizontal scaling for large, diverse datasets—common use cases include content catalogs, real-time analytics, IoT data streams, and social or personalized feeds. So the statement that correctly captures this is that SQL databases use a structured schema and SQL queries, while NoSQL databases are schema-flexible. The other options misstate one or both aspects: SQL isn't non-relational, NoSQL isn't required to use a structured schema with SQL queries, and NoSQL isn't limited to tables.

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://cspathwayeopa.examzify.com>

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

SAMPLE