

IB COMPUTER SCIENCE PRACTICE EXAM (SAMPLE)

STUDY GUIDE



SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

VISIT US ONLINE FOR
THE FULL VERSION STUDY GUIDE

<https://ibcomputerscience.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	16

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 class of laptop computers smaller and with less storage than typical laptops?**
 - A. Notebook
 - B. Ultrabook
 - C. Sub-Laptop
 - D. Chromebook
- 2. Which of the following terms describes a tree node that is the first node along a path?**
 - A. Root
 - B. Child
 - C. Subtree
 - D. Leaf
- 3. Which of the following best describes multitasking in computing?**
 - A. Running a single program at a time
 - B. Running multiple programs sequentially
 - C. Running multiple programs simultaneously on a processor
 - D. Running no programs at all
- 4. What occurs during "acceptance testing"?**
 - A. Performance of a system is monitored during launch
 - B. User requirements of a specification are assessed for fulfillment
 - C. Compatibility issues are identified in software systems
 - D. Baseline performance metrics are established for future reference
- 5. What describes non-regular geometric shapes found in nature that maintain self-similarity?**
 - A. Polygons
 - B. Fractals
 - C. Ellipses
 - D. Symmetries

- 6. What describes a "software-as-a-service" model?**
- A. Software that is installed directly on user machines
 - B. A distribution model with applications hosted by a vendor over a network
 - C. An in-house developed software that is sold to customers
 - D. A service that provides hardware rather than software
- 7. What type of diagram uses geometric figures to represent data flow in a computer system?**
- A. Flowchart
 - B. Block-Arrow Diagram
 - C. Pseudocode Diagram
 - D. Data Structure Diagram
- 8. What does a NOR gate output when both of its inputs are false?**
- A. True
 - B. False
 - C. Invalid
 - D. Neutral
- 9. What is the primary function of a compiler in programming?**
- A. A program that directly executes code
 - B. A program that transforms source code into another language
 - C. An interface for user input
 - D. An emulation of a computer system
- 10. What type of object can a variable represent in programming?**
- A. Words only
 - B. Numbers only
 - C. Any data type, including numbers and text
 - D. Static values only

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. What is the class of laptop computers smaller and with less storage than typical laptops?

- A. Notebook
- B. Ultrabook
- C. Sub-Laptop**
- D. Chromebook

The category of laptop computers that is typically smaller and has less storage than standard laptops is referred to as a sub-laptop. Sub-laptops are designed for portability and often prioritize lightweight and slim designs, focusing on the essential features needed for basic tasks like web browsing, word processing, and email. They are not meant to serve as a full replacement for standard laptops, often sacrificing performance and storage capacity to achieve their compact form. Other types of laptops mentioned, such as notebooks and ultrabooks, share some characteristics but generally fall into different classifications based on their functionality, specifications, and intended use. Notebooks may have similar sizes but can vary significantly in storage and performance. Ultrabooks are a specific class of high-performance, lightweight notebooks but usually still maintain more robust features compared to sub-laptops. Chromebooks, while also often smaller and designed for web-based tasks, are specifically reliant on the Google ecosystem and may not always fit the criteria of being a sub-laptop compared to other classifications.

2. Which of the following terms describes a tree node that is the first node along a path?

- A. Root**
- B. Child
- C. Subtree
- D. Leaf

The term that describes a tree node which is the first node along a path is the "root." In the context of tree data structures, the root is significant because it serves as the starting point from which all other nodes are accessible. It does not have a parent and is the topmost node within the hierarchy of the tree. Understanding the structure of a tree is essential for grasping concepts such as tree traversal and manipulation. The root node is foundational, as it enables navigation to all other nodes through various paths. The hierarchical nature of trees means that every other node can be reached by following a path that originates at the root. In contrast to the root, other terms like "child" and "leaf" refer to different aspects of the tree. A child is a node that is directly connected to another node when moving down the tree, while a leaf is a node that has no children, indicating it is at the end of a path in the tree. A subtree represents a smaller tree structure that includes a node and all its descendants. However, neither of these terms describes the primary, starting node, which is specifically defined as the root.

3. Which of the following best describes multitasking in computing?

- A. Running a single program at a time
- B. Running multiple programs sequentially
- C. Running multiple programs simultaneously on a processor**
- D. Running no programs at all

Multitasking in computing refers to the ability of a processor to execute multiple tasks or programs simultaneously. This capability allows a computer to appear as if it is performing several operations at the same time, enhancing efficiency and user experience. In a multitasking environment, the operating system quickly switches between tasks, allocating processor time to each one, so they seem to run concurrently. This is particularly important for complex applications or when executing background processes while the user is actively interacting with the system. The other options describe different scenarios that do not encapsulate the essence of multitasking. Running a single program at a time indicates a single-task environment. Running multiple programs sequentially suggests that each program waits for the previous one to complete before starting, which is not multitasking but rather a form of time-sharing. Finally, running no programs at all does not involve multitasking, as multitasking inherently requires the execution of at least one program.

4. What occurs during "acceptance testing"?

- A. Performance of a system is monitored during launch
- B. User requirements of a specification are assessed for fulfillment**
- C. Compatibility issues are identified in software systems
- D. Baseline performance metrics are established for future reference

During acceptance testing, the primary focus is on assessing whether the developed system meets the user requirements outlined in the original specifications. The process involves stakeholders, including end-users, who evaluate the system to ensure that it functions as expected and fulfills the intended purpose. This stage is crucial for validating that all user needs and business requirements have been adequately addressed before the system moves to production. Acceptance testing serves as a final verification step, aiming to establish whether the system is ready for deployment. It helps to confirm that the deployment will satisfy the users' expectations and requirements, ultimately minimizing the risk of defects or inadequacies after the system is launched. In contrast, other aspects mentioned in the alternative options focus on different stages of testing or system evaluation, such as performance monitoring during launch or establishing baseline performance metrics, which are not the primary objectives during the acceptance phase.

5. What describes non-regular geometric shapes found in nature that maintain self-similarity?

- A. Polygons
- B. Fractals**
- C. Ellipses
- D. Symmetries

The concept being described in the question relates to the idea of shapes that are non-regular but exhibit self-similarity across different scales. Fractals are a class of geometric shapes that fit this description perfectly. Fractals can be thought of as complex patterns that are built from simple repeating structures, and they are known for their property of self-similarity. This means that if you zoom in on a fractal, you will observe a structure similar to the whole form at different scales. An example of a fractal in nature is the branching of trees, where each branch resembles the overall shape of the tree itself. In contrast, polygons are regular geometric shapes that have straight edges and do not exhibit self-similarity in the same way. Ellipses are smooth, closed curves that also do not have the characteristic repeating patterns of fractals. Symmetries refer to balanced proportions and can be present in various shapes, but they don't specifically describe the self-similar and intricate nature of fractals. Therefore, fractals stand out as the correct answer because they are fundamentally defined by their complex, self-similar patterns that can be found throughout nature.

6. What describes a "software-as-a-service" model?

- A. Software that is installed directly on user machines
- B. A distribution model with applications hosted by a vendor over a network**
- C. An in-house developed software that is sold to customers
- D. A service that provides hardware rather than software

The software-as-a-service (SaaS) model is characterized by applications that are hosted by a third-party vendor and accessed by users over the internet. This model eliminates the need for users to install software directly on their individual machines, as the applications are delivered via the cloud. The vendor is responsible for maintaining the infrastructure, ensuring availability, and managing updates, which allows users to focus on utilizing the software without the complexities of installation or maintenance. Users typically subscribe to these services rather than purchasing them outright, which provides flexibility in terms of cost and scalability. Additionally, since the software is accessible from any device with an internet connection, it enhances collaboration and accessibility for users. This model contrasts sharply with traditional software installation methods where the software resides on local machines and requires users to manage updates and installations independently. In this context, the focus on hosted applications tailored to be accessed remotely by users differentiates the SaaS model significantly from in-house developed solutions or hardware-centric services.

7. What type of diagram uses geometric figures to represent data flow in a computer system?

- A. Flowchart
- B. Block-Arrow Diagram**
- C. Pseudocode Diagram
- D. Data Structure Diagram

The correct choice involves understanding how geometric figures are utilized to represent data flow visually within a computer system. The Block-Arrow Diagram specifically uses blocks and arrows to illustrate the relationships and movement of data between components. The blocks represent processes or data stores, while the arrows show the direction of data flow, making it clear and easy to understand how data is processed within the system. While flowcharts also display processes and data flow, they typically involve a variety of shapes with distinct meanings (like ovals, diamonds, and rectangles) that might not solely focus on geometric representation for data flow. Pseudocode, on the other hand, is textual representation of algorithms and doesn't involve diagrammatic representations with geometric figures. Likewise, Data Structure Diagrams are more concerned with the organization of data rather than the flow of data throughout a system. Thus, the Block-Arrow Diagram distinctly fulfills the requirement of using geometric figures to illustrate data flow.

8. What does a NOR gate output when both of its inputs are false?

- A. True
- B. False**
- C. Invalid
- D. Neutral

A NOR gate is a digital logic gate that outputs true only when all its inputs are false. In the context of Boolean algebra, "true" is typically represented by a value of 1, and "false" is represented by a value of 0. The operation of a NOR gate can be described with the following logic: - If both inputs are false (0, 0), the output of the NOR gate is true (1). - If one or both inputs are true (1), the output is false (0). Therefore, when both inputs of a NOR gate are false, the output is indeed true, confirming that the correct response is false. The understanding of logic gates is crucial as they serve as the foundational building blocks of digital circuits, and NOR gates are particularly significant because they can be used to create any other type of logic gate, making them versatile in circuit design. Consequently, in situations where a NOR gate is evaluated, acknowledging that both inputs being false leads to a true output is pivotal, emphasizing the logical flow of a NOR operation. This reinforces the fundamental principles surrounding digital logic design.

9. What is the primary function of a compiler in programming?

- A. A program that directly executes code
- B. A program that transforms source code into another language**
- C. An interface for user input
- D. An emulation of a computer system

The primary function of a compiler is to transform source code, which is written in a high-level programming language, into another language, typically machine code or an intermediate language. This transformation is essential because computers can only execute code in machine language, which consists of binary instructions that the hardware can process directly. Compilers perform several tasks during this process, including lexical analysis, syntax analysis, semantic analysis, optimization, and code generation. Each step ensures that the code is not only translated correctly but also performs efficiently when executed. This transformation allows developers to write programs in more understandable and manageable languages instead of machine code, which is complex and difficult to work with directly. The output of a compiler can then be run on a specific platform or environment, enabling cross-platform compatibility and software distribution. The other options do not encapsulate the main role of a compiler. For example, directly executing code relates more to interpreters rather than compilers, while user interfaces and emulations serve different functions within computing.

10. What type of object can a variable represent in programming?

- A. Words only
- B. Numbers only
- C. Any data type, including numbers and text**
- D. Static values only

A variable in programming is a fundamental concept that acts as a storage location with a name. It can represent a wide variety of data types, which is why the answer is that a variable can represent any data type, including both numbers and text. Variables are not limited to specific types of data; they can hold integers, floating-point numbers, strings (text), and even more complex data structures like arrays or objects. This flexibility allows programmers to write versatile and dynamic code, where variable types can be determined at runtime in some programming languages, or remain fixed based on the defined type in others. This versatility enables the manipulation of various data types without having to define specific variables for each type, thus improving code efficiency and readability. This understanding is foundational in programming as it allows for the development of robust applications that can handle a diverse set of operations and data.

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://ibcomputerscience.examzify.com>

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

SAMPLE