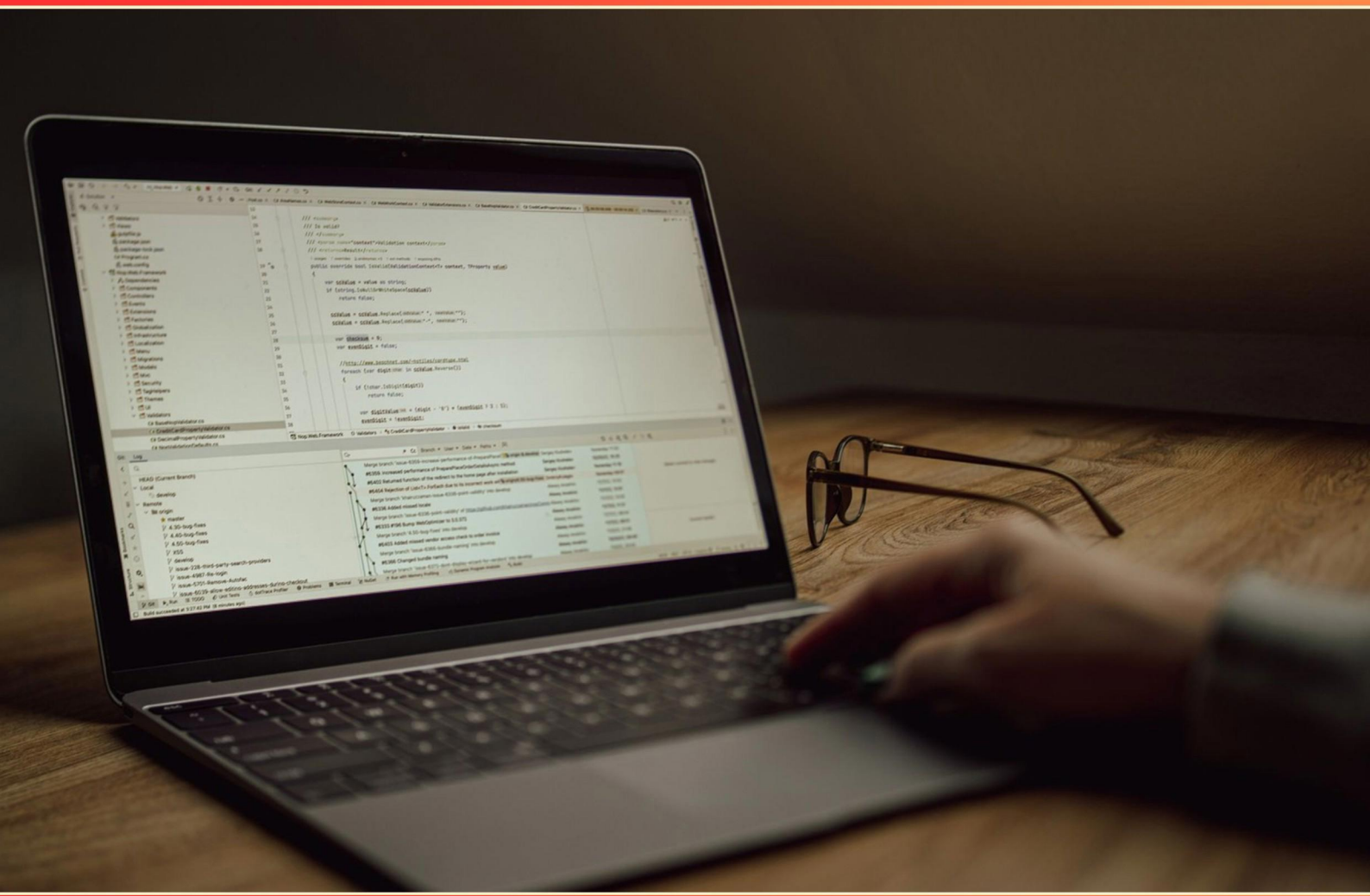
SQA NATIONAL 5 COMPUTING SCIENCE PRACTICE EXAM (SAMPLE)

STUDY GUIDE



SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

VISIT US ONLINE FOR THE FULL VERSION STUDY GUIDE

https://sqanational5computingscience.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.

SAMPLE

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!

SAMPLE

Questions

SAMPLE

1. Which of the following best describes a 'function' in programming?

A. A single-use code block

B. A block of reusable code that performs a specific task

C. A type of data structure

D. A debugging tool

2. Define a variable in programming.

A. A function that performs a specific task

B. A storage location identified by a name that holds data which can change during program execution

C. A method for manipulating data

D. A type of algorithm used for sorting

3. What does 'cloud computing' refer to?

A. The use of local servers for data storage

B. Delivering computing services over the internet

C. A type of network configuration

D. A programming language

4. What is the purpose of an Assembler in programming?

A. To compile high-level languages into machine code

B. To translate assembly language into machine code

C. To debug software applications

D. To store code in a binary format

5. What is the purpose of encryption?

A. To store data in a secure format

B. To allow unauthorized access to information

C. To securely encode data for authorized access only

D. To delete unnecessary data from a database

6. In computing, what does 'virtualization' mean?

A. Enhancing hardware performance

B. Creating a virtual version of resources

C. Storing data remotely

D. Upgrading software applications

SAMPLE

7. Which of the following is an example of a high-level programming language?

A. C

B. Python

C. Assembly

D. Machine Code

8. What distinguishes a runtime error from a syntax error?

A. Runtime errors occur during code execution

B. Syntax errors occur in the design phase

C. Runtime errors are spotted by compilers

D. Syntax errors happen during variable declaration

9. What happens when exception handling is not utilized?

A. The program runs without errors

B. The program is immune to bugs

C. The program may crash unexpectedly

D. The coding process becomes easier

10. Which characteristic distinguishes Assembly language?

A. It is a high-level programming language

B. It directly corresponds to machine code instructions

C. It uses complex syntax for coding

D. It requires extensive libraries

SAMPLE

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. Which of the following best describes a 'function' in programming?

A. A single-use code block

B. A block of reusable code that performs a specific task

C. A type of data structure

D. A debugging tool

A 'function' in programming is best described as a block of reusable code that performs a specific task. Functions allow programmers to encapsulate a sequence of operations under a single name, which can then be invoked whenever needed throughout the program. This not only promotes code reusability but also enhances maintainability and readability, as the same functionality can be accessed with a simple function call instead of writing out the same code multiple times. In contrast, a single-use code block, while it may perform a specific task, does not convey the same reusability that is a hallmark of functions. A type of data structure pertains to the way data is organized, such as arrays or lists, which is distinct from the concept of a function. Lastly, while debugging tools are essential in programming for identifying and fixing errors, they do not fit the definition of what a function fundamentally is. Therefore, the focus on reusability and task performance aligns perfectly with what a function represents, making this the most accurate description.

2. Define a variable in programming.

A. A function that performs a specific task

B. A storage location identified by a name that holds data which can change during program execution

C. A method for manipulating data

D. A type of algorithm used for sorting

A variable in programming is fundamentally a storage location that is assigned a specific name and can hold data which is subject to change as the program runs. This allows programmers to store information that may need to be modified during execution, such as user inputs, computed values, or temporary data needed for processing. The name of the variable acts as a reference to that storage location, enabling the programmer to easily access and manipulate the value contained within. For example, in a program that calculates the total price of items in a shopping cart, a variable could be defined to store the number of items purchased. As items are added or removed from the cart, the value of this variable can be updated, reflecting the current state of the cart throughout the program. This dynamic nature is what distinguishes variables from constants, which hold a fixed value that does not change during execution. The other options refer to different programming concepts. A function is a block of code meant to perform a specific task when called. A method refers to functions that are associated with objects. These do not describe what a variable is. Algorithms are systematic ways to solve problems or perform tasks, such as sorting data, but again, they do not define the concept of a variable.

SAMPLE

3. What does 'cloud computing' refer to?

A. The use of local servers for data storage

B. Delivering computing services over the internet

C. A type of network configuration

D. A programming language

Cloud computing refers to delivering various computing services, including storage, processing power, and applications, over the internet instead of relying solely on local servers or personal devices. This model allows users to access and use resources on demand without the need to invest in extensive hardware or infrastructure. By utilizing cloud services, businesses and individuals can scale their storage and computing capabilities as needed, provide remote access to resources from anywhere with an internet connection, and benefit from cost efficiency. The ability to deploy and manage applications without local hardware constraints is a cornerstone of what makes cloud computing so transformative in today's digital landscape. The other options do not accurately represent cloud computing: using local servers for data storage is contrary to the cloud model, a network configuration refers to the setup of hardware and software in local or other network environments, and a programming language is a tool for writing software, unrelated to the concept of providing computing services over the internet.

4. What is the purpose of an Assembler in programming?

A. To compile high-level languages into machine code

B. To translate assembly language into machine code

C. To debug software applications

D. To store code in a binary format

The purpose of an assembler in programming is to translate assembly language into machine code. Assembly language is a low-level programming language that provides a symbolic representation of a computer's machine code instructions, allowing programmers to write instructions in a more human-readable format compared to binary code. When an assembler processes assembly language code, it converts the mnemonics and symbolic addresses into the binary machine code that a computer's processor can execute. Each instruction in assembly corresponds directly to machine code, so this translation is essential for running the program on hardware. Understanding this function highlights the role of an assembler in the software development process, particularly in systems programming or when performance is crucial, as assembly language often provides more efficient control over hardware compared to high-level programming languages. The other choices do not pertain to the assembler's role; they describe functions meant for compilers, debuggers, and storage formats, which are distinct from the specific task of translating assembly language to machine code.

5. What is the purpose of encryption?

A. To store data in a secure format

B. To allow unauthorized access to information

C. To securely encode data for authorized access only

D. To delete unnecessary data from a database

The purpose of encryption is to securely encode data so that it is only accessible to authorized individuals. By transforming the original data into a format that cannot be understood without a specific key or password, encryption protects sensitive information from unauthorized access. This ensures that even if the data is intercepted or accessed by someone without the proper authorization, they will not be able to interpret or use it. This process is crucial in various scenarios, such as online banking, personal communications, and data storage, where individuals and organizations need to safeguard their private information from potential threats. The effective use of encryption helps maintain confidentiality and integrity, which are key components of information security.

SAMPLE

6. In computing, what does 'virtualization' mean?

A. Enhancing hardware performance

B. Creating a virtual version of resources

C. Storing data remotely

D. Upgrading software applications

Virtualization refers to the creation of a virtual version of resources, such as servers, storage devices, or network resources, rather than using physical hardware. It allows multiple virtual instances to operate on a single physical device, effectively maximizing resource utilization and improving scalability. By utilizing virtualization, organizations can run multiple operating systems and applications on a single server, which can lead to cost savings and efficient management of hardware resources. This capability is key in cloud computing environments and helps in the rapid deployment of new services without needing dedicated physical hardware for each application or task. In contrast, enhancing hardware performance, storing data remotely, and upgrading software applications do not encapsulate the essence of virtualization, as these terms refer to distinct functions and practices in the realm of computing rather than the act of creating virtual representations of resources.

7. Which of the following is an example of a high-level programming language?

A. C

B. Python

C. Assembly

D. Machine Code

High-level programming languages are designed to be easy for humans to read and write, abstracting away much of the complexity of hardware. They allow programmers to focus on writing algorithms without needing to manage hardware details or memory directly. Python is a prime example of a high-level programming language because it uses clear and concise syntax, supports multiple programming paradigms, and comes with a rich set of libraries that simplify complex tasks. This makes it user-friendly and accessible, especially for beginners. In contrast, other options like C are often considered lower-level than Python, as they require more understanding of system operations and memory management, though they are still higher-level than assembly or machine code. Assembly language is a low-level language that is closely tied to machine code and architecture, making it less abstract and harder to read. Machine code consists of binary instructions that a computer's CPU executes directly, making it the lowest level of programming and not human-readable at all.

SAMPLE

8. What distinguishes a runtime error from a syntax error?

A. Runtime errors occur during code execution

B. Syntax errors occur in the design phase

C. Runtime errors are spotted by compilers

D. Syntax errors happen during variable declaration

A runtime error occurs when there is an issue during the execution of the program, meaning the code compiles without any issues, but then behaves unexpectedly when it is run. This can happen due to various reasons such as division by zero, accessing an element outside the bounds of an array, or insufficient memory. The key aspect of runtime errors is that they are detected after the code has been successfully compiled and while it is executing. In contrast, syntax errors are problems with the way the code is written. They are typically detected during the compilation phase, and these errors prevent the code from compiling at all. Syntax errors can include misspelled keywords, missing parentheses, or incorrect punctuation, all of which need to be resolved before the program can run. Understanding this difference is crucial for debugging programs effectively. While runtime errors highlight potential logical or operational issues within the program's flow, syntax errors point toward mistakes in the structure of the code that must be fixed beforehand.

9. What happens when exception handling is not utilized?

A. The program runs without errors

B. The program is immune to bugs

C. The program may crash unexpectedly

D. The coding process becomes easier

When exception handling is not utilized, the program may crash unexpectedly. Exception handling is a crucial aspect of programming that allows a program to manage errors or unexpected situations gracefully. Without it, any error that occurs—whether it's due to invalid input, resource unavailability, or logical flaws—can lead to a total halt in the program's execution. In scenarios where errors occur without proper handling, the program does not have a way to recover from these issues, which can result in crashes or unwanted behavior. Instead of the program providing feedback to the user or taking corrective actions, the application may terminate abruptly, leading to loss of data and a poor user experience. Utilizing exception handling provides structure for capturing errors and defining how the program should respond, thus maintaining stability and reliability under various conditions.

SAMPLE

10. Which characteristic distinguishes Assembly language?

A. It is a high-level programming language

B. It directly corresponds to machine code instructions

C. It uses complex syntax for coding

D. It requires extensive libraries

Assembly language is characterized by its close relationship with machine code, as it directly corresponds to specific machine code instructions of a computer's architecture. Each instruction in an Assembly language program typically maps to a single machine code instruction, which allows for precise control over hardware operations. This characteristic enables programmers to write instructions that the CPU can execute directly without the need for translation into machine code. The other options present misleading definitions. High-level programming languages, for instance, abstract away details of the hardware and are designed to be easier for humans to read and write. Assembly language, conversely, requires a solid understanding of the underlying hardware. Moreover, Assembly language is known for its straightforward syntax rather than complex syntax, which focuses on the use of mnemonic codes instead of intricate symbols or syntax. Lastly, while some programming environments may include libraries, Assembly language is often used in contexts where minimal libraries may be necessary, placing direct emphasis on hardware manipulation rather than on extensive library use.

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://sqanational5computingscience.examzify.com

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

SAMPLE