

INTERNATIONAL SOFTWARE TESTING QUALIFICATIONS BOARD (ISTQB) FOUNDATION LEVEL PRACTICE TEST (SAMPLE) STUDY GUIDE



SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

VISIT US ONLINE FOR
THE FULL VERSION STUDY GUIDE

<https://istqbfoundationlevel.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 key characteristic of exploratory testing?**
 - A. A structured approach based on predefined test cases
 - B. An unstructured method allowing tester creativity
 - C. A technique that strictly follows user requirements
 - D. A formal process adhering to checklists
- 2. What is the main goal of confirmation testing?**
 - A. To assess whether the original issue is fixed
 - B. To confirm no new defects have been introduced
 - C. To validate functionality across different platforms
 - D. To ensure system performance under different conditions
- 3. Which criterion might indicate that a test case is needed for further investigation in a software module?**
 - A. A functionality passes all defined tests
 - B. Condition combinations yield unexpected outputs
 - C. The software runs without crashing
 - D. All requirements are validated
- 4. What is the test basis?**
 - A. The point during software development when testing should start
 - B. The body of knowledge used for test analysis and design
 - C. The source to determine the actual results from a set of tests
 - D. The method used to systematically devise test conditions
- 5. What is a critical requirement for success during a technical review?**
 - A. A = an experienced reviewer
 - B. B = adherence to coding standards
 - C. Timely and adequate preparation
 - D. C = previous review data

- 6. A metric that tracks the number of test cases executed is gathered during which activity?**
- A. Planning
 - B. Implementation
 - C. Execution
 - D. Reporting
- 7. What is error guessing?**
- A. A testing technique used to guess where a developer is likely to have made a mistake
 - B. A technique used for assessing defect metrics
 - C. A development technique to verify that all error paths have been coded
 - D. A planning technique used to anticipate likely schedule variances due to faults
- 8. Which test level is MOST likely being performed if the testing is based on interface specifications and focused on communication failures?**
- A. Component integration testing.
 - B. Acceptance testing.
 - C. System testing.
 - D. Component testing.
- 9. What does component testing primarily focus on?**
- A. Defects in interfaces
 - B. Defects in whole test objects
 - C. Defects in separately testable modules
 - D. Defects not focused on identifying
- 10. Which testing technique would be most effective in determining and improving the maintainability of the code?**
- A. Peer reviews
 - B. Static analysis
 - C. Dynamic testing
 - D. Unit testing

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. What is the key characteristic of exploratory testing?

- A. A structured approach based on predefined test cases
- B. An unstructured method allowing tester creativity**
- C. A technique that strictly follows user requirements
- D. A formal process adhering to checklists

Exploratory testing is defined by its allowance for tester creativity and prioritizes the tester's experience and intuition. This method does not rely on predefined test cases or scripts but encourages testers to explore the application freely, investigate functionalities, and leverage their knowledge to identify defects. As testers navigate through the application, they can adapt their tests based on observations and findings in real-time, which can lead to the discovery of unexpected issues that structured testing might overlook. This adaptability is a hallmark of exploratory testing, focusing on learning and verification simultaneously rather than strictly following a formalized set of steps or requirements.

2. What is the main goal of confirmation testing?

- A. To assess whether the original issue is fixed**
- B. To confirm no new defects have been introduced
- C. To validate functionality across different platforms
- D. To ensure system performance under different conditions

The main goal of confirmation testing, often referred to as re-testing, is to verify that a specific defect or issue that was previously reported has been successfully fixed. When a defect is identified, it undergoes a process where developers implement a fix. Confirmation testing is then carried out to ensure that the specific failure condition no longer exists and that the original issue has been resolved. This process serves to validate the effectiveness of the modification made. In this context, while confirming that no new defects have been introduced is important in overall testing processes, it is not the primary focus of confirmation testing. Similarly, validating functionality across different platforms and ensuring system performance under various conditions pertains to broader testing objectives, rather than specifically assessing the resolution of a previously identified defect. Hence, the emphasis in confirmation testing is strictly on the rectification of the original issue.

3. Which criterion might indicate that a test case is needed for further investigation in a software module?

- A. A functionality passes all defined tests
- B. Condition combinations yield unexpected outputs**
- C. The software runs without crashing
- D. All requirements are validated

The indication that a test case is needed for further investigation in a software module occurs when condition combinations yield unexpected outputs. This scenario suggests that there may be underlying issues in how the software handles different input values, which can lead to bugs or behavior that is not in line with the expected outcomes. Unexpected outputs often signal that the logic in the software is not functioning correctly under certain conditions or combinations of inputs, warranting further testing to uncover and address these discrepancies. In contrast, passing all defined tests, running without crashing, and having all requirements validated suggest that the software is functioning as intended. These scenarios typically indicate that the software meets the specified criteria and does not present any identified issues. Therefore, encountering unexpected behavior during tests highlights a critical area for further scrutiny and testing to ensure the software's reliability and correctness.

4. What is the test basis?

- A. The point during software development when testing should start
- B. The body of knowledge used for test analysis and design**
- C. The source to determine the actual results from a set of tests
- D. The method used to systematically devise test conditions

The test basis refers to the body of knowledge that is utilized for test analysis and design. It encompasses all of the documentation, requirements, specifications, and other sources of information that inform the testing process. This foundational element is critical because it provides the necessary context and criteria against which the software will be tested. A well-defined test basis allows testers to create meaningful test cases that effectively evaluate whether the software meets its requirements and functions as intended. It is important to understand that while other aspects of testing, such as the timing of when to start testing or how to derive expected results, are relevant to the testing process, they do not capture the essence of what the test basis is. Instead, the test basis primarily serves as the reference point for understanding what should be tested and why, facilitating a more structured and effective approach to software testing.

5. What is a critical requirement for success during a technical review?

- A. A = an experienced reviewer
- B. B = adherence to coding standards
- C. Timely and adequate preparation**
- D. C = previous review data

Timely and adequate preparation is crucial for the success of a technical review because it ensures that participants come to the review session well-informed and ready to contribute effectively. Proper preparation allows reviewers to familiarize themselves with the material, identify areas that require attention, and formulate constructive feedback ahead of time. This level of preparedness facilitates a more efficient and productive discussion during the review, as participants can focus on specific issues and collaborate more effectively. While experienced reviewers, adherence to coding standards, and previous review data can support the review process, the primary success factor hinges on the readiness of the participants. When all individuals involved in the review have prepared adequately, it enhances the quality of the feedback and leads to better outcomes. This preparation stage is fundamental in ensuring that the technical review achieves its objective of identifying defects and improving the quality of the work being reviewed.

6. A metric that tracks the number of test cases executed is gathered during which activity?

- A. Planning
- B. Implementation
- C. Execution**
- D. Reporting

The metric that tracks the number of test cases executed pertains specifically to the execution phase of the testing process. During execution, testers run the test cases that have been previously designed and prepared. As they carry out these tests, they collect various metrics, including the number of test cases executed, to gauge progress, assess coverage, and identify any issues that arise during testing. This metric serves as an important indicator of testing activity and can help inform stakeholders about the status of testing and whether it aligns with the planned schedule. Understanding how many test cases have been executed allows teams to manage resources effectively and anticipate the completion of testing as they work towards meeting quality goals.

7. What is error guessing?

- A. A testing technique used to guess where a developer is likely to have made a mistake**
- B. A technique used for assessing defect metrics
- C. A development technique to verify that all error paths have been coded
- D. A planning technique used to anticipate likely schedule variances due to faults

Error guessing is a testing technique that involves making educated guesses about where defects may exist in a system based on the tester's experience and intuition. By understanding common pitfalls and areas where developers are likely to make mistakes, a tester can focus their efforts on those areas, effectively increasing the chances of uncovering defects. This approach leverages the tester's skills and knowledge about the application, the domain, and previous experiences with similar systems. The other options describe different concepts that are not aligned with the essence of error guessing. Assessing defect metrics pertains to the measurement and analysis of defects after they have been identified, which does not involve making proactive guesses about where to find them. Verifying that all error paths have been coded relates more to ensuring complete and correct coding rather than investigating potential errors. A planning technique for anticipating schedule variances focuses on project management rather than on identifying specific areas of potential defects in the software. This reinforces that error guessing is distinctly about anticipating where defects might be present based on insight rather than established metrics or development processes.

8. Which test level is MOST likely being performed if the testing is based on interface specifications and focused on communication failures?

- A. Component integration testing.**
- B. Acceptance testing.
- C. System testing.
- D. Component testing.

The most relevant test level in this scenario is component integration testing. This level of testing specifically focuses on the interactions and interfaces between components or systems. When testing is based on interface specifications and is concerned with communication failures, it indicates that the main aim is to ensure that different modules or components interact correctly and effectively with one another. Component integration testing aims to identify issues that arise when integrated components communicate, as these interactions can often lead to failures that would not be evident when testing the components individually. This level of testing ensures that data is correctly passed between components and that the interfaces meet the specified communication protocols. Other levels of testing, such as acceptance testing, which verifies the software against user requirements, or system testing, which evaluates the complete system's compliance with specified requirements, do not focus specifically on the interfaces or communication between components. Component testing, on the other hand, involves testing individual components in isolation rather than their interactions, making it less relevant to the context described in the question. Thus, component integration testing is the most appropriate choice in this situation.

9. What does component testing primarily focus on?

- A. Defects in interfaces
- B. Defects in whole test objects
- C. Defects in separately testable modules**
- D. Defects not focused on identifying

Component testing primarily focuses on defects in separately testable modules or components of a software system. In this testing approach, individual parts of the software—such as functions, classes, or modules—are examined in isolation from the rest of the application. The purpose of this testing is to identify and address any issues within these specific components before they are integrated into larger systems. By concentrating on individual modules, testers can ensure that each component operates correctly and meets its specifications, leading to a more reliable overall system. This focus on isolating and testing components allows for more precise debugging and higher-quality software, as defects can be identified and resolved at an early stage in the development cycle. Therefore, the emphasis on testing modules separately is what sets component testing apart from other types of testing that might look at the system as a whole or at the interactions between various components.

10. Which testing technique would be most effective in determining and improving the maintainability of the code?

- A. Peer reviews
- B. Static analysis**
- C. Dynamic testing
- D. Unit testing

Selecting static analysis as the most effective technique for determining and improving the maintainability of code is appropriate because static analysis involves examining the code without executing it. This method can identify various issues, such as coding standards violations, complexity metrics, and potential bugs, all of which contribute to maintainability. By utilizing static analysis tools, developers can get insights into code structure, readability, and adherence to best practices - factors that significantly impact how easy it is to maintain the code over time. These tools can highlight areas of the code that may be overly complex or poorly structured, enabling developers to refactor and improve the codebase proactively. In contrast, peer reviews can provide valuable feedback on code quality and maintainability but are often subjective and depend on the skills and perspectives of the reviewers. Dynamic testing, while useful for assessing functionality, does not directly address maintainability issues as it focuses on the behavior of the code during execution. Unit testing is crucial for verifying individual components but doesn't comprehensively evaluate issues that could affect maintainability across the complete system. Thus, static analysis stands out as the most effective technique specifically targeting the maintainability of the code.

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://istqbfoundationlevel.examzify.com>

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

SAMPLE