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 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. What does high severity indicate about a defect?

A. It must be resolved immediately to maintain release schedule

B. It poses a critical risk to system functionality

C. It impacts only special scenarios

D. It requires a complex workaround

2. Which of the following lists contains only typical exit criteria from testing?

A. Reliability measures, test coverage, test cost, schedule, and status about fixing errors

B. Reliability measures, test coverage, degree of tester's independence, and product completeness

C. Reliability measures, test coverage, test cost, availability of test environment, and product completeness

D. Time to market, remaining defects, tester qualification, and test coverage

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. Using the decision table technique, what is the minimum number of test cases needed based on provided conditions and results?

A. 7

B. 13

C. 15

D. 18

5. What is a task that can be carried out as part of the test process?

A. Analyzing a defect

B. Designing test data

C. Assigning a version to a test item

D. Writing a user story

SAMPLE

6. **Which of the following is a correct match of test work products with their descriptions?**

A. Test suite - A group of test scripts with a sequence of instructions.

B. Test case - A set of preconditions, inputs, actions, expected results, and post conditions.

C. Test script - A sequence of instructions for the execution of a test.

D. Test charter - A statement of test objectives with suggestions on how to test.

7. **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

8. **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

9. **How does maintaining traceability between the test basis and test artifacts add value?**

A. Maintenance testing can be fully automated based on changes to the initial requirements.

B. It is possible to determine if a new test case has increased coverage of the requirements.

C. Test managers can identify which testers found the highest severity defects.

D. Areas that may be impacted by side-effects of a change can be targeted by confirmation testing.

10. **In an Agile environment, which benefit of static testing directly applies to increased team velocity?**

A. Increasing total cost of quality

B. Reducing testing cost

C. Increasing development productivity

D. Reducing total cost of quality

SAMPLE

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. What does high severity indicate about a defect?

A. It must be resolved immediately to maintain release schedule

B. It poses a critical risk to system functionality

C. It impacts only special scenarios

D. It requires a complex workaround

High severity indicates that a defect poses a critical risk to system functionality. When a defect is categorized as high severity, it means that it significantly affects the system's ability to perform its intended functions accurately and reliably. This type of defect could potentially lead to system crashes, data corruption, or other critical failures that prevent users from using the system effectively. In software testing, severity reflects the impact of a defect on the application's operation. A defect that undermines core functionalities or leads to system anomalies is deemed high severity because it threatens the overall performance and usability of the software. Addressing such defects is crucial to ensure the software meets quality standards and satisfies user requirements, as they can directly affect user experience and system integrity.

2. Which of the following lists contains only typical exit criteria from testing?

A. Reliability measures, test coverage, test cost, schedule, and status about fixing errors

B. Reliability measures, test coverage, degree of tester's independence, and product completeness

C. Reliability measures, test coverage, test cost, availability of test environment, and product completeness

D. Time to market, remaining defects, tester qualification, and test coverage

The correct choice includes typical exit criteria from testing that focus on measurable and tangible aspects of the testing process. Exit criteria are the conditions that must be met before testing can be considered complete, and they are critical for assessing whether a software product is ready for release. In this case, the reference to reliability measures indicates the stability and performance expectations that need to be verified before concluding testing. Test coverage is also crucial, as it tells us how much of the software has been tested against planned test cases and whether the critical functionalities have been adequately verified. The inclusion of test cost and status about fixing errors adds practical considerations that reflect the overall efficiency of the testing process and the resolution of identified issues. Other options may incorporate elements relevant to the testing process, but they do not represent the standard exit criteria as comprehensively as the correct option. For instance, elements such as degree of tester's independence or tester qualification do not directly relate to product readiness, rather they reflect the testing environment or personnel aspects. Time to market focuses on deadlines rather than concrete testing outcomes, and remaining defects, while important, do not provide a complete view of whether testing has met all necessary criteria for exit.

SAMPLE

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. Using the decision table technique, what is the minimum number of test cases needed based on provided conditions and results?

A. 7

B. 13

C. 15

D. 18

The decision table technique is a systematic approach used in software testing to explore all possible combinations of conditions and corresponding actions or outcomes. It is particularly useful when dealing with complex logical relationships between different inputs and the system's responses. To determine the minimum number of test cases needed using this technique, one must first break down the various conditions and the potential outcomes associated with them. Generally, the number of test cases can be calculated using the expression \(2^n\), where \(n\) is the number of conditions, assuming that each condition has two possible states (true or false). However, in many situations, conditions can have more than two possible states or outputs, which would require additional considerations. For instance, if you have conditions with varying states, you might end up needing more combinations than just the binary setup, thus increasing the number of required test cases. In this scenario, if we arrive at the answer of 15 by performing the calculations correctly based on the number of conditions and decisions within the provided context, it indicates that we may have conditions with more complexity than binary choices, or there are additional combinations that must be considered in order to cover all scenarios adequately. Considering decision tables as a structured method ensures that all logic paths are accounted for,

SAMPLE

5. What is a task that can be carried out as part of the test process?

A. Analyzing a defect

B. Designing test data

C. Assigning a version to a test item

D. Writing a user story

Designing test data is a crucial task within the test process because it directly impacts the quality and effectiveness of the testing effort. Test data is needed to execute test cases and verify that the software behaves as expected under various conditions. By creating appropriate test data, testers can simulate real-world scenarios, ensuring that all features and functionalities are rigorously evaluated. Designing effective test data involves understanding the application's requirements and how data flows through the system, which is essential for comprehensive testing. In contrast, while analyzing a defect is an important part of the overall testing lifecycle, it typically occurs after defects are identified and does not fall strictly under the core test process tasks. Assigning a version to a test item may be relevant for test management but is more administrative in nature rather than a direct testing activity. Writing a user story, on the other hand, is primarily a requirement-gathering activity related to Agile methodologies, focusing on the needs of users, rather than a specific task within the test process.

6. Which of the following is a correct match of test work products with their descriptions?

A. Test suite - A group of test scripts with a sequence of instructions.

B. Test case - A set of preconditions, inputs, actions, expected results, and post conditions.

C. Test script - A sequence of instructions for the execution of a test.

D. Test charter - A statement of test objectives with suggestions on how to test.

The correct match for the description provided in the question is the test script, which is accurately defined as a sequence of instructions for the execution of a test. A test script outlines the specific steps that need to be followed to carry out a test, including what actions to perform, the inputs to use, and the expected outcomes. This allows testers to consistently replicate tests and gather reliable results, which is essential in a structured testing process. The test suite, while related, refers to a collection of test cases or test scripts, potentially organized for a particular testing purpose. It is not limited to simply a group of test scripts with a sequence of instructions but is broader in scope. The test case is another foundational element of testing, as it includes not just the execution steps but also defines preconditions, inputs, actions, expected results, and postconditions. This definition makes it comprehensive for validating functionality. Lastly, the test charter outlines the objectives of testing and provides guidance on how to conduct tests but does not serve as a sequence of instructions itself, making it distinct from the other descriptions. Therefore, the correct match for the definitions provided in the question points to the test script as it concisely encapsulates the essence of what a test script is meant to

SAMPLE

7. 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.

8. 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.

SAMPLE

9. How does maintaining traceability between the test basis and test artifacts add value?

A. Maintenance testing can be fully automated based on changes to the initial requirements.

B. It is possible to determine if a new test case has increased coverage of the requirements.

C. Test managers can identify which testers found the highest severity defects.

D. Areas that may be impacted by side-effects of a change can be targeted by confirmation testing.

Maintaining traceability between the test basis (such as requirements, design documents, etc.) and test artifacts (like test cases, test scripts, etc.) is vital for ensuring that all requirements are adequately tested. By establishing this clear connection, it becomes possible to assess whether a new test case enhances the coverage of the requirements. This connection not only helps in validating that every requirement is being addressed but also aids in identifying gaps in testing where certain requirements may not have corresponding test cases. When a new requirement is introduced or an existing requirement is modified, traceability enables testers to quickly update or create necessary test cases. This ensures that the test suite remains aligned with the project's objectives and that any risks associated with incomplete testing are minimized. Consequently, test coverage can be quantified and improved, leading to a higher level of confidence in the software's functionality and reliability. This approach ultimately contributes to the quality assurance process, as it allows stakeholders to verify that the software being developed aligns with the originally intended requirements, thus enhancing the overall effectiveness of testing efforts.

10. In an Agile environment, which benefit of static testing directly applies to increased team velocity?

A. Increasing total cost of quality

B. Reducing testing cost

C. Increasing development productivity

D. Reducing total cost of quality

The benefit of static testing that directly applies to increased team velocity is increasing development productivity. In an Agile environment, teams focus on delivering working software quickly and efficiently. Static testing techniques, such as reviews and inspections, help identify defects early in the development process before the code is executed. This early detection allows developers to correct issues at a lower cost and with less effort, which can lead to faster turnaround times for feature development. By catching defects early and ensuring that the code meets the necessary quality standards before it progresses further in the development cycle, teams can avoid the time-consuming process of fixing issues later in the workflow. This increased level of productivity means that the team can focus more on developing new features and less on rework, ultimately enhancing the overall team velocity in delivering value to stakeholders. In contrast, while other options may touch on relevant concepts, they do not directly contribute to increasing team velocity as effectively as enhanced development productivity. Reducing testing cost and total cost of quality focus on economic aspects rather than the efficiency of development processes, while increasing total cost of quality does not align with the goal of improving team velocity.

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://istqbfoundationlevel.examzify.com

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

SAMPLE