

Software Development Engineer in Test (SDET) Interview Practice Test (Sample)

Study Guide



Everything you need from our exam experts!

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

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

- 1. Which of the following is NOT a part of White Box Testing?**
 - A. Testing the flow of specific inputs through the code**
 - B. Testing for performance issues**
 - C. Checking for internal security holes**
 - D. Testing of each statement and function**
- 2. What action should be taken if a defect is identified while testing?**
 - A. Ignore the defect**
 - B. Create a defect report and continue manual testing**
 - C. Immediately retest without modifications**
 - D. Permanently stop all automation efforts**
- 3. What is the main purpose of using Git in development?**
 - A. To document code changes**
 - B. To manage database connections**
 - C. To maintain version control**
 - D. To automate deployments**
- 4. In a Data Driven Framework, where does the test data typically come from?**
 - A. Hardcoded values**
 - B. Database queries**
 - C. External files like Excel**
 - D. Web services**
- 5. What is a key aspect of Integration Testing?**
 - A. Testing individual components in isolation**
 - B. Testing the system as a whole**
 - C. Testing the interaction of different parts of the system**
 - D. Testing for acceptance criteria**
- 6. What is a key feature of the List Interface in Java?**
 - A. Allows for only unique elements**
 - B. Preserves the order of elements**
 - C. Enforces type safety**
 - D. Does not allow null values**

- 7. How do you handle JavaScript alerts in Selenium?**
- A. Using WebDriver's switchTo method**
 - B. Using JavaScriptExecutor**
 - C. Using Alert class methods**
 - D. Using pageObject methods**
- 8. What command is used to convert a Maven project to an Eclipse project?**
- A. mvn eclipse:build**
 - B. mvn eclipse:convert**
 - C. mvn eclipse:eclipse**
 - D. mvn eclipse:setup**
- 9. Which of the following statements describes the purpose of Selenium in relation to web applications?**
- A. To develop web applications**
 - B. To test web applications by simulating user interactions**
 - C. To build backend servers**
 - D. To create content management systems**
- 10. What type of alert prompts the user to fill out required information?**
- A. Simple Alert**
 - B. Confirmation Alert**
 - C. Prompt Alert**
 - D. Warning Alert**

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. Which of the following is NOT a part of White Box Testing?

- A. Testing the flow of specific inputs through the code**
- B. Testing for performance issues**
- C. Checking for internal security holes**
- D. Testing of each statement and function**

White Box Testing focuses on the internal workings of an application, allowing testers to access the source code and identify issues by examining the implementation. The primary goal is to verify the flow of logic, validate the correctness of the code, and ensure that all functions and statements are adequately tested. Testing the flow of specific inputs through the code is a core aspect of White Box Testing. Testers design tests based on knowledge of the code structure to determine how inputs are processed and to ensure that the expected outputs are produced. Similarly, checking for internal security holes is relevant, as it involves examining the code for vulnerabilities that could be exploited. Additionally, the testing of each statement and function aligns with the goal of making sure that every aspect of the code is executed and validated. In contrast, testing for performance issues typically falls under the domain of performance testing or non-functional testing, rather than White Box Testing. While performance can sometimes be influenced by the internal implementation, the focus in White Box Testing is primarily on validation of logic and correctness rather than measuring performance metrics. Therefore, selecting testing for performance issues as something that is not part of White Box Testing accurately reflects the distinction between the objectives of White Box Testing and other testing methodologies.

2. What action should be taken if a defect is identified while testing?

- A. Ignore the defect**
- B. Create a defect report and continue manual testing**
- C. Immediately retest without modifications**
- D. Permanently stop all automation efforts**

Creating a defect report and continuing manual testing is the appropriate action when a defect is identified during testing. This approach ensures that the defect is formally logged, allowing developers and other stakeholders to track its status and prioritize its resolution. By documenting the issue, including steps to reproduce it and any relevant screenshots or logs, the quality team provides crucial context that can help in fixing the problem effectively. Continuing manual testing while the defect is being addressed allows the testing process to proceed without unnecessary delays. This ensures that other areas of the application can still be evaluated and additional defects may be identified, which is critical for maintaining overall test coverage and quality. This systematic handling of defects also promotes efficient development cycles and improves the overall product quality. In contrast, ignoring the defect would lead to unresolved issues in the software, negatively impacting user experience. Immediately retesting without modifications lacks a foundation for understanding whether the defect has been addressed properly. Halting all automation efforts would disrupt the development workflow and does not focus on resolving the defect efficiently.

3. What is the main purpose of using Git in development?

- A. To document code changes
- B. To manage database connections
- C. To maintain version control**
- D. To automate deployments

Using Git is primarily focused on maintaining version control, which is essential in software development. Version control systems like Git allow developers to track and manage changes to code over time. This feature enables teams to collaborate effectively, as multiple developers can work on the same project concurrently without overwriting each other's changes. Git keeps a history of changes, allowing developers to revert to previous versions if necessary, compare changes, and understand the evolution of the codebase. Maintaining version control also facilitates branching and merging, which is crucial for developing new features or fixing bugs in isolation before integrating them back into the main codebase. This practice helps in ensuring code stability and reliability, making it easier to manage complex projects and coordinate efforts among team members. Other options, while they may be associated with software development practices, do not capture the primary purpose of Git. Documenting code changes is a byproduct of version control rather than its main purpose. Managing database connections and automating deployments are responsibilities handled by different tools and systems, distinct from the core functionalities offered by Git.

4. In a Data Driven Framework, where does the test data typically come from?

- A. Hardcoded values
- B. Database queries
- C. External files like Excel**
- D. Web services

In a Data Driven Framework, the test data is most commonly sourced from external files like Excel. This approach allows for dynamic testing where data can be easily updated, reused, and maintained outside of the code. The external files can contain numerous sets of data that can be iterated over during test execution, enabling testers to validate the application with various inputs without modifying the test scripts. This method enhances flexibility and scalability, ensuring that tests can cover a wide range of scenarios and data variations efficiently. Using external files also promotes better organization and separation of concerns, as test logic remains independent of test data, making it easier to collaborate on test cases and adjust input data as needed without any code changes. Additionally, data stored in formats like Excel is often more user-friendly for stakeholders, allowing non-technical team members to view or modify test data easily.

5. What is a key aspect of Integration Testing?

- A. Testing individual components in isolation
- B. Testing the system as a whole
- C. Testing the interaction of different parts of the system**
- D. Testing for acceptance criteria

A key aspect of Integration Testing is focused on testing the interaction of different parts of the system. The primary goal of this testing phase is to identify issues that may arise when individual components or modules integrate and work together. While unit testing emphasizes the functionality of isolated components, Integration Testing is concerned with the collaboration and communication between those components to ensure they operate seamlessly when combined. This type of testing helps uncover defects in the interfaces and interactions between integrated components, addressing scenarios that may not be evident when components are tested in isolation. By validating these interactions, Integration Testing contributes to ensuring that the entire system functions as intended when all parts are assembled.

6. What is a key feature of the List Interface in Java?

- A. Allows for only unique elements
- B. Preserves the order of elements**
- C. Enforces type safety
- D. Does not allow null values

The List interface in Java is designed to maintain the order of the elements as they are inserted. This means that when you retrieve elements from a List, they will appear in the same sequence in which they were added. This characteristic is fundamental for scenarios where the order of data is important, such as when implementing a playlist, a queue, or any scenario that requires ordered data processing. While type safety and uniqueness of elements are features that might apply to other collections (like sets), the primary defining characteristic of a List is its ability to preserve the order of elements. Additionally, unlike some other collections, Lists can contain duplicates and allow null values, further setting them apart in terms of functionality.

7. How do you handle JavaScript alerts in Selenium?

- A. Using WebDriver's switchTo method
- B. Using JavaScriptExecutor
- C. Using Alert class methods**
- D. Using pageObject methods

Handling JavaScript alerts in Selenium is primarily done through the Alert class methods. This class provides a straightforward interface to interact with alert pop-ups generated by JavaScript in the browser. When a JavaScript alert appears, typical actions include accepting the alert, dismissing it, or retrieving the text displayed in the alert. The Alert class methods such as ``accept()`, `dismiss()`, `getText()`, and `sendKeys()` allow SDET professionals to perform these actions seamlessly. Using the Alert class is essential because it specifically handles the modal dialog behavior that JavaScript alerts exhibit, providing developers with precise control over these components. If a user tries to interact with any other part of the web page while the alert is visible, it would lead to exceptions or errors, highlighting the need for a dedicated approach provided by the Alert class. While the other options present valid approaches for different tasks within Selenium, they do not focus on the specific needs for handling JavaScript alerts. The `switchTo` method allows switching between different windows or frames, `JavaScriptExecutor` is for executing browser-specific JavaScript commands directly, and page object methods typically serve to encapsulate page interactions rather than addressing JavaScript alerts directly. Thus, the Alert class is the most appropriate`

8. What command is used to convert a Maven project to an Eclipse project?

- A. `mvn eclipse:build`
- B. `mvn eclipse:convert`
- C. `mvn eclipse:eclipse`**
- D. `mvn eclipse:setup`

The command used to convert a Maven project into an Eclipse project is `"mvn eclipse:eclipse."` This command generates the necessary Eclipse project files and classpath settings that allow Eclipse to recognize and work with the Maven project. When executed, it creates a ``.project`` file and a ``.classpath`` file in the root directory of the Maven project. This setup makes it easy for developers to import the project directly into the Eclipse IDE without having to configure it manually. Additionally, this command supports the integration of Maven dependencies into the Eclipse workspace, ensuring that all required libraries are correctly linked. The other options provided either do not exist as valid Maven commands or are incorrectly named, which is why they would not successfully convert a Maven project to an Eclipse project.

9. Which of the following statements describes the purpose of Selenium in relation to web applications?

- A. To develop web applications**
- B. To test web applications by simulating user interactions**
- C. To build backend servers**
- D. To create content management systems**

The purpose of Selenium is to test web applications by simulating user interactions. Selenium is a powerful tool primarily used for automating tests in web applications. It allows testers to write scripts in various programming languages to interact with the web elements, such as buttons, forms, and links, just as a user would. This capability enables the testing of user scenarios and ensures that the web application behaves as expected under various conditions. By automating these interactions, Selenium helps identify bugs, verify functionalities, and ensure user experience quality, making it an essential tool for quality assurance in web development. Testing through Selenium is integral to the software development lifecycle, supporting continuous integration and delivery practices by providing quick feedback on code changes. In contrast, other options like developing web applications, building backend servers, or creating content management systems do not align with Selenium's capabilities, as these activities pertain to software development rather than testing.

10. What type of alert prompts the user to fill out required information?

- A. Simple Alert**
- B. Confirmation Alert**
- C. Prompt Alert**
- D. Warning Alert**

The correct choice is Prompt Alert because this type of alert is specifically designed to request input from the user. It typically presents a message along with an input field, enabling users to enter the required information directly. This interaction is crucial in scenarios where the application needs user input for continuation, such as asking for a username or password. In contrast, a Simple Alert displays a message to the user but does not request any input. Its function is primarily to inform the user rather than engage them interactively. A Confirmation Alert offers options to the user, like "OK" or "Cancel," typically used for confirming actions rather than collecting user input. Meanwhile, a Warning Alert conveys caution or informs about potential issues but similarly does not provide an input mechanism—it's meant for alerting rather than interaction.

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://sdetinterview.examzify.com>

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