SOFTWARE DEVELOPMENT ENGINEER IN TEST (SDET) INTERVIEW PRACTICE TEST (SAMPLE)

STUDY GUIDE

SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

VISIT US ONLINE FOR THE FULL VERSION STUDY GUIDE

https://sdetinterview.examzify.com

Table of Contents

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. What does a job in Jenkins represent?

A. A set of triggers for when the code runs

B. A task performed by Jenkins based on a schedule

C. A method for deploying code to production

D. A specific code repository used in projects

2. Which of the following best describes non-functional testing?

A. Testing the UI/UX of the application

B. Testing the functionality with valid inputs

C. Testing performance and security of the application

D. Testing broken paths in the code

3. In what scenario would you use a Prompt Alert?

A. When you want to display information to the user

B. When you need user input for a task

C. When confirming user action

D. When needing to inform about a successful operation

4. How can parameters be passed to a scenario in Cucumber?

A. They are defined in the feature file as variables

B. They must be hardcoded within the steps

C. They are included in the Background section

D. They cannot be passed

5. Which feature of Selenium allows for waiting until a specific condition is met?

A. Fluent Wait

B. Implicit Wait

C. Sleep Function

D. Inactivity Wait

6. What condition typically causes a NullPointerException in Java?

A. Trying to access a null reference of an object

B. Accessing an array with an out-of-bounds index

C. Calling a method on an uninitialized variable

D. Assigning a value to a final variable

7. In the context of Selenium, how would you handle a hidden element?

A. Use CSS selectors

B. Use driver.findElement()

C. Use JavaScript to click the element

D. Use wait functions

8. 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

9. What is the primary role of a constructor in a Java class?

A. To execute any method of the class

B. To initialize an object of the class

C. To set the values of class attributes

D. To perform tasks associated with the object

10. What does the HTTP status code 201 indicate?

A. Request failed

B. Request fulfilled

C. Created resource

D. Unauthorized access

Answers

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

Explanations

1. What does a job in Jenkins represent?

A. A set of triggers for when the code runs

B. A task performed by Jenkins based on a schedule

C. A method for deploying code to production

D. A specific code repository used in projects

A job in Jenkins represents a task that is performed by the Jenkins automation server, often executed based on a defined schedule or in response to specific events, such as code commits. Jobs can encompass a variety of tasks, including building applications, running tests, and deploying code. The scheduling aspect allows teams to automate repetitive tasks, ensuring they run at designated times or conditions without manual intervention. This understanding of jobs in Jenkins highlights their role in continuous integration and continuous delivery (CI/CD) processes, where automating tasks is crucial for maintaining efficient workflows. While other options may touch upon aspects of Jenkins functionality, they do not encompass the core meaning of a job as effectively as the correct choice.

2. Which of the following best describes non-functional testing?

A. Testing the UI/UX of the application

B. Testing the functionality with valid inputs

C. Testing performance and security of the application

D. Testing broken paths in the code

Non-functional testing primarily focuses on the attributes and aspects of the software that define its performance, security, reliability, and usability, rather than testing specific functional requirements. The answer highlights that non-functional testing encompasses critical aspects such as performance, ensuring the application meets speed and responsiveness benchmarks, and security, which guarantees that the application is safeguarded against vulnerabilities. These aspects are crucial for determining how well the software operates under various conditions, making it a fundamental part of the testing process that goes beyond just checking if the application behaves correctly under normal circumstances. For instance, understanding how many users an application can handle simultaneously without degrading performance is a key aspect of non-functional testing, as is verifying that data is securely encrypted and that user privacy is maintained. The other options focus on elements that are more aligned with functional testing, which verifies that the application performs its intended functions as expected based on specific criteria. Testing the UI/UX of the application deals with the design and user interaction, validating the features with valid inputs checks correctness but not the performance or security, and testing broken paths looks at the application's failure modes rather than evaluating its non-functional aspects.

3. In what scenario would you use a Prompt Alert?

A. When you want to display information to the user

B. When you need user input for a task

C. When confirming user action

D. When needing to inform about a successful operation

Using a prompt alert is specifically designed for situations where user input is required. In this scenario, the prompt alert presents a dialog box that not only conveys a message but also includes an input field for the user to enter data. This interaction allows the application to gather information directly from the user, making it a direct line for responses to be processed further in the application logic. For example, if the application needs to know a user's name, the prompt alert can request that information and proceed based on the user's input. This function distinguishes it from other alert types, which serve different purposes. Other options include displaying information, confirming actions, or informing about successful operations, but none of these tasks necessitate direct input from the user, which is the fundamental purpose of a prompt alert.

4. How can parameters be passed to a scenario in Cucumber?

A. They are defined in the feature file as variables

B. They must be hardcoded within the steps

C. They are included in the Background section

D. They cannot be passed

Parameters in Cucumber can be passed to a scenario by defining them in the feature file as variables. This approach allows you to create more dynamic and reusable test scenarios. When using Cucumber, you typically define scenarios in a feature file using a plain language format. By specifying parameters within the scenario's steps using syntax such as "<parameter>", you can later map these parameters to actual values in your step definitions. This means that you can execute the same scenario with different data inputs, enhancing the flexibility and maintainability of your tests. For instance, if you have a step that involves logging into an application, you might define it as "Given I have a username '<username>' and password '<password>'". In this way, you could run the same step with different usernames and passwords, making your tests more efficient. The other options are not suitable for passing parameters. Hardcoding values within steps would limit reusability and would not allow for parameter variations across scenarios, while including parameters in the Background section is intended for setup steps applicable to all scenarios in a feature file but does not serve the purpose of passing individual parameters to steps. Lastly, stating that parameters cannot be passed is incorrect as it overlooks one of the central features of Cucumber's testing framework

5. Which feature of Selenium allows for waiting until a specific condition is met?

A. Fluent Wait

B. Implicit Wait

C. Sleep Function

D. Inactivity Wait

Fluent Wait is a feature in Selenium that allows for waiting until a specific condition is met before proceeding with the execution of the test script. It enables you to define the maximum wait time for a certain condition to occur, as well as to specify polling intervals to check for the condition. This is particularly useful when dealing with dynamic web elements that may not be immediately available for interaction, allowing for more resilience and reliability in automated tests. In addition to specifying a maximum wait time, Fluent Wait also allows you to set up exceptions to ignore during the wait period. This way, if the condition is not immediately met due to transient issues, the script can continue checking until the specified timeout or until the condition is fulfilled. This makes it a powerful tool for handling asynchronous behavior in web applications, which is common in modern software development. The other waiting mechanisms, such as Implicit Wait, are more suited for setting a global timeout for finding elements, while the Sleep function introduces a static wait period that doesn't account for dynamic conditions. Inactivity Wait is not a standard feature in Selenium, making Fluent Wait the clear choice for this scenario.

6. What condition typically causes a NullPointerException in Java?

A. Trying to access a null reference of an object

B. Accessing an array with an out-of-bounds index

C. Calling a method on an uninitialized variable

D. Assigning a value to a final variable

A NullPointerException in Java typically arises when you attempt to access an object or call its methods using a reference that is null. In Java, a null reference indicates that the variable does not point to any object in memory. Therefore, when you try to perform operations such as accessing fields or invoking methods on this null reference, the Java Virtual Machine throws a NullPointerException to signal that the operation cannot be completed because there is no valid object to work with. This is a common scenario for developers because it can often occur unintentionally, especially in complex codebases where the lifecycle of objects can be unclear, leading to a situation where an expected object is null. Recognizing this condition is crucial for debugging and ensuring the reliability of Java applications.

7. In the context of Selenium, how would you handle a hidden element?

A. Use CSS selectors

B. Use driver.findElement()

C. Use JavaScript to click the element

D. Use wait functions

Using JavaScript to click the element is an effective approach for handling hidden elements in Selenium. In many scenarios, an element may not be interactable through standard Selenium commands because it is hidden due to CSS properties like `display: none;` or `visibility: hidden;`. Selenium typically cannot engage with elements that are not visible, since it adheres to the web page's rendered view. By executing JavaScript, you bypass these visibility constraints imposed by the browser. JavaScript can directly manipulate the Document Object Model (DOM) and interact with elements regardless if they are currently rendered on the screen or not. For instance, using JavaScript's `element.click()` method allows you to trigger a click event on the element, even if it is considered hidden, which can be crucial in automated testing scenarios where you need to ensure specific actions are performed correctly. The strategy of using JavaScript augments your ability to interact with dynamically displayed elements, especially in applications where UI elements may change based on user actions or other conditions on the page.

8. 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.

9. What is the primary role of a constructor in a Java class?

A. To execute any method of the class

B. To initialize an object of the class

C. To set the values of class attributes

D. To perform tasks associated with the object

The primary role of a constructor in a Java class is to initialize an object of the class. When an instance of a class is created, the constructor is called automatically, and it sets up the initial state of the object. This is where you typically define any necessary initial values for class attributes and perform actions needed when an object is instantiated. Constructors can have parameters that allow for dynamic initialization of object attributes, enhancing flexibility and control over how objects are created. If no constructor is explicitly defined, Java provides a default constructor that initializes object attributes to their default values (e.g., null for objects, 0 for numeric types, false for boolean). The involvement of the constructor doesn't extend to executing arbitrary methods of the class or performing non-initialization-related tasks. While a constructor can call other methods within the class, its main purpose remains focused on the initial setup of the object itself.

10. What does the HTTP status code 201 indicate?

A. Request failed

B. Request fulfilled

C. Created resource

D. Unauthorized access

The HTTP status code 201 indicates that a request has been successfully fulfilled and that a new resource has been created as a result. This response is typically sent after a POST request, where the client submits data to the server in order to create a new entry. When a server responds with a 201 status code, it usually includes a Location header that provides the URL of the newly created resource, allowing the client to access it directly. This acknowledgment is critical in RESTful APIs, as it informs clients that their creation request was processed successfully and the intended resource now exists. The significance of the 201 status code lies in its role in indicating the successful creation of resources, which is a fundamental aspect of many web applications and APIs. It confirms that the server performed the expected action and provides useful information back to the client.

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!