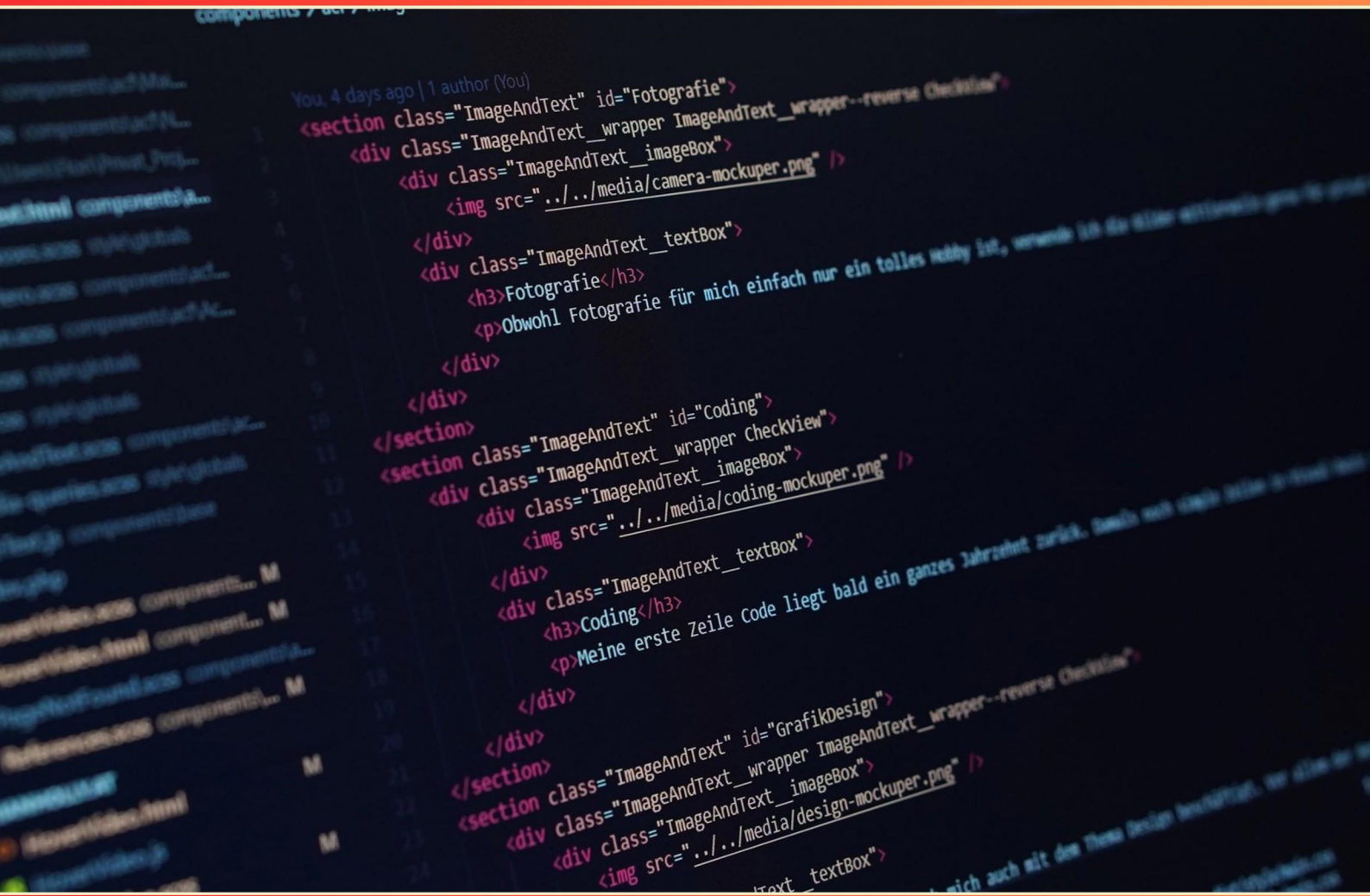
AUTOMATION DEVELOPER PROFESSIONAL PRACTICE TEST (SAMPLE)

STUDY GUIDE



SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

VISIT US ONLINE FOR THE FULL VERSION STUDY GUIDE

https://automationdevpro.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 framework is primarily used for behavior-driven development (BDD)?

A. JUnit

B. TestNG

C. Cucumber

D. Mockito

2. How does automation testing increase reliability?

A. By allowing random testing practices

B. Through consistent execution of tests without human error

C. By creating random test scripts

D. By using less documentation

3. What is the primary purpose of cross-browser testing in automation?

A. To ensure compatibility with multiple devices

B. To verify application security across platforms

C. To ensure applications function correctly across different web browsers

D. To evaluate the performance of web servers

4. What role does artificial intelligence (AI) play in automation testing?

A. AI reduces the cost of testing

B. AI enhances test automation by analyzing results

C. AI ensures manual testing accuracy

D. AI provides an intuitive user interface

5. Which collection is the most encountered with a dynamic size?

A. Dictionary

B. Array

C. List

D. Queue

SAMPLE

6. Which option best describes a descriptor in UI automation?

A. The unique ID of a UI element.

B. The "path" to the UI element, starting from the root, all the way to the target element.

C. Target and Anchor pair to identify and locate UI elements using unique attributes or identifiers to interact with the UI accurately and consistently.

D. A container for UI elements.

7. What is the primary purpose of automation in software development?

A. To enhance efficiency by reducing manual intervention in processes

B. To eliminate all manual work in software development

C. To increase the number of software bugs

D. To ensure every software is developed in the same way

8. State True or False: Check App State activity can be used as a condition for the Retry Scope activity.

A. True

B. False

9. What is a selector in automation development?

A. The "path" to the UI element, starting from the root, all the way to the target element.

B. The unique ID of a UI element.

C. An activity used to select an element.

D. A container for UI elements.

10. What kind of variable type can be useful for iterating over collected UI elements?

A. List

B. Dictionary

C. Array

D. IEnumerable<UiElement>

SAMPLE

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. Which framework is primarily used for behavior-driven development (BDD)?

A. JUnit

B. TestNG

C. Cucumber

D. Mockito

Cucumber is primarily used for behavior-driven development (BDD) because it allows developers to write test cases in a natural, human-readable language that can be understood by non-technical stakeholders. This aligns with the principles of BDD, which emphasizes collaboration between developers, testers, and business analysts. Cucumber uses a Gherkin-based syntax to define test scenarios, enabling teams to create a shared understanding of the system's behavior through features and scenarios that describe the desired functionality. This helps ensure that the development process remains aligned with business objectives, as user stories can be directly translated into executable specifications. The other frameworks mentioned serve different purposes. JUnit and TestNG are primarily testing frameworks for unit and integration testing but do not provide the same level of expressiveness for behavior-driven specifications. Mockito is a mocking framework used in unit testing to create mock objects, but it also does not align with the principles of behavior-driven development in the way Cucumber does.

2. How does automation testing increase reliability?

A. By allowing random testing practices

B. Through consistent execution of tests without human error

C. By creating random test scripts

D. By using less documentation

Automation testing increases reliability primarily through consistent execution of tests without human error. In manual testing, the risk of human mistakes—such as missed steps, incorrect data entry, or variations in testing environments—can introduce variability and potential inaccuracies in test results. However, automation allows for tests to be run in the same way each time, significantly reducing inconsistencies and enhancing the overall reliability of the testing process. Automated tests can be programmed to follow precise steps, thus ensuring that all aspects of the application are tested under the same conditions. This repeatability means that any defects or issues identified are more likely to be due to actual problems in the code rather than mistakes made during the testing process. Consequently, automation not only improves the reliability of test outcomes but also builds greater confidence in the software's quality before it reaches production. While other options may touch on various aspects of testing, they do not align with the essence of reliability. Random testing practices, creating random test scripts, and using less documentation can introduce uncertainty and reduce the thoroughness of testing, which ultimately detracts from the reliability that automation aims to achieve.

SAMPLE

3. What is the primary purpose of cross-browser testing in automation?

A. To ensure compatibility with multiple devices

B. To verify application security across platforms

C. To ensure applications function correctly across different web browsers

D. To evaluate the performance of web servers

The primary purpose of cross-browser testing in automation is to ensure applications function correctly across different web browsers. This type of testing is crucial because web applications may behave differently due to variations in how browsers interpret HTML, CSS, and JavaScript. By performing cross-browser testing, developers can identify issues related to user interface rendering, functionality, and overall user experience that may arise in specific browsers. Cross-browser testing helps ensure that users have a consistent experience regardless of the browser they choose to use. This is increasingly important as users may access the web through various browsers such as Chrome, Firefox, Safari, and Edge, which can each present unique challenges. By validating that an application works properly across these platforms, developers can minimize the risk of alienating users due to compatibility issues. While ensuring compatibility with multiple devices can be a part of overall testing strategies, the specific focus of cross-browser testing is on browser behavior rather than device types. Similarly, evaluating application security or web server performance falls outside the realm of what cross-browser testing addresses, as those focus areas target different aspects of application quality and stability.

4. What role does artificial intelligence (AI) play in automation testing?

A. AI reduces the cost of testing

B. AI enhances test automation by analyzing results

C. AI ensures manual testing accuracy

D. AI provides an intuitive user interface

Artificial intelligence significantly enhances test automation by providing advanced capabilities for analyzing results and making informed decisions. AI can sift through large volumes of test data quickly, identifying patterns, anomalies, and trends that may not be immediately apparent to human testers. This capability allows for more efficient debugging and optimization of test cases. Furthermore, AI-driven tools can predict potential failure points before they happen, enabling proactive adjustments to test strategies. In contrast, other roles ascribed to AI in this context may not accurately represent its primary benefits in automation testing. While reducing costs and ensuring accuracy in manual testing are valid goals, these are indirect benefits rather than the primary role of AI. Similarly, while an intuitive user interface can improve user experience, it does not encapsulate the core functionalities that AI brings to the testing process itself.

SAMPLE

5. Which collection is the most encountered with a dynamic size?

A. Dictionary

B. Array

C. List

D. Queue

The list is the most encountered collection with a dynamic size because it allows for constant resizing as elements are added or removed. Unlike arrays, which have a fixed size upon initialization, lists can grow and shrink dynamically based on the number of elements they contain. This makes lists particularly versatile for scenarios where the total number of items is not known at compile time or is expected to change frequently during runtime. In programming, lists are implemented in various languages with built-in mechanisms to handle resizing, which simplifies the management of collections. This dynamic nature ensures that when more items are added, the underlying structure can allocate additional memory, enabling seamless expansion. Conversely, when items are removed, lists can release memory appropriately, maintaining efficient use of resources. While dictionaries and queues also offer dynamic resizing capabilities, they serve different purposes and are not as commonly used for general-purpose collections of items compared to lists. Arrays, on the other hand, are static in size, making them less suitable for scenarios requiring frequent changes in the number of elements.

6. Which option best describes a descriptor in UI automation?

A. The unique ID of a UI element.

B. The "path" to the UI element, starting from the root, all the way to the target element.

C. Target and Anchor pair to identify and locate UI elements using unique attributes or identifiers to interact with the UI accurately and consistently.

D. A container for UI elements.

A descriptor in UI automation serves the purpose of identifying and interacting with UI elements accurately and consistently. This is accomplished through the use of a Target and Anchor pair, where specific attributes or identifiers are leveraged to locate the desired UI elements. The Target part of the descriptor refers to the UI element you want to interact with, while the Anchor serves as a reference point that helps to identify the Target in relation to other elements. This combination is essential when navigating complex UI frameworks where multiple elements might have similar identifiers but differ in context or hierarchy. Using a descriptor enables developers to create more robust automation scripts that can withstand changes in the UI layout or minor alterations to element properties, ensuring reliable interaction with the application during automated tests. The other choices, while they refer to elements of UI automation, do not encapsulate the comprehensive concept of a descriptor involving both Target and Anchor elements. This makes the understanding of the descriptor's function key in utilizing effective UI automation strategies.

SAMPLE

7. What is the primary purpose of automation in software development?

A. To enhance efficiency by reducing manual intervention in processes

B. To eliminate all manual work in software development

C. To increase the number of software bugs

D. To ensure every software is developed in the same way

The primary purpose of automation in software development is to enhance efficiency by reducing manual intervention in processes. Automation streamlines repetitive tasks, allowing developers to focus on more complex aspects of software creation and maintenance. By automating processes such as testing, deployment, and integration, teams can achieve faster turnaround times, reduce the likelihood of human error, and improve overall productivity. Automation also supports consistency in processes, ensuring that tasks are carried out in a uniform manner, which further enhances quality. While it aims to minimize manual work, it does not advocate for the complete elimination of human oversight, as there are aspects of software development that require human judgment and creativity. The other options do not accurately reflect the nuanced role of automation in software development. For instance, the idea that automation could eliminate all manual work underestimates the importance of human involvement in the development process. Similarly, suggesting that automation would increase bugs contradicts its purpose, which is to help identify and mitigate issues more efficiently. Finally, although automation can lead to standardization, it does not imply that every piece of software must be developed in exactly the same way, as flexibility and adaptability are also key components of effective software development practices.

8. State True or False: Check App State activity can be used as a condition for the Retry Scope activity.

A. True

B. False

The statement is true because the Check App State activity is indeed usable as a condition within the Retry Scope activity. The Retry Scope activity is designed to attempt a sequence of activities multiple times until a specified condition is met or a maximum number of retries is reached. When utilizing Check App State within the Retry Scope, you can set conditions based on whether a certain application state is active, which allows for more robust and reliable automation. For instance, if the application is not in the desired state, the Retry Scope can repeatedly execute the defined activities, checking the application state during each iteration. This integration enhances the resilience of automation workflows by allowing actions to continue until the application reaches the expected state. In this scenario, the condition checked by Check App State forms a crucial part of the decision-making process, ensuring that retries occur only when necessary, ultimately improving the efficiency and effectiveness of the automated process.

SAMPLE

9. What is a selector in automation development?

A. The "path" to the UI element, starting from the root, all the way to the target element.

B. The unique ID of a UI element.

C. An activity used to select an element.

D. A container for UI elements.

A selector in automation development serves as the "path" to a user interface (UI) element, tracing a sequence from the root of the document down to the specific target element. This path is crucial for accurately identifying and interacting with UI elements during automated tests or processes. By clearly specifying how to navigate to that element, selectors ensure that automation scripts can perform actions like clicking buttons, entering text, or verifying the visibility of elements. Selectors can be constructed in various ways, often incorporating attributes such as ID, class, or tag names, and can use different techniques like XPath or CSS selectors. This flexibility allows automation developers to target UI elements even in complex applications, enhancing the reliability and precision of their scripts. Understanding this concept is essential because a well-defined selector contributes significantly to the robustness of automation processes, helping to minimize errors and ensuring that the correct elements are manipulated as intended.

10. What kind of variable type can be useful for iterating over collected UI elements?

A. List

B. Dictionary

C. Array

D. IEnumerable<UiElement>

The choice of using IEnumerable<UiElement> is particularly advantageous for iterating over a collection of UI elements due to its flexibility and power within the context of automation. IEnumerable<T> is an interface that allows for iteration over a collection using a simple and efficient approach. When working with UI elements, it is common to have collections that can be fetched dynamically, and IEnumerable<T> provides a way to handle these collections without needing to load all elements into memory at once. This lazy evaluation allows for more efficient resource usage, particularly with a potentially large number of UI elements. Additionally, IEnumerable<T> supports LINQ queries, which can significantly enhance the ability to manipulate or filter UI elements on-the-fly. This capability can be extremely useful when trying to find specific elements based on various criteria, as well as performing operations on the collection. In contrast, while a List can also be used to hold multiple elements, it requires upfront allocation of all items in memory. A Dictionary is better suited for key-value pairs, which is less applicable when simply iterating over UI elements. An Array has a fixed size, making it less flexible for scenarios in which the number of elements is indeterminate or can change dynamically. Therefore, the IEnumerable<UiElement> option

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://automationdevpro.examzify.com

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

SAMPLE