

APPIAN ASSOCIATE DEVELOPER (ACD101) PRACTICE EXAM (SAMPLE)

STUDY GUIDE



**SAMPLE STUDY GUIDE
WITH LIMITED QUESTIONS**

**VISIT US ONLINE FOR
THE FULL VERSION STUDY GUIDE**

<https://appianacd101.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. How can you add test data into your rule inputs while editing an interface object?**
 - A. Enable a Test Mode in the interface settings.
 - B. Use a data file to prefill inputs.
 - C. Create a sample input in the Process Model.
 - D. Select the Test button and set test values.
- 2. Which of the following is NOT a recommended way to reduce the memory footprint of a process model?**
 - A. Split the process into sub processes so that only part of the overall process is active.
 - B. Reduce the number of process variables.
 - C. Do not store large data chunks like text strings in process variables.
 - D. Store large data chunks like text strings in process variables.
- 3. After selecting a record, you want to initiate an activity in the context of that selected record. What should you do next?**
 - A. Create a new record.
 - B. Add the process model as a record related action to that record.
 - C. Launch the process in a separate window.
 - D. Update the record's data directly
- 4. To set pv!dealershipName using a dealership lookup, which approach should you implement?**
 - A. Add another output variable defined as rule!VIM_getDealershipName(pv!vehicle.dealershipName) that saves into pv!dealershipName.
 - B. Modify pv!vehicle to contain dealershipName directly.
 - C. Create a new data store variable.
 - D. Change the function to accept a dealershipName field from input.
- 5. In an Appian process model, which feature allows a User Input Task to loop back to the same task after submission?**
 - A. Activity chained flow
 - B. Loop back control
 - C. Subprocess
 - D. Decision node

- 6. Where should you place logic to derive a computed attribute for a synced record type, such as a customer segment?**
- A. In a custom record field on the record type
 - B. In the interface
 - C. In the data source
 - D. In a separate process
- 7. When applying a default filter to a record type, what is a true statement about data excluded by the filter?**
- A. The data does not show up in the record list. End users are unable to clear the filter condition to view the data.
 - B. The data remains visible but is flagged as filtered.
 - C. The data is accessible by clearing the filter via a UI control.
 - D. The data is archived but accessible via a special query.
- 8. What prerequisite configuration is required to add a process model into a site as a tab?**
- A. Include a Start form in the process model.
 - B. Build a grid field with the data source as a query entity.
 - C. Configure a start event to trigger on site load.
 - D. Attach a record type as the data source.
- 9. You are configuring an employee onboarding User Input task that will be assigned to the Human Resources group. Based on the default behavior for task assignments, which statement is valid?**
- A. All HR members must accept before completion.
 - B. The task is automatically completed upon assignment.
 - C. One user in the HR group can accept the task for themselves and complete it.
 - D. The task cannot be completed by a single user.
- 10. What is the typical use of a rule input in Appian?**
- A. To pass values into a rule from a higher-level interface or integration.
 - B. To store data in a CDT.
 - C. To define UI components.
 - D. To trigger a process model.

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. How can you add test data into your rule inputs while editing an interface object?

- A. Enable a Test Mode in the interface settings.
- B. Use a data file to prefill inputs.
- C. Create a sample input in the Process Model.**
- D. Select the Test button and set test values.

Rule inputs on an interface are tested by supplying example values that match the declared inputs. Creating a sample input in the Process Model provides a ready-made dataset that aligns with those rule inputs, letting you apply realistic values directly while you're editing the interface. This gives you an immediate preview of how the interface behaves with real data and you can reuse that sample across tests, making the design process more efficient. Other options either aren't standard features or require manual, one-off entry, which is less convenient for iterative testing during design.

2. Which of the following is NOT a recommended way to reduce the memory footprint of a process model?

- A. Split the process into sub processes so that only part of the overall process is active.
- B. Reduce the number of process variables.
- C. Do not store large data chunks like text strings in process variables.
- D. Store large data chunks like text strings in process variables.**

Managing memory footprint in a process model hinges on keeping in-memory data small and loaded only as needed. Splitting the process into sub-processes helps because only the active portion consumes memory at a given moment, rather than the entire long-running flow. Reducing the number of process variables lowers the amount of data held in memory for each running instance. Avoid storing large data chunks directly in process variables because those variables stay in memory for the life of the process instance; large strings or blocks increase memory usage and can slow down execution. Instead, store big data externally (in a database or file/content store) and keep lightweight references in the process variables, loading or streaming data only when required. This approach keeps memory usage predictable and scalable.

3. After selecting a record, you want to initiate an activity in the context of that selected record. What should you do next?

- A. Create a new record.
- B. Add the process model as a record related action to that record.**
- C. Launch the process in a separate window.
- D. Update the record's data directly

When you want to act on a specific record, trigger the work through a related action on that record. In Appian, you define a related action on the record type that launches the process model, and you map the selected record's identifier (or other needed fields) into the process inputs. This makes the process run in the context of that exact record, so it can operate on or update that record as intended. Creating a new record would start something unrelated to the chosen one, launching the process in a separate window wouldn't tie it to the record, and updating the record directly bypasses the process and its rules.

4. To set pv!dealershipName using a dealership lookup, which approach should you implement?

- A. Add another output variable defined as rule!VIM_getDealershipName(pv!vehicle.dealershipName) that saves into pv!dealershipName.
- B. Modify pv!vehicle to contain dealershipName directly.
- C. Create a new data store variable.
- D. Change the function to accept a dealershipName field from input.

The main idea being tested is how to obtain a user-friendly display value from a data item using a reusable lookup rule and then store that result in a local display variable. In this approach, you create or use an existing rule that takes the dealership identifier (pv!vehicle.dealershipName, or whatever ID it is) and returns the actual dealership name. You then wire the result of that rule into the local output pv!dealershipName by defining an output variable for the rule invocation, and assign rule!VIM_getDealershipName(pv!vehicle.dealershipName) into pv!dealershipName. This keeps the original vehicle data structure intact, promotes reuse of the lookup logic, and ensures the UI shows the correct, human-readable name even if the underlying data changes. This pattern is preferable over altering the vehicle object to include the name directly, which would couple the display data to the data model; over creating a new data store variable just for this derived value, which adds unnecessary persistence; or over changing a function to pull the name from input fields, which can complicate the function's responsibilities.

5. In an Appian process model, which feature allows a User Input Task to loop back to the same task after submission?

- A. Activity chained flow
- B. Loop back control
- C. Subprocess
- D. Decision node

Looping a User Input Task in Appian is done with an activity-chained flow. This feature lets you connect the User Input Task back into the same process path, so after the user submits, the flow can return to the same task and present it again for additional input. By wiring the task back to itself through the chained flow, you create a cycle that repeats until a condition is met to exit. A subprocess would run a separate, child process and isn't used to loop within the same task. A decision node handles branching, not looping back to the same node. The looping effect is achieved using the activity-chained flow, which is why that option is the best choice.

6. Where should you place logic to derive a computed attribute for a synced record type, such as a customer segment?

- A. In a custom record field on the record type**
- B. In the interface
- C. In the data source
- D. In a separate process

When working with a synced record type, derive attributes belong in the record model itself by adding a custom record field on the record type. You can compute its value from existing fields (for example, a customer segment derived from revenue and industry) and expose it as part of the record payload. This makes the derived attribute available across all interfaces, grids, and logic that use the record, and keeps the calculation centralized in Appian rather than tying it to the UI or the data source. Putting the logic in the interface would limit its use to a single UI and require duplicating logic elsewhere. Implementing it in the data source couples business rules to the database and complicates maintenance, while a separate process adds unnecessary complexity and risk of drift.

7. When applying a default filter to a record type, what is a true statement about data excluded by the filter?

- A. The data does not show up in the record list. End users are unable to clear the filter condition to view the data.**
- B. The data remains visible but is flagged as filtered.
- C. The data is accessible by clearing the filter via a UI control.
- D. The data is archived but accessible via a special query.

When a default filter is set on a record type, the dataset shown in the record list is restricted to only those records that meet that filter. Records that don't meet the criteria are hidden from the list. In this context, the filter is applied automatically and, for that configuration, end users cannot clear it to view the hidden data. So the data excluded by the default filter does not appear in the record list, and the user cannot clear the filter to see it.

8. What prerequisite configuration is required to add a process model into a site as a tab?

- A. Include a Start form in the process model.**
- B. Build a grid field with the data source as a query entity.
- C. Configure a start event to trigger on site load.
- D. Attach a record type as the data source.

To add a process model into a site as a tab, you must provide a start form in the process model. The start form is what users interact with to begin a new process instance from the site tab, capturing the initial inputs needed to run the process. Without a start form, there isn't a user-facing interface to initiate the process from the site, so the tab cannot function as a proper entry point. The other options don't enable starting a process from a site tab: creating a grid field with a query entity is about displaying data, not starting a process; configuring a start event to trigger on site load would begin the process automatically in a way that's not intended for standard site interaction; and attaching a record type as the data source pertains to data display or relationships rather than enabling the start flow from a site tab.

9. You are configuring an employee onboarding User Input task that will be assigned to the Human Resources group. Based on the default behavior for task assignments, which statement is valid?

- A. All HR members must accept before completion.
- B. The task is automatically completed upon assignment.
- C. One user in the HR group can accept the task for themselves and complete it.**
- D. The task cannot be completed by a single user.

When a task is assigned to a group, it sits in the group's queue and is not automatically completed. The default behavior is that a single member from that group can claim (accept) the task, then proceed to complete it. This means one HR member can take ownership, finish the onboarding task, and mark it as done. The task does not require all HR members to accept, nor does it auto-complete upon assignment, and it can be completed by one user, not prevented from doing so.

10. What is the typical use of a rule input in Appian?

- A. To pass values into a rule from a higher-level interface or integration.**
- B. To store data in a CDT.
- C. To define UI components.
- D. To trigger a process model.

Rule inputs define the parameters a reusable rule expects and are used to pass values from a higher-level interface or integration into the rule so it can compute a result. This keeps the rule logic generic and reusable, since you supply the actual values at the call site rather than hardcoding them inside the rule. For example, a rule that calculates a discount can take inputs like price and discountRate; values come from the UI or from an external system and are passed into the rule to produce the result. These inputs aren't for storing data in a CDT, aren't used to define UI components, and don't directly trigger a process model. In short, the typical use is to pass values into a rule from an interface or integration.

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://appianacd101.examzify.com>

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

SAMPLE