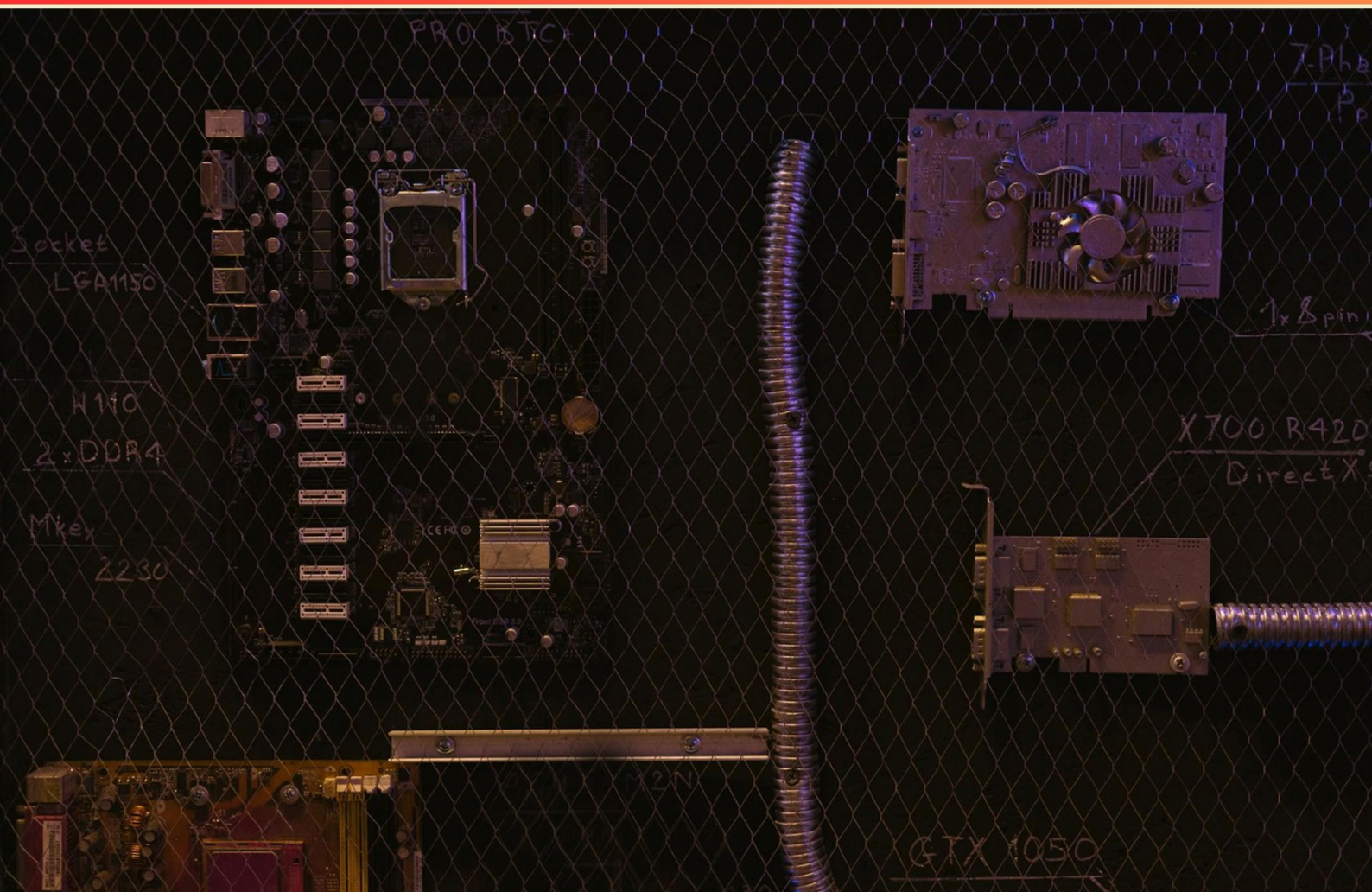


CLAUDE CERTIFIED ARCHITECT FOUNDATIONS (CCAR- F) PRACTICE EXAM (SAMPLE) STUDY GUIDE



SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

VISIT US ONLINE FOR
THE FULL VERSION STUDY GUIDE

<https://ccarf.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.

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. Which term describes a multi-agent anti-pattern in which a single coordinator manages all inter-subagent communication, error handling, and information routing?**
 - A. Hub-and-spoke architecture
 - B. Iterative refinement loop
 - C. Coordinator agent
 - D. Agentic loop anti-patterns
- 2. How does the microservices architecture handle data management challenges?**
 - A. Centralized data store shared by all microservices.
 - B. Use a single relational database across services.
 - C. Use a data lake for all stores.
 - D. Each microservice manages its own data store; requires data consistency strategies like eventual consistency and data replication patterns.
- 3. Which pattern describes a coordinator that evaluates synthesis output for gaps, re-delegates to search and analysis subagents with targeted queries, and re-invokes synthesis until coverage is sufficient?**
 - A. Fork_session
 - B. Coordinator agent
 - C. Parallel subagent spawning
 - D. Hub-and-spoke architecture
- 4. Which term refers to the Agent SDK configuration that restricts which tools an agent can call and must include 'Task' for a coordinator to invoke subagents?**
 - A. Model-driven decision-making
 - B. AgentDefinition
 - C. allowedTools
 - D. Iterative refinement loop
- 5. Scratchpad files are used to do what?**
 - A. To cache images for faster rendering.
 - B. To store user credentials.
 - C. To persist key findings across context boundaries to counteract context degradation during extended exploration sessions.
 - D. To log performance metrics.

- 6. What best describes the purpose of per-file local analysis passes followed by a cross-file integration pass?**
- A. Increase overall inconsistency
 - B. Improve depth and maintain consistency across files
 - C. Remove confidence signals
 - D. Eliminate the need for summaries
- 7. Describe the concept of defense in depth in security design.**
- A. A single strong firewall at perimeter.
 - B. Multiple overlapping controls across layers to reduce risk even if one layer fails.
 - C. Frequent password changes only.
 - D. Isolating all components in a single-layer security.
- 8. Which technique maximizes first-pass success and reduces resubmission cost?**
- A. Multi-pass review architecture
 - B. Sample-then-batch refinement
 - C. Attention dilution
 - D. Self-review limitation
- 9. Which statement best describes the purpose of stratified random sampling across document types and fields?**
- A. It reduces computational cost.
 - B. It measures ongoing error rates and detects novel error patterns across document types and fields.
 - C. It eliminates bias completely.
 - D. It standardizes all data types to a single format.
- 10. Which objective is typically established to measure and manage expected service performance in CCAR-F using SRE practices?**
- A. User interface aesthetics.
 - B. Marketing goals.
 - C. Hardware vendor selection.
 - D. Service Level Objectives.

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. Which term describes a multi-agent anti-pattern in which a single coordinator manages all inter-subagent communication, error handling, and information routing?

- A. Hub-and-spoke architecture
- B. Iterative refinement loop
- C. Coordinator agent

D. Agentic loop anti-patterns

Centralized coordination in a multi-agent system is the situation this term describes. When one coordinator handles all inter-agent communication, error handling, and information routing, the system becomes tightly coupled to that single agent. This creates a bottleneck and a single point of failure, making the system harder to scale as more agents are added. It also hampers autonomy and parallelism because other agents must rely on the coordinator to mediate every interaction, which can slow down decisions and propagate faults through the entire loop. The idea captured by this term—agentic loop anti-patterns—is that interaction is driven by one central authority rather than distributed, autonomous coordination among agents. To avoid it, distribute coordination responsibilities, adopt decentralized communication patterns, or use choreography or federated orchestration so agents communicate and resolve issues based on agreed protocols rather than funneling everything through a single coordinator.

2. How does the microservices architecture handle data management challenges?

- A. Centralized data store shared by all microservices.
- B. Use a single relational database across services.
- C. Use a data lake for all stores.

D. Each microservice manages its own data store; requires data consistency strategies like eventual consistency and data replication patterns.

In microservices, data management relies on decentralized ownership: each service has its own data store and operates independently. This setup supports autonomy, scalability, and resilience, but it means transactions across multiple services can't be treated as a single, all-or-nothing operation the same way they are in a monolith. To keep data coherent without global locking, you use strategies like eventual consistency and data replication patterns. Services publish events when their data changes, and other services listen and update their own stores accordingly, enabling asynchronous, decoupled synchronization. For coordinating multi-service workflows, the sagas pattern provides a way to execute steps with compensating actions if something fails, avoiding distributed transactions. Centralized data stores or a single relational database across services reintroduce tight coupling and bottlenecks, undermining autonomy and scalability. A data lake is useful for analytics, not for the operational data management of individual services. So the approach where each microservice manages its own data store and relies on eventual consistency and replication to stay in sync best fits the architecture.

3. Which pattern describes a coordinator that evaluates synthesis output for gaps, re-delegates to search and analysis subagents with targeted queries, and re-invokes synthesis until coverage is sufficient?

- A. Fork_session
- B. Coordinator agent**
- C. Parallel subagent spawning
- D. Hub-and-spoke architecture

This pattern is a central coordinator that orchestrates the whole process. It monitors the output of synthesis, identifies where information is missing or gaps exist, and then delegates targeted searches and analyses to subagents. After collecting the new data, it re-invokes synthesis and repeats the cycle until the coverage is complete. This iterative gap-filling and re-synthesis loop is the hallmark of a coordinating role that ensures the produced synthesis becomes progressively more complete. That's why this fits best: it explicitly describes evaluating outputs for gaps, re-delegating with focused queries to specialized subagents, and iterating synthesis until the result meets the required coverage. Other patterns describe alternative structural or parallel actions—like forking into separate paths, spawning multiple subagents without a central feedback loop, or a hub-and-spoke layout—without capturing this deliberate, iterative coordination and refinement flow.

4. Which term refers to the Agent SDK configuration that restricts which tools an agent can call and must include 'Task' for a coordinator to invoke subagents?

- A. Model-driven decision-making
- B. AgentDefinition
- C. allowedTools**
- D. Iterative refinement loop

Restricting which tools an agent can call is about controlling its capabilities through a specific configuration that lists the permitted tools. This allows the system to enforce safety and predictability by preventing arbitrary tool usage. The term that names this configuration is allowedTools, because it explicitly describes the set of tools the agent is allowed to invoke. Including the Task tool in that list is essential when a coordinator needs to delegate work to subagents, since Task is the mechanism used to orchestrate or invoke those subagents. Other terms don't describe this precise mechanism: model-driven decision-making refers to a broader approach, not a concrete tool-restriction setting; AgentDefinition is a general descriptor of an agent, not specifically about which tools are allowed; an iterative refinement loop is a problem-solving pattern, not a configuration for tool access.

5. Scratchpad files are used to do what?

- A. To cache images for faster rendering.
- B. To store user credentials.
- C. To persist key findings across context boundaries to counteract context degradation during extended exploration sessions.**
- D. To log performance metrics.

Scratchpad files act as external memory to keep track of findings, decisions, and important results across long, multi-step explorations where the model's current context window can't hold everything at once. By recording key insights in a scratchpad, you counter context degradation, so earlier conclusions don't vanish as you generate new analysis. This lets you recheck constraints, compare alternatives, and maintain consistency as you progress. In practice, during an extended architectural exploration, you might note the main requirements, critical trade-offs, and numerical results in the scratchpad and reference or update them later instead of re-deriving everything from scratch. That persistent record directly supports continuity of reasoning across context boundaries. Other options describe caching assets, storing credentials, or logging metrics, which serve different purposes and don't address maintaining the thread of reasoning over a long session.

6. What best describes the purpose of per-file local analysis passes followed by a cross-file integration pass?

- A. Increase overall inconsistency
- B. Improve depth and maintain consistency across files**
- C. Remove confidence signals
- D. Eliminate the need for summaries

The idea is to perform deep analysis within each file and then combine the results in a separate cross-file pass to enforce consistency across the entire set. Local analysis can dive into great detail, capturing nuanced patterns and signals specific to that file's structure. But analyzing files in isolation can lead to conflicting conclusions or missed context for decisions that span multiple files. The cross-file integration pass brings in global context, reconciles discrepancies, propagates common rules, and ensures that decisions are applied uniformly across all files. This two-step approach gives rich, file-level insight while maintaining a coherent, consistent overall result across the project.

7. Describe the concept of defense in depth in security design.

- A. A single strong firewall at perimeter.
- B. Multiple overlapping controls across layers to reduce risk even if one layer fails.**
- C. Frequent password changes only.
- D. Isolating all components in a single-layer security.

Defense in depth means you don't rely on a single barrier for security. Instead you place multiple overlapping controls across different layers—network, systems, applications, identity, and processes—so if one layer is breached, others are still in place to block, detect, or slow the attacker. This approach reduces risk because no single control is perfect or future-proof. For example, a strong firewall might be bypassed, or a password policy alone won't stop an attacker with stolen credentials. With layered protections—perimeter defenses, access controls, authentication, encryption, monitoring, backups, and incident response—the attacker has to defeat multiple independent measures, making breaches far less likely. The other options describe relying on one control, focusing on a single type of measure, or isolating everything in one layer, which does not provide the overlapping protection that defense in depth requires.

8. Which technique maximizes first-pass success and reduces resubmission cost?

- A. Multi-pass review architecture
- B. Sample-then-batch refinement**
- C. Attention dilution
- D. Self-review limitation

Maximizing first-pass success comes from catching and fixing the common issues early, before large-scale work is done. The sample-then-batch refinement approach does exactly that: you test a small, representative subset, identify the errors or inefficiencies, iterate on the fixes, and then apply the improved process to the full batch. This creates a fast and focused feedback loop, so the majority of outputs emerge correct on the first try. Because the big batch is built on validated improvements, the cost of resubmission is greatly reduced—they're less likely to recur once the fixes are proven to work in the smaller sample. Other approaches tend to add more review cycles or operate on the assumption that issues will still be caught later. A multi-pass review architecture can improve quality but often increases latency and total cost. Attention dilution describes a spread of focus that can undermine quality rather than enhance it. Self-review limitation restricts the system's ability to correct itself, which harms first-pass performance.

9. Which statement best describes the purpose of stratified random sampling across document types and fields?

- A. It reduces computational cost.
- B. It measures ongoing error rates and detects novel error patterns across document types and fields.**
- C. It eliminates bias completely.
- D. It standardizes all data types to a single format.

Stratified random sampling across document types and fields ensures representation from each category, so you can monitor how error rates differ across types and catch new error patterns as they emerge. By dividing the overall population into strata by document type and field and then sampling randomly within each stratum, you get a clearer picture of performance across the full diversity of inputs rather than a sample biased toward a single type. This approach makes it possible to measure ongoing error rates within each category and to detect novel or shifting error patterns that may appear in some document types but not others, which is essential for maintaining quality in a heterogeneous data set. While sampling can help manage workload, the primary goal here is to observe and understand error behavior across the different types and fields, not to eliminate bias completely or to standardize data formats.

10. Which objective is typically established to measure and manage expected service performance in CCAR-F using SRE practices?

- A. User interface aesthetics.
- B. Marketing goals.
- C. Hardware vendor selection.

D. Service Level Objectives.

In SRE practices, you measure and manage expected service performance by setting Service Level Objectives. SLOs are concrete, time-bound targets that describe how the service should perform from the user's perspective. They translate reliability and performance into numbers you can track, such as availability, latency, and error rate, often expressed as a percentage or a percentile over a defined period. These targets guide how the system is operated: they determine when to allocate more reliability effort, how to pace feature release (via the accompanying error budget), and how to respond to incidents. In CCAR-F, using SRE-style targets helps ensure the service meets defined performance expectations and that decisions around fixes, updates, and capacity are data-driven. Other options don't fit because they're not measurement targets for service performance: user interface aesthetics concern appearance rather than reliability; marketing goals are business outcomes unrelated to technical performance metrics; hardware vendor selection is about procurement, not ongoing performance measurement.

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://ccarf.examzify.com>

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

SAMPLE