

CCA-F

Claude Certified Architect – Foundations

Complete Practice Study Pack

Diagnostic Test · Full-Length Timed Mock Exam · Pre-Exam Mini-Simulator

PassITExams

passitexams.com

Independent, unofficial exam-preparation content. All questions in this guide are original and written by PassITExams; none are reproductions of real or leaked exam questions. This material is not affiliated with, endorsed by, or reviewed by Anthropic. "CCA-F" and "Claude Certified Architect – Foundations" refer to the third-party certification this guide is designed to help you prepare for; consult the certification provider's own official materials for authoritative exam content and policies.

How to Use This Guide

This study pack is built around PassITExams' Exam-Focused Study Approach: short, targeted practice mapped directly to the five CCA-F blueprint domains, with a plain rationale for every answer so you learn the reasoning, not just the letter.

What's inside

- **Part 1 — Free Diagnostic Test (30 questions, open-book):** a low-pressure baseline across all five domains. Use it before you study to find weak spots, or after your first pass through the domains to check retention.
- **Part 2 — Premium Timed Mock Exam (60 questions, closed-book, flagship product):** a full-length simulation weighted to match the published CCA-F blueprint (27% / 18% / 20% / 20% / 15%), so your practice score reflects the actual exam's emphasis, not an even split across domains.
- **Bonus — 10-Question Mini-Simulator:** a short, closed-book final check the night before — or morning of — your real exam appointment.
- **Full Answer Keys:** every question across all three sections, each with a one-line rationale explaining why the correct answer is correct.
- **Scoring Guide:** how to interpret your results and what to study next.

Recommended study sequence

1. Diagnose	Take the free 30-question diagnostic open-book. Note which domains feel shaky — this tells you where to spend study time.
2. Study the gaps	Revisit official domain material for your two lowest-scoring domains before moving on.
3. Simulate	Take the 60-question premium mock exam closed-book and timed (120 minutes is a reasonable target to match a proctored pace).
4. Review rationales	Read every rationale in the answer key — for both right and wrong answers — not just your missed questions.
5. Final check	The morning of your exam, run the 10-question mini-simulator closed-book as a warm-up, not a diagnostic.

A note on content

Every question in this guide is original, written by PassITExams to test the same concepts and judgment the real exam targets. None are copied or adapted from real, leaked, or proctored exam material. This guide is an independent study aid, not an official product of the certification provider or of Anthropic.

Domain weighting at a glance

Domain	Focus Area	Blueprint Weight
D1	Agentic Architecture & Orchestration	27%
D2	Tool Design & MCP Integration	18%
D3	Claude Code Configuration & Workflows	20%
D4	Prompt Engineering & Structured Output	20%
D5	Context Management & Reliability	15%

Part 1 — Free Diagnostic Test

30 questions · Open-book · Untimed · All 5 CCA-F domains. Take this before you study to see where you stand, or use it as a first checkpoint. Look things up as needed — the goal here is orientation, not exam simulation.

D1 — Agentic Architecture & Orchestration (27% of exam blueprint)

1. What best distinguishes an agentic system from a simple conversational assistant?

- A. It runs in a loop, selecting and invoking tools based on observed results until a goal is met
- B. It uses a larger context window than a conversational model
- C. It always requires multiple LLM calls per user turn
- D. It cannot be interrupted once started

2. A support ticket must be routed, summarized, and answered with a canned template, and the steps never change. What's the best pattern?

- A. A fixed workflow with no agentic decision-making
- B. A fully autonomous single agent with open-ended tool access
- C. A multi-agent hub-and-spoke system
- D. A peer-to-peer agent swarm

3. Which scenario most justifies moving from a simple workflow to an agentic architecture?

- A. The task has a fixed number of steps known in advance
- B. The task requires dynamic replanning based on unpredictable intermediate results
- C. The task must run faster than a single LLM call
- D. The task only needs one tool call

4. During task decomposition, an orchestrator agent should primarily aim to break work into subtasks that are:

- A. As numerous as possible for granularity
- B. Independent enough to execute with minimal cross-talk
- C. Always run in parallel regardless of dependencies
- D. Assigned to a single subagent regardless of specialty

5. Two subtasks have no data dependency between them. What execution strategy improves latency without harming correctness?

- A. Force sequential execution for safety
- B. Run them in parallel
- C. Retry each one ten times regardless of outcome
- D. Merge them into a single oversized prompt

6. An agent's plan fails partway through because a tool returned unexpected data. What is the recommended agentic response?

- A. Abort the entire session immediately
- B. Replan using the new information before continuing
- C. Ignore the unexpected data and proceed with the original plan
- D. Escalate to a human for every unexpected result, with no exceptions

7. In the orchestrator-subagent model, what is the orchestrator's primary responsibility?

- A. Executing every tool call itself
- B. Decomposing goals, delegating to subagents, and integrating their results
- C. Storing the entire conversation history for all subagents
- D. Replacing the need for any tool definitions

8. Why do well-designed subagents typically operate with context isolation?

- A. To reduce their own token costs indirectly
- B. To keep unrelated conversation history from leaking into their reasoning and confusing scope
- C. Because subagents cannot use tools
- D. Because the orchestrator requires it for billing purposes

D2 — Tool Design & MCP Integration (18% of exam blueprint)

9. In MCP, which primitive is best suited for exposing read-only contextual data, like a file's contents, to a model?

- A. Tools
- B. Resources
- C. Prompts
- D. Sampling

10. Which MCP primitive represents an invocable action the model can call with arguments?

- A. Resources
- B. Tools
- C. Roots
- D. Logging

11. MCP prompts are best described as:

- A. Arbitrary user chat messages
- B. Reusable, server-defined templates that structure a common interaction
- C. A replacement for tool definitions
- D. A logging mechanism only

12. A tool's description should primarily help the model decide:

- A. How much the tool costs in dollars
- B. When and how to call the tool correctly
- C. What color to render output in
- D. Which programming language implemented the tool

13. Which practice most improves an agent's ability to select the correct tool among many similar ones?

- A. Naming tools with generic, overlapping names
- B. Giving each tool a distinct name and a description that clarifies its unique boundary
- C. Hiding tool descriptions from the model
- D. Merging all tools into one mega-tool

D3 — Claude Code Configuration & Workflows (20% of exam blueprint)

14. What is the primary purpose of a CLAUDE.md file in a repository?

- A. To store API keys
- B. To give Claude Code persistent, project-specific context and conventions automatically
- C. To replace the README entirely
- D. To configure network firewall rules

15. Which is the best use of Plan Mode in Claude Code?

- A. Skipping review and applying changes immediately
- B. Reviewing a proposed approach before any code changes are made
- C. Disabling all tool use
- D. Replacing version control

16. Slash commands in Claude Code are best used for:

- A. One-off, unrepeatable exploratory questions
- B. Frequently repeated, structured workflows or prompts a team wants to reuse
- C. Disabling agentic behavior
- D. Managing billing settings

17. What is an Agent Skill in Claude Code primarily meant to encapsulate?

- A. A one-time chat message
- B. A packaged, reusable set of instructions and best practices for handling a specific kind of task
- C. A network security policy
- D. A billing plan

18. Integrating Claude Code into a CI/CD pipeline is most appropriate for tasks like:

- A. Manual, one-off interactive debugging sessions only
- B. Automated, repeatable checks such as code review or test generation on every pull request
- C. Replacing all human code review permanently
- D. Editing production databases directly with no oversight

19. What is the difference between resuming and forking a Claude Code session?

- A. They are identical
- B. Resuming continues the same named session; forking branches into a new session from a shared baseline
- C. Forking deletes prior history while resuming does not
- D. Resuming only works for read-only tasks

D4 — Prompt Engineering & Structured Output (20% of exam blueprint)**20. Which technique most reliably improves an agent's ability to produce a specific JSON structure?**

- A. Asking politely with no example
- B. Providing an explicit schema plus a worked example of valid output
- C. Increasing the temperature
- D. Removing all formatting instructions

21. Few-shot prompting refers to:

- A. Using zero examples and relying purely on instructions
- B. Including a small number of example input/output pairs to demonstrate the desired pattern
- C. Fine-tuning model weights
- D. Increasing the context window size only

22. When extracting structured data from unstructured text, which is the most reliable format to request for downstream parsing?

- A. Free-form prose
- B. A well-defined JSON schema with explicit field types
- C. A poem summarizing the data
- D. An unlabeled bullet list

23. Using XML-style tags, such as `<code>` and `</code>`, in a prompt primarily helps by:

- A. Making prompts shorter
- B. Giving the model and downstream parsers clear structural boundaries between sections of output
- C. Disabling structured output entirely
- D. Replacing the need for instructions

24. Context engineering is best described as:

- A. Randomly inserting text into a prompt
- B. Deliberately curating what information the model sees, and in what order, to optimize its performance
- C. Increasing the model's parameter count
- D. A synonym for fine-tuning

25. Which is a common cause of a model producing malformed JSON despite a schema being specified?

- A. The schema is too strict
- B. Free-form reasoning text was mixed into the same output field as the JSON, without separation
- C. The temperature was set to zero
- D. The prompt was too short

D5 — Context Management & Reliability (15% of exam blueprint)**26. The lost in the middle effect refers to:**

- A. Models forgetting instructions placed at the very start or end of context
- B. Models attending less reliably to information placed in the middle of a long context compared to the start or end
- C. A bug that only affects short prompts
- D. An effect only seen in image inputs

27. Which technique helps mitigate the lost-in-the-middle effect for a critical fact in a long document?

- A. Burying the fact in the middle with no emphasis
- B. Placing or repeating the critical information near the beginning or end of the context
- C. Removing all other context entirely regardless of relevance
- D. Using a smaller font

28. In a RAG pipeline, chunking refers to:

- A. Deleting irrelevant documents
- B. Splitting source documents into smaller retrievable segments
- C. Compressing the model's weights
- D. Encrypting the vector database

29. Hybrid retrieval typically combines which two approaches?

- A. Two different chunk sizes only
- B. Semantic, embedding-based search and keyword-based search like BM25
- C. Two separate language models
- D. Prompt caching and batch processing

30. Why might a production RAG system add citations to retrieved passages in its output?

- A. Citations have no practical purpose
- B. Citations let users verify claims against the actual source material, increasing trust and auditability
- C. Citations reduce retrieval accuracy
- D. Citations are required only for legal documents

Part 2 — Premium Timed Mock Exam

FLAGSHIP PRODUCT · PassITExams Full-Length Simulation

60 questions · Closed-book · Recommended time limit: 120 minutes · Weighted exactly to the published CCA-F blueprint so your score reflects real exam emphasis: Agentic Architecture & Orchestration 27%, Claude Code Configuration & Workflows 20%, Prompt Engineering & Structured Output 20%, Tool Design & MCP Integration 18%, Context Management & Reliability 15%.

Treat this like exam day: no notes, no lookups, one sitting. Mark your answers, then score yourself against the answer key in Part 6 before reviewing rationales.

D1 — Agentic Architecture & Orchestration (27% of exam blueprint)

1. Which multi-agent topology routes every message through a single coordinating node?

- A. Peer-to-peer
- B. Pipeline
- C. Hub-and-spoke
- D. Broadcast mesh

2. In a pipeline topology, subagent B typically:

- A. Starts only after receiving output from subagent A
- B. Runs fully in parallel with subagent A with no dependency
- C. Ignores the orchestrator entirely
- D. Communicates directly with every other agent simultaneously

3. What is a key risk of a peer-to-peer agent topology compared to hub-and-spoke?

- A. It is impossible to implement
- B. Coordination and conflict resolution become harder without a central authority
- C. It cannot scale past two agents
- D. It requires no communication protocol at all

4. Which is the most robust loop-termination condition for an autonomous coding agent?

- A. A fixed iteration cap regardless of task state
- B. Stopping the first time any tool call succeeds
- C. A combination of goal-completion checks and a bounded iteration or cost ceiling
- D. Never terminating automatically, always waiting for a keyword

5. An agent keeps calling the same failing tool with identical arguments. What design flaw does this most likely indicate?

- A. Missing error feedback that would let the agent adjust its next action
- B. Excessive parallelism
- C. Overuse of subagents
- D. Too small a context window

6. When a task's requirements are ambiguous, the most agentic-appropriate first move is to:

- A. Guess and proceed silently with the most likely interpretation
- B. Ask a clarifying question or gather more information before committing to a plan
- C. Terminate the session
- D. Run every possible interpretation in parallel

7. Which best describes conversational systems as opposed to agentic ones?

- A. They cannot call tools
- B. They primarily respond turn-by-turn without an autonomous multi-step action loop
- C. They always require multiple agents
- D. They cannot handle any user input

8. A production agent should generally use a tool-call budget primarily to:

- A. Guarantee a specific answer format
- B. Bound cost and prevent runaway loops
- C. Increase the model's creativity
- D. Disable retries entirely

9. What is the main benefit of adding a human-in-the-loop checkpoint before an agent takes an irreversible action, such as sending an email?

- A. It slows down every single tool call needlessly
- B. It lets a person catch mistakes before consequences that can't be undone occur
- C. It eliminates the need for planning
- D. It removes the need for any tool descriptions

10. Idempotent tool design matters most for agentic systems because:

- A. Agents never retry actions
- B. Retried actions, due to timeouts or errors, won't cause duplicate side effects if the tool is idempotent
- C. It makes tools run faster
- D. It removes the need for authentication

11. Which factor most influences whether a single-LLM-call approach or a multi-agent approach is more cost-effective for a task?

- A. The color scheme of the output
- B. Whether the task's complexity and need for specialization outweigh the added coordination overhead
- C. The programming language used
- D. The user's time zone

12. An orchestrator delegates a subtask, then receives back an ambiguous or incomplete result from the subagent. What should it do?

- A. Accept it uncritically and move forward
- B. Discard the whole plan and start over from scratch
- C. Request clarification or reassign/retry the subtask before integrating the result
- D. Terminate the entire agent system permanently

13. What is a handoff in multi-agent communication?

- A. A permanent shutdown of an agent
- B. The structured transfer of task ownership and relevant context from one agent to another
- C. A user login event
- D. A billing checkpoint

14. Why is scope-limiting a subagent's tool access, rather than granting all tools to every subagent, considered a reliability best practice?

- A. It reduces the chance of unintended or off-task actions
- B. It makes the system slower on purpose
- C. It removes the need for an orchestrator
- D. It guarantees zero errors

15. Sequential execution is generally preferred over parallel execution when:

- A. Subtasks have no shared dependencies
- B. A later subtask depends on the output of an earlier one
- C. Latency is the only concern
- D. Cost is irrelevant

16. Which is the best indicator that an agentic system, not a simpler prompt chain, is warranted?

- A. The task always follows the exact same three steps
- B. The number and order of steps can't be predicted before execution begins
- C. The output must be in JSON
- D. The task takes less than one second

D2 — Tool Design & MCP Integration (18% of exam blueprint)

17. What is a tool boundary problem?

- A. A tool's description is too short
- B. A tool's scope is too broad or overlaps with another tool's responsibility, causing selection errors
- C. A tool has no boundary conditions in unit tests
- D. A tool requires authentication

18. When designing a new MCP server for a ticketing system, which approach best supports interoperability across different agent clients?

- A. Hardcoding responses in a proprietary, undocumented format
- B. Following the MCP specification for tool schemas and structured responses
- C. Returning raw HTML meant only for a specific chat UI
- D. Requiring a specific LLM vendor to function

19. What should a well-designed tool return when an underlying API call fails?

- A. Nothing, so the agent assumes success
- B. A structured error the agent can reason about and potentially recover from
- C. A random success message
- D. The entire raw stack trace with no summary

20. Why should tool input schemas use strict typing and required fields where applicable?

- A. To make tools harder to call
- B. To reduce malformed calls and give the model clear expectations for valid arguments
- C. It has no effect on agent behavior
- D. To increase token usage intentionally

21. An MCP server exposes 40 tools with vague, overlapping names. What is the most likely production consequence?

- A. Faster agent responses
- B. Increased tool-selection errors and wasted calls
- C. Reduced token usage
- D. Guaranteed correctness

22. Which is a best practice for versioning an MCP tool's interface when its behavior changes?

- A. Silently changing behavior under the same tool name and schema
- B. Introducing a new tool name or version marker so existing integrations aren't silently broken
- C. Deleting the tool without notice
- D. Ignoring backward compatibility entirely

23. What does resource pagination or filtering support in an MCP server primarily help with?

- A. Reducing UI colors
- B. Letting an agent retrieve large data sets in manageable, targeted chunks
- C. Increasing latency intentionally
- D. Preventing any data access

24. Designing an MCP tool interface for interoperability across agents means prioritizing:

- A. A UI-specific rendering format
- B. A stable, self-describing schema independent of any one client's internals
- C. Hardcoded assumptions about a single agent's memory layout
- D. Proprietary binary encodings with no documentation

25. Which scenario best justifies splitting one broad tool into several narrower tools?

- A. The broad tool never fails
- B. The broad tool handles many unrelated operations, making its effects hard for the model to predict
- C. Narrow tools always cost more to run
- D. The broad tool has too short a name

26. What is the benefit of including example arguments in a tool's description?

- A. It has no measurable benefit
- B. It helps the model form correctly structured calls, especially for complex parameters
- C. It replaces the need for a schema
- D. It slows down the model unnecessarily

27. A tool call that partially succeeds, updating 3 of 5 records, should ideally:

- A. Report only success with no detail
- B. Return which parts succeeded and which failed so the agent can react appropriately
- C. Silently retry indefinitely without informing the agent
- D. Roll back regardless of what the agent needs

D3 — Claude Code Configuration & Workflows (20% of exam blueprint)

28. A team wants every developer's Claude Code sessions to follow the same lint rules and commit-message style. What's the most maintainable approach?

- A. Repeating the rules in every prompt manually
- B. Documenting them once in CLAUDE.md so they load automatically
- C. Emailing the rules to each developer
- D. Hardcoding them into the model weights

29. Which hook would be most useful for automatically running a formatter immediately after Claude Code edits a file?

- A. A hook that triggers before the session starts
- B. A PostToolUse hook tied to file-editing tools
- C. A hook that only runs on session termination
- D. There is no way to trigger actions after a tool runs

30. When should a developer prefer starting a brand-new session over resuming an existing one?

- A. Whenever continuing the same task with relevant prior context
- B. When the new task is unrelated to the prior session's context and a clean slate reduces noise
- C. Never; always resume
- D. Only when the task fails

31. What is a key benefit of naming Claude Code sessions explicitly?

- A. It has no benefit
- B. It lets you resume a specific prior conversation by name later, rather than guessing which session to continue
- C. It disables session history
- D. It is required for billing

32. Which is the most appropriate way to give Claude Code awareness of a monorepo's multiple, differently-configured subprojects?

- A. One global CLAUDE.md only, with no subproject detail
- B. Nested or subproject-level CLAUDE.md files alongside a top-level one
- C. Renaming all subprojects to match
- D. Disabling context loading entirely

33. In an agentic coding workflow, why is it good practice to have Claude Code run tests after making a change, rather than only asking it to describe what it did?

- A. Descriptions are always more reliable than test results
- B. Running tests provides objective, verifiable evidence the change works
- C. Tests are optional and add no value
- D. Tests replace the need for CLAUDE.md

34. What is the main risk of granting Claude Code unrestricted, un-reviewed permission to run arbitrary shell commands in a production environment?

- A. No risk, permissions don't matter
- B. Destructive or unintended commands could execute without a safety check
- C. It would make the session slower
- D. It disables slash commands

35. A slash command is defined to run a fixed multi-step release checklist. Compared to typing the same instructions from scratch each time, this primarily improves:

- A. Model accuracy on unrelated tasks
- B. Consistency and speed of repeating a known workflow
- C. The size of the context window
- D. Network security

36. Which configuration best supports letting Claude Code operate safely with tool permissions tailored per project?

- A. One identical, maximally permissive configuration shared by every project
- B. Project-specific permission settings that reflect that project's actual risk profile
- C. Disabling all tools everywhere
- D. Permissions configured only at the organization's billing level

37. What is the primary purpose of a PreToolUse hook?

- A. To log results after a tool finishes
- B. To inspect, modify, or block a tool call before it executes
- C. To restart the entire session
- D. To format final output only

38. When onboarding Claude Code to a new codebase, what is the most effective first step for productive sessions?

- A. Immediately asking for large, sweeping refactors
- B. Letting Claude Code explore the repo structure and, ideally, populating CLAUDE.md with key context
- C. Disabling all read tools
- D. Skipping any setup and hoping for the best

39. Which workflow best demonstrates using Claude Code for continuous integration rather than local development only?

- A. A developer chatting interactively on their laptop
- B. An automated pipeline step that invokes Claude Code non-interactively to review a pull request diff
- C. A one-time manual migration run by hand
- D. Disabling Claude Code during builds

D4 — Prompt Engineering & Structured Output (20% of exam blueprint)

40. What is a practical benefit of asking a model to think step by step before giving a final structured answer?

- A. It always reduces token usage
- B. It can improve reasoning quality on complex tasks before committing to a final output
- C. It removes the need for any output schema
- D. It guarantees a correct answer every time

41. Which practice helps when a structured extraction task involves many possible optional fields?

- A. Marking fields explicitly as required vs. optional in the schema and instructions
- B. Never mentioning which fields are optional
- C. Requiring every field to always be populated regardless of the input
- D. Omitting the schema entirely

42. What is a key risk of over-specifying a prompt with excessive, redundant instructions?

- A. There is no risk
- B. Important instructions can get buried or diluted, and cognitive load can reduce compliance
- C. It always improves output quality
- D. It removes ambiguity completely with no downside

43. When designing a system prompt for a customer-support agent, which is the best practice for tone control?

- A. Leaving tone completely unspecified
- B. Explicitly describing the desired tone and providing example phrasing
- C. Never giving any examples
- D. Instructing the model to be as verbose as possible always

44. Which output format is generally easiest for a downstream program to validate automatically?

- A. Natural-language prose
- B. A JSON object validated against a schema
- C. A hand-written summary paragraph
- D. An unstructured transcript

45. What's the benefit of separating a model's reasoning from its final answer using distinct output sections?

- A. It has no benefit
- B. Downstream systems can extract just the final answer while reasoning stays available for debugging
- C. It prevents the model from reasoning at all
- D. It guarantees zero errors

46. Which is the most effective way to reduce hallucinated fields in structured extraction from a source document?

- A. Instructing the model to always fill every field even without supporting evidence
- B. Instructing the model to omit or null out fields not supported by the source text
- C. Removing the schema entirely
- D. Increasing temperature

47. A prompt asks for output in three different formats in three different sentences: prose, then a list, then JSON. What issue does this create?

- A. No issue, models handle contradictions perfectly
- B. Conflicting formatting instructions confuse the model about which format actually applies
- C. It always improves formatting
- D. It reduces token usage

48. For a task with many similar but subtly different categories to classify text into, which prompting technique most helps disambiguate them?

- A. Providing zero examples and a one-word category list
- B. Providing a few labeled examples that specifically illustrate the tricky boundary cases
- C. Removing all category definitions
- D. Randomizing category names each run

49. Why might a developer prefer defining output structure via a strict schema, such as JSON Schema, over relying on natural-language formatting instructions alone?

- A. Schemas cannot be validated programmatically
- B. Schemas provide a machine-checkable contract, reducing drift and format errors at scale
- C. Schemas increase ambiguity
- D. Natural language is always more precise than a schema

50. What does an output formatting instruction typically control, separate from correctness of the underlying answer?

- A. The factual accuracy of the response
- B. The shape or structure, such as markdown, JSON, or plain text, in which the answer is delivered
- C. The model's training data
- D. The tool the model has access to

51. A prompt for a data-extraction agent includes ten unrelated example outputs before the real task. What's the likely effect?

- A. It has no effect on output
- B. Irrelevant examples can dilute the demonstrated pattern and increase token cost without benefit
- C. It always improves accuracy
- D. It removes the need for a schema

D5 — Context Management & Reliability (15% of exam blueprint)

52. What is the main cost benefit of prompt caching?

- A. It permanently deletes old prompts
- B. It avoids reprocessing unchanged portions of a prompt across repeated calls, lowering token cost and latency
- C. It increases the context window size
- D. It removes the need for retrieval

53. Which factor most affects whether a portion of a prompt is eligible for caching?

- A. The color of the text
- B. Whether that portion is identical and stable across repeated calls
- C. The number of tool calls made
- D. The user's account plan alone

54. What is a defining characteristic of the batch API compared to standard synchronous calls?

- A. It guarantees instant responses
- B. It processes large volumes of requests asynchronously, often at a lower cost, within a defined completion window
- C. It cannot handle more than one request
- D. It requires a live, open connection for the entire duration

55. For a workload that isn't latency-sensitive, such as overnight bulk document analysis, which approach best reduces cost?

- A. Standard synchronous calls exclusively
- B. The batch API
- C. Increasing the context window unnecessarily
- D. Running everything through a live agent conversation

56. As a conversation grows very long, what problem does compaction address?

- A. Running out of disk space
- B. The context window filling up with historical turns, threatening to exceed the limit or degrade coherence
- C. Network latency
- D. Tool authentication expiring

57. Which strategy best preserves long-conversation coherence when compacting history?

- A. Deleting all history with no summary
- B. Summarizing older turns while preserving key facts, decisions, and open threads
- C. Randomly deleting arbitrary turns
- D. Only keeping the very first message and nothing else

58. What is a key tradeoff in choosing a chunk size for RAG retrieval?

- A. There is no tradeoff, bigger is always better
- B. Smaller chunks improve retrieval precision but may lose surrounding context; larger chunks preserve context but reduce precision
- C. Chunk size only affects storage cost
- D. Chunk size has no effect on retrieval quality

59. Multi-index RAG is most useful when:

- A. All source data is homogeneous and small
- B. Different data types or sources, such as code versus policy documents, benefit from separate, specialized indexes
- C. It guarantees a single index will always suffice
- D. It is legally required

60. What does production hardening of a RAG pipeline typically include?

- A. Ignoring retrieval failures silently
- B. Handling retrieval failures gracefully, monitoring quality, and validating that retrieved content actually supports the answer
- C. Removing all citations
- D. Disabling retries

Bonus — 10-Question Mini-Simulator

10 questions · Closed-book · Recommended time limit: 15 minutes. A short, final pre-exam check spanning all five domains. Use it the day before or the morning of your exam as a confidence and warm-up check — not as a diagnostic.

D1 — Agentic Architecture & Orchestration (27% of exam blueprint)

1. In evaluating an agentic system before production rollout, what should be measured beyond simple task accuracy?

- A. Only the total number of lines of code
- B. Reliability across retries, cost per task, and behavior under tool failures
- C. The visual theme of the chat UI
- D. The agent's vocabulary size

2. A multi-agent system experiences context drift, where subagents lose track of the overall goal. What architectural fix most directly addresses this?

- A. Adding more subagents
- B. Having the orchestrator periodically re-share the current goal and relevant state with active subagents
- C. Removing all context from every subagent
- D. Increasing the tool-call budget

3. What is the primary tradeoff introduced by using a hub-and-spoke topology instead of direct peer-to-peer communication?

- A. Slightly more overhead through the hub in exchange for simpler, centralized coordination
- B. Impossibility of adding new agents
- C. Total elimination of latency
- D. No ability to delegate tasks

D2 — Tool Design & MCP Integration (18% of exam blueprint)

4. Why is least-privilege access an important principle in MCP tool design?

- A. It has no security relevance
- B. It limits what an agent, or a compromised tool, can do if misused, reducing potential damage
- C. It always improves speed
- D. It is required only for read-only tools

5. Which MCP concept lets a server advertise which parts of the filesystem or workspace it is allowed to operate on?

- A. Tools
- B. Roots
- C. Sampling
- D. Elicitation

D3 — Claude Code Configuration & Workflows (20% of exam blueprint)

6. What's a good reason to break a large feature request into multiple smaller Claude Code sessions or plan steps?

- A. It has no practical benefit
- B. Smaller, well-scoped steps are easier to verify and less likely to drift off course
- C. It always increases total cost with no benefit
- D. It disables version control

7. Why might a team version-control their CLAUDE.md file alongside the codebase?

- A. It has no relation to the code
- B. So project conventions stay in sync with the code as both evolve together
- C. To hide it from new developers
- D. Because it cannot be edited otherwise

D4 — Prompt Engineering & Structured Output (20% of exam blueprint)

8. Which best describes the tradeoff of adding many few-shot examples to a prompt?

- A. No tradeoff exists
- B. Better pattern demonstration versus increased token cost and potential context crowding
- C. Fewer examples always perform better in every case
- D. Examples eliminate the need for instructions entirely

9. When structured output must feed directly into another program, what is the most important prompt-engineering goal?

- A. Maximizing creative language
- B. Producing consistently parseable, schema-conformant output every time
- C. Varying the format for variety
- D. Minimizing instruction clarity

D5 — Context Management & Reliability (15% of exam blueprint)

10. Why does understanding token limits and cost tradeoffs matter when architecting the context window?

- A. Tokens have no cost implication
- B. Larger contexts increase cost and can affect latency and attention quality, so architects must balance completeness against these tradeoffs
- C. Token limits never apply in production
- D. Cost is unrelated to context size

Full Answer Keys

Every question from all three sections, with the correct answer and a one-line rationale. Review the rationales for questions you missed *and* the ones you guessed right.

Part 1 — Diagnostic Test Answer Key (Q1–30)

#	Domain	Answer	Rationale
1	D1	A	Autonomy plus an observe-act loop, not raw context size or call count, defines agentic behavior.
2	D1	A	When steps and order are fixed and predictable, a deterministic workflow is simpler and more reliable than an agent.
3	D1	B	Agentic architectures earn their added complexity when the system must adapt its plan as new information appears.
4	D1	B	Well-decomposed subtasks minimize the coordination overhead subagents need to complete their piece.
5	D1	B	Independent subtasks can run concurrently, reducing wall-clock time without changing the result.
6	D1	B	Dynamic replanning lets the agent incorporate new evidence rather than blindly following a stale plan.
7	D1	B	The orchestrator coordinates the overall goal while subagents handle bounded, delegated pieces of work.
8	D1	B	Isolated context keeps a subagent focused on its delegated task instead of inheriting irrelevant history.
9	D2	B	MCP resources are designed to expose data or content for context rather than invoking an action.
10	D2	B	Tools are the MCP primitive for actions the model actively invokes with parameters.
11	D2	B	MCP prompts let a server expose parameterized templates for common workflows.
12	D2	B	A clear description helps the model choose the right tool and supply the right arguments.
13	D2	B	Distinct names and clear boundaries reduce ambiguity when the model must choose between similar tools.
14	D3	B	CLAUDE.md supplies durable project context and conventions Claude Code reads automatically each session.
15	D3	B	Plan Mode lets you inspect and approve an approach before Claude Code executes changes.
16	D3	B	Slash commands package a recurring workflow into a short, reusable invocation.
17	D3	B	Skills bundle instructions and know-how for a recurring task type so Claude can apply them consistently.
18	D3	B	CI/CD integration suits automated, repeatable tasks that run consistently as part of the pipeline.
19	D3	B	Resuming picks up an existing session by name, while forking creates a new divergent branch from that point.
20	D4	B	A concrete schema and example give the model an unambiguous target to match.
21	D4	B	Few-shot prompting demonstrates the desired behavior through a handful of illustrative examples.

#	Domain	Answer	Rationale
22	D4	B	A defined JSON schema is machine-parseable and reduces ambiguity in field names and types.
23	D4	B	Tags create unambiguous boundaries that both the model and a parser can reliably locate.
24	D4	B	Context engineering is the practice of intentionally shaping what's in context to get better results.
25	D4	B	Interleaving reasoning with the JSON payload without separation often breaks strict parsers.
26	D5	B	Research on long-context recall shows information in the middle of a long context is retrieved less reliably than content near the edges.
27	D5	B	Positioning or repeating key facts near context boundaries improves the odds the model attends to them.
28	D5	B	Chunking breaks large documents into smaller pieces that can be independently retrieved and ranked.
29	D5	B	Hybrid retrieval blends semantic similarity search with keyword-based methods like BM25 to catch both conceptual and exact-term matches.
30	D5	B	Citations make retrieved claims traceable back to source text, supporting verification and trust.

Part 2 — Premium Mock Exam Answer Key (Q1–60)

#	Domain	Answer	Rationale
1	D1	C	Hub-and-spoke centralizes coordination through one hub that all spoke agents communicate with.
2	D1	A	Pipelines chain agents so each stage consumes the previous stage's output.
3	D1	B	Without a hub, peers must negotiate coordination themselves, which raises the chance of conflicting actions.
4	D1	C	Combining a real completion check with a safety ceiling avoids both premature stops and runaway loops.
5	D1	A	Without informative error feedback, the agent has no signal to change its approach on the next attempt.
6	D1	B	Resolving ambiguity before committing avoids wasted work on a plan built on a wrong assumption.
7	D1	B	Conversational systems focus on responding to each turn rather than autonomously pursuing a goal across many steps.
8	D1	B	Budgets cap how much an agent can spend in tool calls, protecting against uncontrolled loops.
9	D1	B	Checkpoints protect against irreversible harm from an agent's error by inserting human review at the critical moment.
10	D1	B	Idempotency ensures that if an agent retries a call, it doesn't create duplicate real-world effects like double charges.
11	D1	B	Multi-agent systems add coordination cost, so they should only be used when task complexity or specialization justifies it.
12	D1	C	Validating and, if needed, following up on a subagent's output prevents bad results from propagating into the final answer.

#	Domain	Answer	Rationale
13	D1	B	A handoff passes both responsibility and the context needed to continue work from one agent to the next.
14	D1	A	Narrowing a subagent's available tools to what its task requires limits the blast radius of mistakes.
15	D1	B	Genuine dependencies between steps require sequencing so each step has the input it needs.
16	D1	B	Unpredictable step count or order is a hallmark case for agentic planning rather than a fixed chain.
17	D2	B	Overly broad or overlapping tool scope makes it hard for the model to know which tool applies to a given situation.
18	D2	B	Adhering to the MCP spec keeps the server usable by any compliant client, not just one implementation.
19	D2	B	Structured, actionable error information lets the calling agent adapt instead of misinterpreting failure as success.
20	D2	B	Strict schemas reduce ambiguity, which lowers the rate of invalid or malformed tool calls.
21	D2	B	Vague, overlapping tools make correct selection harder, leading to more mistaken or wasted calls.
22	D2	B	Explicit versioning avoids breaking existing agents that rely on the old contract.
23	D2	B	Pagination and filtering let the agent pull only the relevant slice of a large resource instead of everything at once.
24	D2	B	A stable, self-describing schema lets many different agent clients use the same tool correctly.
25	D2	B	Splitting an overloaded tool into focused ones makes each tool's effect and correct usage more predictable.
26	D2	B	Concrete examples reduce formatting errors when a tool has complex or nested parameters.
27	D2	B	Granular partial-success reporting lets the agent decide how to handle the remaining failures.
28	D3	B	CLAUDE.md centralizes shared conventions so they apply automatically without repeating them each session.
29	D3	B	A PostToolUse hook fires right after a matching tool call, making it the right place to auto-format edited files.
30	D3	B	An unrelated task benefits from a clean context rather than carrying over irrelevant prior history.
31	D3	B	Named sessions can be looked up and resumed deliberately instead of relying on the most recent session.
32	D3	B	Nested CLAUDE.md files let each subproject specify its own conventions in addition to shared top-level guidance.
33	D3	B	Actually executing tests verifies the change instead of relying on the model's self-report of success.
34	D3	B	Without review, a mistaken or unintended command could cause irreversible production damage.
35	D3	B	Reusable commands standardize a repeated workflow so it's executed the same way every time.
36	D3	B	Tailoring permissions per project matches access to that project's actual sensitivity and risk.
37	D3	B	PreToolUse hooks intervene before execution, enabling validation or blocking of risky calls.

#	Domain	Answer	Rationale
38	D3	B	Establishing codebase context up front, such as via CLAUDE.md, leads to more accurate subsequent work.
39	D3	B	Non-interactive, pipeline-triggered invocation on pull requests is the hallmark of a CI/CD-integrated workflow.
40	D4	B	Explicit intermediate reasoning often improves accuracy on complex problems compared to jumping straight to an answer.
41	D4	A	Explicitly marking optional fields prevents the model from guessing or fabricating values for missing data.
42	D4	B	Piling on redundant instructions can bury the important ones and paradoxically reduce instruction-following.
43	D4	B	Explicit tone guidance plus examples gives the model a concrete target rather than an ambiguous adjective.
44	D4	B	JSON validated against a schema can be checked programmatically for correctness, unlike free prose.
45	D4	B	Separated sections let a parser cleanly grab the final answer while keeping reasoning available if needed.
46	D4	B	Explicitly permitting omission of unsupported fields discourages fabricating values that aren't in the source.
47	D4	B	Contradictory format instructions leave the model to guess which requirement takes priority.
48	D4	B	Examples targeting the ambiguous boundary cases teach the model exactly where categories diverge.
49	D4	B	A formal schema is enforceable and checkable, unlike loosely worded natural-language formatting requests.
50	D4	B	Formatting instructions shape presentation, which is a distinct concern from whether the content itself is correct.
51	D4	B	Examples should be relevant and few; irrelevant ones add cost and can confuse the intended pattern.
52	D5	B	Caching reuses previously processed context instead of recomputing it, cutting cost and latency on repeated calls.
53	D5	B	Caching relies on the cached segment staying identical across calls; changed content invalidates the cache.
54	D5	B	The batch API trades immediate responses for asynchronous, lower-cost processing within a set window.
55	D5	B	Batch processing suits non-urgent, high-volume workloads because it lowers cost in exchange for delayed results.
56	D5	B	Compaction summarizes or trims older turns so a long-running conversation can continue without overflowing context.
57	D5	B	A good compaction summary retains the decisions and facts still relevant to the ongoing task, not just arbitrary content.
58	D5	B	Chunk size balances precise, targeted retrieval against retaining enough surrounding context to be useful.

#	Domain	Answer	Rationale
59	D5	B	Separate indexes let each data type be embedded, chunked, and retrieved in a way suited to its structure.
60	D5	B	Hardening means the pipeline degrades gracefully and its outputs remain verifiable and monitored in production.

Bonus — Mini-Simulator Answer Key (Q1–10)

#	Domain	Answer	Rationale
1	D1	B	Production readiness depends on consistent behavior and cost under real-world failure conditions, not accuracy alone.
2	D1	B	Reinforcing the shared goal and state at checkpoints keeps distributed agents aligned as work progresses.
3	D1	A	Routing through a hub adds a coordination hop but simplifies conflict resolution and oversight.
4	D2	B	Granting only the access a tool needs limits the impact of a mistaken or malicious tool call.
5	D2	B	Roots communicate the boundaries of the filesystem or workspace a server is scoped to work within.
6	D3	B	Smaller scoped steps are easier to check for correctness and less prone to compounding errors.
7	D3	B	Keeping CLAUDE.md in version control ensures it evolves alongside the codebase it documents.
8	D4	B	More examples can clarify the pattern but consume tokens and context space that could hold other information.
9	D4	B	Downstream automation needs reliable, consistent structure far more than it needs varied phrasing.
10	D5	B	Every extra token in context adds cost and potential latency, so context size must be balanced against actual need.

Scoring Guide

These are informal PassITExams study benchmarks for self-assessment — they are not the certification provider's official passing standard. Always confirm the current passing score and format with the certification's official guide.

Score Range	What it means	Recommended next step
90–100%	Exam-ready	Do a final closed-book run of the mock exam within your target time, then rest before exam day.
75–89%	Close, with gaps	Re-study your lowest-scoring domain(s) from the table below, then retake the mock exam.
60–74%	Foundational gaps	Work through official domain material domain-by-domain before attempting another timed mock.
Below 60%	Not yet ready	Start with the diagnostic open-book, study each domain in turn, and re-test before scheduling your exam.

Score by domain, not just overall. Divide correct answers by questions attempted within each domain (use the Domain column in the answer keys) to find precisely where to focus your remaining study time — this matters more than your total score, since the real exam weights domains unevenly.

Ready for the real thing?

PassITExams builds portable, expert-reviewed practice materials so you can study anywhere — no subscription lock-in, no fluff, just exam-focused practice at an affordable price.

Explore more CCA-F practice sets, additional domain-focused drills, and other certification prep packs:

passitexams.com

Disclaimer

This is an independent, unofficial study resource. All practice questions are original content created by PassITExams and are not reproductions of any real or leaked exam questions. PassITExams is not affiliated with, endorsed by, or sponsored by Anthropic or the CCA-F certification provider. Certification names, exam blueprints, and passing standards referenced here belong to their respective owners; always verify current exam requirements against the official certification guide before you schedule or sit for the exam.