

FREE PDF

Claude Certified Architect – Professional (CCAR-P) practice questions

20 free questions with answers and explanations, written to the official exam guide.

Questions	pages 2–21
Check your answers	page 22
Answers and explanations	pages 23–36
Exam tips	page 37

Covers all 7 domains of the exam guide

Solution Design & Architecture	17%
Claude Models, Prompting & Context Engineering	13%
Integration	19%
Evaluation, Testing & Optimization	16%
Governance, Safety & Risk Management	14%
Stakeholder Communication & Lifecycle Management	14%
Developer Productivity & Operational Enablement	7%



Cert Made Easy
App for iPhone and Android



Question 1

A travel company evaluates its Claude booking agent on 150 tasks, each run in a sandbox with test airline and hotel systems. The grader passes a task when the agent's tool calls match a recorded reference sequence. After a model upgrade the pass rate fell from 81 to 64 percent. Engineers reading the failed transcripts found that most ended with correct, policy-compliant bookings reached through fewer or reordered calls.

How should the architect change the evaluation?

- ☐ (A) Record extra reference sequences for the failed tasks so the grader also accepts the new orders of calls.
- ☐ (B) Check the final booking records in the sandbox against each task's requirements instead of the call sequence.
- ☐ (C) Replace the sequence check with an LLM grader that scores each transcript against a written rubric for the task.
- ☐ (D) Run each task several times and pass it when most of the trials match the recorded reference sequence.

Note your answers as you go, on paper or in your notes app, then check them all at the answer key.

Question 2

A logistics firm asks for a Claude service that extracts delivery exceptions, such as damage, refusals and access problems, from drivers' free-text notes. The request says it must be "real time" and "low cost". Workshop notes show the notes arrive in one upload at the end of each shift, about 40,000 a night, and that the exception categories are already defined and ranked in the dispatch manual. The notes often contain customer names and home addresses.

Which two discovery actions should the architect take before choosing a design?

Choose 2

- ☐ A Benchmark Haiku, Sonnet and Opus on a sample of driver notes to settle the model tier for the cost estimate.
- ☐ B Ask dispatch how soon after the upload each exception must reach them, and what a late exception costs them.
- ☐ C Draft an SLA with a p95 response-time target for each extraction call and circulate it to dispatch for sign-off.
- ☐ D Run a workshop with dispatch and finance to rank the exception categories by the cost of missing each one.
- ☐ E Get a ruling from legal on the personal data in the notes, and on whether it must be removed before any call.

Question 3

Developers ask a sandboxed coding agent to 'suggest some changes to improve this function', and it lists ideas while its file-edit tool goes unused.

Which change makes the agent apply edits itself?

- ☐ (A) Raise the effort setting on the agent so it plans and carries out multi-step edits more thoroughly.
- ☐ (B) Switch the agent to an Opus model, which follows a developer's underlying intent more closely.
- ☐ (C) Add tool descriptions telling the agent when the file-edit tool fits, with an example input for each.
- ☐ (D) Add a system prompt instruction making action on requests, rather than advice, the agent's default.

Question 4

An architect is choosing human oversight for coding agents that take hundreds of routine file and shell actions on each task.

Which oversight design should the architect choose?

- ☐ (A) Run the agents in a sandbox with scoped file and network access, and require approval for actions outside it.
- ☐ (B) Require an engineer to approve each action the agents take, and log each approval decision for a later audit.
- ☐ (C) Let the agents run with full auto-approval, and have engineers review the complete session transcripts each evening.
- ☐ (D) Group each step's pending actions into a single approval prompt, so that engineers approve the actions in batches.

Question 5

Claude Models, Prompting & Context Engineering

An investment team's analyst assistant on Sonnet answers questions over ten annual reports, each about 8,000 words, placed in one prompt with the reports first and the question last. Analysts ask about specific figures and clauses. Audits find two failure patterns: answers that attribute a figure to the wrong company or draw on unrelated sections, and confident answers to questions that the reports do not address at all.

Which two changes should the architect make? **Choose 2**

- ☐ A Summarize each report with a Haiku call first and pass Claude the ten summaries in place of the full reports.
- ☐ B Enable prompt caching on the ten reports so that follow-up questions in a session reuse the processed prefix.
- ☐ C Instruct Claude to first copy the passages that bear on the question into quote tags, then answer from them.
- ☐ D Raise the effort level so that Claude reasons for longer over the ten reports before it writes each answer.
- ☐ E Allow Claude to reply that the reports do not cover a question when it finds no passage that supports one.

Question 6

An architect is hardening a support agent whose refund tool already enforces a strict schema on amount and order identifier.

Which control should stop refunds a customer is not entitled to?

- ☐ (A) Add a system-prompt rule stating that a refund amount may not exceed the customer's original order total.
- ☐ (B) Add minimum and maximum value constraints to the amount field in the refund tool's strict input schema.
- ☐ (C) Check in the tool handler that the amount fits the order total and that the caller owns the order.
- ☐ (D) Send each refund call to a second model that reviews it against the written refund policy before it runs.

Question 7

Teammates see Claude Code follow the testing conventions in a monorepo's root CLAUDE.md. One engineer, who works in a separate workspace that an internal script fills with copies of the service folders they own, says Claude ignores those conventions in every session and proposes rewriting them in capital letters or moving the team to a larger model.

What should the architect do FIRST?

- ☐ (A) Run /context in the engineer's session and check whether the root CLAUDE.md is listed under Memory files.
- ☐ (B) Rewrite the testing conventions in CLAUDE.md with IMPORTANT markers and concrete, checkable wording, then compare sessions.
- ☐ (C) Move the engineer's sessions to a more capable model tier and compare how often the conventions are followed.
- ☐ (D) Turn each testing convention into a hook in the shared project settings so it runs whatever Claude decides.

Question 8

A CRM maintenance agent built on the Claude API pauses before each database write, and a reviewer approves it on a screen that shows Claude's own summary of the change, such as 'I will archive 3 stale records.' Last week a reviewer approved that summary after Claude had read a vendor email, and the single tool call that followed archived 1,400 active accounts. Reviewers approve about 200 changes a day and want the screen to stay quick to read.

Which two changes to the approval screen would have let the reviewer stop this change? **Choose 2**

- ☐ A Display the tool name and every input argument from the tool_use block in place of Claude's own summary.
- ☐ B Instruct Claude to describe each change in more detail, naming the table, the filter and the expected row count.
- ☐ C Ask for approval before each tool_use block runs, rather than once for all of the calls in a turn.
- ☐ D Have the harness run the call's filter as a read-only count and show how many rows would change.
- ☐ E Require a second reviewer to approve each change summary before the harness executes the database call.

Question 9

A regional bank has never offered small-business loans under \$50,000, because manual underwriting took nine days and the margin could not cover the cost. Claude now drafts credit memos from applicants' bank statements and tax returns, and underwriters approve them. The sponsor's case for the investment is a new same-day loan product for businesses that competitors serve online. The board wants a quarterly scorecard showing whether that case is coming true.

Which two measures belong on the scorecard? **Choose 2**

- ☐ A The share of eligible applications that receive a credit decision on the same day they are submitted.
- ☐ B Underwriter hours spent on each credit memo, compared with the hours the manual process took per memo.
- ☐ C The number of small-business loans booked through the new product each quarter and the revenue they earn.
- ☐ D The underwriting staff cost saved since launch, measured against the bank's underwriting budget before launch.
- ☐ E The 95th-percentile time Claude takes to return a draft memo after an underwriter submits the documents.

Question 10

Evaluation, Testing & Optimization

An architect is choosing the release gate for a customer-facing booking agent whose customers expect a correct booking each time they make a request.

Which evaluation metric should gate the release?

- (A) Pass@k, the probability that at least one of k trials on each task ends in a correct booking.
- (B) Single-run success rate, the share of eval tasks that end in a correct booking when each task runs once.
- (C) Pass^k, the probability that k repeated trials of a task all end in a correct booking.
- (D) Mean success across k trials of each task, reported with a confidence interval around the average rate.

Question 11

A competitive-intelligence system has a lead agent that splits each request among four to six parallel subagents. Its messages to them are one line each, such as 'research competitor pricing'. Traces of weak reports show two subagents running near-identical searches, one returning prose where the lead needed a table, and regional pricing covered by none. Each subagent has a sound toolset and finds good sources when its task is clear.

Which change should the architect make?

- ☐ (A) Have the lead agent give each subagent an objective, an output format, guidance on tools, and its task boundaries.
- ☐ (B) Add a shared scratchpad where running subagents post the searches they have run and the findings they hold.
- ☐ (C) Add a reviewer agent that checks each finished report and sends any gaps back to the lead for another round.
- ☐ (D) Move the subagents to Opus so they apply better judgment when interpreting the short instructions they get from the lead agent.

Question 12

A platform team of 40 engineers wants Claude Code to inspect and change the team's own cloud resources while they work in the terminal. The cloud provider's command-line tool is installed on each laptop and signed in through each engineer's single sign-on role, and in trials Claude used it correctly on the tasks tried. No other Claude client needs this access. The central cloud team runs an operations agent that applies change requests through its pipeline, usually within a day.

Which integration mechanism BEST fits this need?

- ☐ (A) Build an MCP server that wraps the provider's API operations and register it in the project's MCP settings.
- ☐ (B) Let Claude Code run the provider's existing command-line tool through its shell, as engineers already do.
- ☐ (C) Delegate infrastructure requests to the cloud team's operations agent, which applies them through its pipeline.
- ☐ (D) Package the team's infrastructure runbooks as a Skill that Claude loads when an infrastructure task arises.

Question 13

Before funding a Claude assistant for HR policy questions, the sponsor watched a demo in which it answered ten hand-picked questions perfectly, and now tells leadership it will be fully accurate at launch. On 400 real employee questions it answers 88 percent correctly. Nearly every miss is a question the policy library does not cover, where the assistant guesses an answer. Launch is three weeks away.

What should the architect do FIRST?

- (A) Review the 400-question results with the sponsor by cause, then agree a launch bar and a reply for gaps.
- (B) Add a reranking step to retrieval and more worked examples to the prompt to lift accuracy before the launch review.
- (C) Delay the launch until the HR team has written policy articles for the question types that failed in the testing.
- (D) Release the assistant to a ten percent pilot group so live use shows the sponsor what accuracy to expect at launch.

Question 14

An equipment maker indexes 9,000 pump service manuals for a technician assistant, split into 800-token chunks in a vector store. Technicians ask questions such as "impeller bolt torque on the HX-40". In failed lookups, the right passage was indexed, but its chunk reads only "tighten the impeller bolts on this unit to 55 Nm"; the pump model is named once, on the manual's cover page. The top 20 results hold the same procedure from other pumps' manuals.

Which change BEST fixes these failed lookups?

- ☐ (A) Add a BM25 keyword index beside the vector store and merge both result lists with reciprocal rank fusion before generation.
- ☐ (B) Rerank the top 150 retrieved chunks with a dedicated reranking model and pass the best 20 of them to Claude.
- ☐ (C) Have Claude write a short context for each chunk from its full manual and prepend it before embedding and BM25 indexing.
- ☐ (D) Split the manuals into 200-token chunks with a 50-token overlap so that each embedding covers a single procedure step.

Question 15

Evaluation, Testing & Optimization

An architect is deciding what to do with a coding agent's capability eval, which the agent now passes almost completely.

Which two actions should the architect take? **Choose 2**

- ☐ A Keep comparing prompt variants on the capability eval, adding more trials per task to resolve small score differences.
- ☐ B Keep running the eval on every change as a regression suite whose pass rate should stay near full.
- ☐ C Retire the eval and rely on production monitoring of user-reported failures and draft acceptance rates.
- ☐ D Raise the grading strictness on the agent's existing tasks so that fewer of them pass and the scores spread.
- ☐ E Build a new capability set from tasks the agent currently fails, so that it starts at a low pass rate.

Question 16

Integration

A code-fixing service runs an Agent SDK agent in a fresh container per customer job, working on repositories that customers upload. The agent opens pull requests on GitHub and pulls packages from a private registry; both tokens are set as environment variables in the container, and egress is limited to those two hosts. In a red-team test, a README told the agent to put its environment into a commit message, and the registry token was published in the pull request.

Which design change closes this gap?

- ☐ (A) Issue both tokens fresh for each job with a one-hour lifetime, and revoke them as soon as the container shuts down.
- ☐ (B) Move both tokens to a proxy outside the container that adds them to the agent's outbound GitHub and registry calls.
- ☐ (C) Scan each commit message and request body leaving the container for token patterns, and block any match.
- ☐ (D) Run each job in a microVM instead of a container so the agent stays isolated from the host kernel.

Question 17

An internal IT help assistant retrieves knowledge-base passages and has Claude answer employees' questions, and employees rate each answer. Over one month the share rated unhelpful rose from 8 to 14 percent. In that month the documentation team re-chunked the knowledge base and the prompt team shipped two system-prompt revisions, and more of both are planned every few weeks. The application stores only each question, its answer and its rating.

Which change to the feedback loop should the architect make?

- ☐ (A) Log each answer's retrieved chunk IDs, prompt version and model in its trace, then compare the unhelpful answers by layer.
- ☐ (B) Add a weekly sample of the unhelpful answers to the system prompt as examples of responses Claude should avoid.
- ☐ (C) Score each rated conversation nightly with an LLM grader that gives every answer an overall helpfulness score from 1 to 5.
- ☐ (D) Re-embed the re-chunked knowledge base with a stronger embedding model and run the month's questions against it again.

Question 18

An account-management agent loads each customer's full record at the start of a session: five years of tickets, contracts, amendments and invoices, often 150,000 tokens. Most tasks need the latest invoice or the contract terms now in force, and sessions run three to ten turns. Reviewers find the agent quoting terms from superseded amendments in 12 percent of sessions, and its first reply takes about 20 seconds.

Which context strategy BEST fixes both problems?

- ☐ (A) Keep the full record in context behind a cache breakpoint so that later turns read it from the cache.
- ☐ (B) Have Claude summarize the full record at session start and load that summary in the record's place.
- ☐ (C) Embed the record's documents and attach the ten chunks closest to each user message before replying.
- ☐ (D) Load a dated index of the record's documents and give the agent a tool that fetches one by its ID.

Question 19

An architect has evaluated two designs for a claims-intake extractor on 5,000 graded claims: a Sonnet design scores 99.1 percent and a cheaper Haiku design 97.6 percent, each with a 95 percent confidence interval narrower than 1 percentage point. The company handles 40,000 claims a month, and each extraction error costs an adjuster about 20 minutes of rework. The operations director who must choose and fund one design says both look like about 98 percent and asks what the gap means for the team.

How should the architect present the trade-off to the director?

- (A) Add each design's confidence interval and the size of the graded set to the chart, so the gap reads as real.
- (B) Restate each design's error rate as adjuster rework hours per month, and set those beside each design's monthly model cost.
- (C) Show ten claims the Haiku design extracted wrongly beside the Sonnet design's correct output, so the rework becomes concrete.
- (D) Explain how the 5,000 graded claims were sampled and scored, so the director can trust both designs' accuracy figures.

Question 20

A pharmacy chain upgraded its customer assistant to a newer Claude model. The system prompt, written for the earlier model, includes: "CRITICAL: NEVER provide dosage information. ALWAYS refer to a pharmacist." The rule was added after the earlier model suggested dose changes to customers. Since the upgrade, 22 percent of customers who ask what dose is printed on their own prescription label, or which strengths a product comes in, are told to call a pharmacist. Questions about changing a dose are still referred correctly.

Which change BEST fixes the unnecessary referrals?

- ☐ (A) Add an exception rule beneath it that exempts label doses and product strengths from the dosage ban.
- ☐ (B) Move the dosage rule out of the system prompt and into each user turn, just before the question.
- ☐ (C) Rewrite the rule in plain words saying why dose changes need a pharmacist and what stays answerable.
- ☐ (D) Raise the effort level so Claude reasons longer about whether each dose question falls under the rule.

Check your answers

Tick each one you got right. Explanations start on the next page.

Count a multiple-response question as right only if you picked every correct letter and nothing else.

☐ Q1	B	☐ Q2	B, E	☐ Q3	D	☐ Q4	A	☐ Q5	C, E
☐ Q6	C	☐ Q7	A	☐ Q8	A, D	☐ Q9	A, C	☐ Q10	C
☐ Q11	A	☐ Q12	B	☐ Q13	A	☐ Q14	C	☐ Q15	B, E
☐ Q16	B	☐ Q17	A	☐ Q18	D	☐ Q19	B	☐ Q20	C

Your score ____ / 20

By domain

Solution Design & Architecture	Q9, Q11, Q17	____ / 3
Claude Models, Prompting & Context Engineering	Q3, Q5, Q20	____ / 3
Integration	Q12, Q14, Q16, Q18	____ / 4
Evaluation, Testing & Optimization	Q1, Q10, Q15	____ / 3
Governance, Safety & Risk Management	Q4, Q6, Q8	____ / 3
Stakeholder Communication & Lifecycle Management	Q2, Q13, Q19	____ / 3
Developer Productivity & Operational Enablement	Q7	____ / 1



Question 1

- B** Check the final booking records in the sandbox against each task's requirements instead of the call sequence.

Why B is right

Engineers found that most failed transcripts ended in correct bookings reached through fewer or reordered calls. The sandbox holds the bookings the agent made. Code can check them against each task's dates, travellers and policy rules. That accepts any valid route and still fails a wrong booking.

Why the other options are wrong

- A** Record extra reference sequences for the failed tasks so the grader also accepts the new orders of calls. Each new model or prompt can find another valid route, so the reference list keeps growing and the next unrecorded path fails.
- C** Replace the sequence check with an LLM grader that scores each transcript against a written rubric for the task.
The outcome is a set of records that code can check exactly. An LLM grader adds cost and noise and can still reward or punish the route.
- D** Run each task several times and pass it when most of the trials match the recorded reference sequence. The new routes are consistent, not random, so repeating the trials reproduces the same mismatches.

Exam tip: For agent evals, grade the outcome in the environment rather than the sequence of tool calls.

Source: anthropic.com/engineering/demystifying-evals-for-ai-agents

Question 2

- B** Ask dispatch how soon after the upload each exception must reach them, and what a late exception costs them.
- E** Get a ruling from legal on the personal data in the notes, and on whether it must be removed before any call.

Why B and E are right

How soon dispatch needs each exception, and what a late one costs them, decides between a synchronous service and the Message Batches API. Nobody has asked yet. The names and home addresses in the notes are personal data with no ruling. Legal's ruling decides whether redaction sits in front of every call, so it shapes the design too.

Why the other options are wrong

- A** Benchmark Haiku, Sonnet and Opus on a sample of driver notes to settle the model tier for the cost estimate.
Benchmarking is design work whose cost estimate depends on the open timing and data answers, and it would send uncleared personal data to the model.
- C** Draft an SLA with a p95 response-time target for each extraction call and circulate it to dispatch for sign-off.
A p95 response-time target assumes a synchronous call per note, which is the very choice the timing question has to settle first.

Question 2, continued

- D** Run a workshop with dispatch and finance to rank the exception categories by the cost of missing each one.

The dispatch manual already defines and ranks the categories, so this workshop repeats settled work.

Exam tip: In discovery, settle vague requirements and data constraints before you benchmark models or draft SLAs.

Sources: [Official exam page](#), [Anthropic Partner Academy](#); platform.claude.com/docs/en/api/service-tiers

Question 3

- D** Add a system prompt instruction making action on requests, rather than advice, the agent's default.

Why D is right

Listing ideas is a literal reading of the word 'suggest', not a fault in reasoning or tools. A default-to-action instruction in the system prompt makes implementing changes the default, so the agent edits files without each developer rephrasing. Rewording each request as a direct instruction also works, but only if every developer does it every time. The system prompt default does not depend on that.

Why the other options are wrong

- A** Raise the effort setting on the agent so it plans and carries out multi-step edits more thoroughly. Effort scales how thoroughly the agent works, but it does not change whether the agent reads 'suggest' as a request to act.
- B** Switch the agent to an Opus model, which follows a developer's underlying intent more closely. A more capable model follows the literal instruction just as precisely, so 'suggest' still yields a list, at a higher cost.
- C** Add tool descriptions telling the agent when the file-edit tool fits, with an example input for each. Better descriptions help the agent pick the right tool, but here it judged that no edit was requested, which a tool description does not change.

Exam tip: If an agent advises when you wanted it to act, set the default in the system prompt. A bigger model or more effort does not change how it reads the request.

Source: platform.claude.com/docs/en/build-with-claude/prompt-engineering/claude-prom...

Question 4

- A** Run the agents in a sandbox with scoped file and network access, and require approval for actions outside it.

Why A is right

Approval fatigue means people stop paying close attention to what they approve, so the prompts add delay without reliably adding safety. A sandbox fixes which directories the agent can write to and which hosts it can reach. Inside those limits the agent works without prompts. It asks a person only when an action would cross them, which keeps engineers able to step in when it matters.

Question 4, continued

Why the other options are wrong

- B** Require an engineer to approve each action the agents take, and log each approval decision for a later audit.
Per-action approval at this volume is the pattern that produces rubber-stamping, and a log of the clicks does not make them meaningful.
- C** Let the agents run with full auto-approval, and have engineers review the complete session transcripts each evening.
An evening review comes after the agents have acted, so a harmful command or exposed secret can be found but not stopped.
- D** Group each step's pending actions into a single approval prompt, so that engineers approve the actions in batches.
A batch asks for more attention per prompt, not less, so engineers skim the list and approve it whole.

Exam tip: For high-volume agent actions, put the human at the edge of a sandbox, not on every action.

Sources: anthropic.com/engineering/claude-code-sandboxing; code.claude.com/docs/en/agent-sdk/secure-deployment

Question 5

- C** Instruct Claude to first copy the passages that bear on the question into quote tags, then answer from them.
- E** Allow Claude to reply that the reports do not cover a question when it finds no passage that supports one.

Why C and E are right

Ten long reports in one prompt make a long-document task. Asking Claude to copy the passages that bear on the question into quote tags, word for word, grounds the answer in that text and sets the rest aside. That deals with figures pinned to the wrong company and drift into unrelated sections. The second failure is a different one. Explicit permission to say the material has no answer removes the pressure to produce one.

Why the other options are wrong

- A** Summarize each report with a Haiku call first and pass Claude the ten summaries in place of the full reports.
Analysts ask about specific figures and clauses, and summaries drop exactly those details while adding a step that can bring its own errors.
- B** Enable prompt caching on the ten reports so that follow-up questions in a session reuse the processed prefix.
Caching cuts cost and latency for a repeated prefix, but Claude reads the same content, so the grounding errors stay.

Question 5, continued

- D** Raise the effort level so that Claude reasons for longer over the ten reports before it writes each answer. Longer reasoning over the same noisy context adds no step that ties a claim to a passage and gives no permission to decline.

Exam tip: Two failure patterns usually need two fixes. Quote extraction targets misattribution, and permission to say 'I don't know' targets invented answers.

Sources: platform.claude.com/docs/en/test-and-evaluate/strengthen-guardrails/reduce-h...; platform.claude.com/docs/en/build-with-claude/prompt-engineering/claude-prom...

Question 6

- C** Check in the tool handler that the amount fits the order total and that the caller owns the order.

Why C is right

Anthropic's documentation promises schema-valid parameters and nothing more. A valid call can still ask for a refund far above the purchase, whether the model misreads the request or someone misleads it. Only code at the tool boundary can look up the order total and who owns the order. That check is exact, which a prompt rule or a second model cannot be.

Why the other options are wrong

- A** Add a system-prompt rule stating that a refund amount may not exceed the customer's original order total.
A prompt rule is guidance the model usually follows, not enforcement, so a schema-valid call above the order total can still run.
- B** Add minimum and maximum value constraints to the amount field in the refund tool's strict input schema.
A fixed ceiling in the schema cannot know each order's total, so a refund many times a small order still passes.
- D** Send each refund call to a second model that reviews it against the written refund policy before it runs.
A second model can also be manipulated, and it gives a probabilistic verdict on what is an exact arithmetic and ownership check.

Exam tip: Enforce business rules in code at the tool boundary. Schemas check shape and prompts give guidance, but neither checks entitlement.

Sources: platform.claude.com/docs/en/build-with-claude/structured-outputs; platform.claude.com/docs/en/agents-and-tools/tool-use/overview

Question 7

- A** Run `/context` in the engineer's session and check whether the root `CLAUDE.md` is listed under Memory files.

Why A is right

Only this engineer sees the problem, and the only thing different is the workspace. Claude Code loads `CLAUDE.md` from the working directory and every directory above it, so a workspace filled with copied service folders may not include the monorepo's root file at all. `/context` lists the memory files that loaded. If the root `CLAUDE.md` isn't there, Claude never saw the conventions, and no rewording or larger model will fix that. If it is there, the wording is the next thing to check.

Why the other options are wrong

- B** Rewrite the testing conventions in `CLAUDE.md` with **IMPORTANT** markers and concrete, checkable wording, then compare sessions.
Teammates already follow the same text, so the wording isn't what differs. If the file never loads in this workspace, stronger wording doesn't reach Claude either.
- C** Move the engineer's sessions to a more capable model tier and compare how often the conventions are followed.
A larger model can't follow instructions that aren't in its context, and the teammates show that the current model follows them when they load.
- D** Turn each testing convention into a hook in the shared project settings so it runs whatever Claude decides.
Hooks run shell commands at fixed events. That suits running the tests, not conventions about how to write them, and the copied workspace may lack the shared project settings too.

Exam tip: When instructions work for everyone but one person, check what loaded in that person's session before changing the instructions or the model.

Sources: code.claude.com/docs/en/memory; Official exam page, Anthropic Partner Academy

Question 8

- A** Display the tool name and every input argument from the `tool_use` block in place of Claude's own summary.
- D** Have the harness run the call's filter as a read-only count and show how many rows would change.

Why A and D are right

After reading the vendor email, Claude's summary said 3 stale records while the call's own arguments selected 1,400 active accounts. A screen that shows the `tool_use` arguments in place of the summary exposes that gap and stays as quick to read. A count the harness computes from those arguments does not depend on anything Claude says. Seeing 1,400 where the summary promised 3 stops the change.

Question 8, continued

Why the other options are wrong

- B** Instruct Claude to describe each change in more detail, naming the table, the filter and the expected row count.
The fuller description still comes from Claude, and an injected email can pull the description and the call apart again.
- C** Ask for approval before each tool_use block runs, rather than once for all of the calls in a turn.
The incident was a single tool call that already waited for approval, so per-call prompts that still show the summary would pass it.
- E** Require a second reviewer to approve each change summary before the harness executes the database call.
Both reviewers read the same misleading summary, which doubles the review work without ever showing the call.

Exam tip: An approval screen should show what will run, taken from the tool call itself, not the model's description of it.

Source: platform.claude.com/docs/en/agents-and-tools/tool-use/overview

Question 9

- A** The share of eligible applications that receive a credit decision on the same day they are submitted.
- C** The number of small-business loans booked through the new product each quarter and the revenue they earn.

Why A and C are right

The bank never offered these loans because manual underwriting was too slow and too costly to pay for itself. The sponsor's case rests on business the bank could not serve before. The share of eligible applications decided on the same day shows whether the new capability works end to end, underwriter approval included. Loans booked and the revenue they earn show whether that capability is turning into value.

Why the other options are wrong

- B** Underwriter hours spent on each credit memo, compared with the hours the manual process took per memo.
Hours per memo measure efficiency on one task. The bank never wrote these memos for small loans, and a faster memo does not show that any new loans are being made.
- D** The underwriting staff cost saved since launch, measured against the bank's underwriting budget before launch.
Staff cost saved is a cost measure, but the programme adds a product rather than replacing work, so there is little to save and the figure says nothing about new loans.

Question 9, continued

- E** The 95th-percentile time Claude takes to return a draft memo after an underwriter submits the documents.

Draft time is a technical service level that can sit at seconds while applications still wait days for approval.

Exam tip: Match each scorecard measure to the value the business case claims. A case built on new business needs outcome measures such as loans booked and revenue.

Source: [Official exam page](#), Anthropic Partner Academy

Question 10

- C** Pass^k , the probability that k repeated trials of a task all end in a correct booking.

Why C is right

Pass^k falls as k grows. If trials are independent, an agent that succeeds on 90 percent of trials passes three in a row only about 73 percent of the time. That gap is what customers would feel. $\text{Pass}@k$, a single run and a mean across trials all let a task that sometimes fails through the gate.

Why the other options are wrong

- A** $\text{Pass}@k$, the probability that at least one of k trials on each task ends in a correct booking. $\text{Pass}@k$ rises with k because one success among several tries is enough, which flatters an agent whose customers get one try per request.
- B** Single-run success rate, the share of eval tasks that end in a correct booking when each task runs once. One run per task cannot show whether a task that passed today fails on the next try, so flaky tasks pass the gate.
- D** Mean success across k trials of each task, reported with a confidence interval around the average rate. The mean estimates the per-trial rate more precisely, but a task that fails one trial in three still scores well, so the gate ignores the every-time requirement.

Exam tip: Read the requirement. 'Correct every time' points to pass^k , while 'at least one success in k tries' points to $\text{pass}@k$.

Source: anthropic.com/engineering/demystifying-evals-for-ai-agents

Question 11

- A** Have the lead agent give each subagent an objective, an output format, guidance on tools, and its task boundaries.

Why A is right

Subagents start with no view of the lead's plan or of each other's tasks. A one-line message such as 'research competitor pricing' leaves each one to guess its scope and format, so two overlap, one returns prose and nobody covers regional pricing. The traces show the subagents do well when the task is clear, so the fix belongs in the lead's delegation. Each brief should state an objective, an output format, guidance on tools and sources, and clear task boundaries.

Question 11, continued

Why the other options are wrong

- (B) Add a shared scratchpad where running subagents post the searches they have run and the findings they hold.
Subagents could see each other's searches, but nothing would tell one to return a table or give anyone regional pricing.
- (C) Add a reviewer agent that checks each finished report and sends any gaps back to the lead for another round.
It finds gaps only after a full round of work, so the duplicated searches are still paid for and each report needs another round.
- (D) Move the subagents to Opus so they apply better judgment when interpreting the short instructions they get from the lead agent.
No model can work out from 'research competitor pricing' which region another subagent covers or what format the lead wants. That information is missing from the message.

Exam tip: When subagents overlap or leave gaps, read the lead's delegation messages before you change the model or the tools.

Source: anthropic.com/engineering/multi-agent-research-system

Question 12

- (B) Let Claude Code run the provider's existing command-line tool through its shell, as engineers already do.

Why B is right

Claude Code can run the existing command-line tool through its shell, as the engineers already do. No tool definitions enter the context, and nothing new has to be built or kept current. Each action runs with the engineer's own permissions. MCP pays off when several clients need the same connection, and here no other client does.

Why the other options are wrong

- (A) Build an MCP server that wraps the provider's API operations and register it in the project's MCP settings.
A server wrapping the API adds tool definitions to each session and a component to maintain, for one client that already has a working interface.
- (C) Delegate infrastructure requests to the cloud team's operations agent, which applies them through its pipeline.
Handing requests to another team's agent turns interactive work into a queue measured in hours, for resources the platform team owns.
- (D) Package the team's infrastructure runbooks as a Skill that Claude loads when an infrastructure task arises.
A Skill adds instructions and reference files to Claude's context, but it opens no connection to the cloud.

Exam tip: Before building an MCP server, check whether a CLI already does the job for the one client that needs it. MCP pays off when several clients share a connection.

Source: code.claude.com/docs/en/best-practices

Question 13

- A** Review the 400-question results with the sponsor by cause, then agree a launch bar and a reply for gaps.

Why A is right

On 400 employee questions the assistant answered 88 percent correctly, and nearly every miss was a question the policy library does not cover. The misses share one cause, missing policy content. Agreeing a launch bar with the sponsor resets the promise using the data. A reply that admits the gap and points to HR replaces guessing with a defined behaviour.

Why the other options are wrong

- B** Add a reranking step to retrieval and more worked examples to the prompt to lift accuracy before the launch review.
A reranker reorders passages that retrieval found and examples shape answers, but neither supplies policy the library does not hold.
- C** Delay the launch until the HR team has written policy articles for the question types that failed in the testing.
Delaying on the architect's own call leaves the sponsor's promise standing, and any question type still uncovered will be guessed at again.
- D** Release the assistant to a ten percent pilot group so live use shows the sponsor what accuracy to expect at launch.
A pilot shows employees guessed HR answers while the sponsor still expects full accuracy, and the 400 employee questions already show what live use looks like.

Exam tip: When a sponsor's promise outruns the evidence, take the measured results to them first and agree the launch bar together.

Source: [Official exam page](#), Anthropic Partner Academy

Question 14

- C** Have Claude write a short context for each chunk from its full manual and prepend it before embedding and BM25 indexing.

Why C is right

The chunk says 'this unit' and never names the HX-40, so neither search can match it to the question. Contextual retrieval runs once, at indexing time. Claude reads each chunk with its full manual and writes a sentence or two that places it, such as 'HX-40 impeller service procedure'. The pipeline then embeds and keyword-indexes the enriched text, so both searches can match the model name.

Why the other options are wrong

- A** Add a BM25 keyword index beside the vector store and merge both result lists with reciprocal rank fusion before generation.
BM25 matches exact terms, but 'HX-40' is not in the chunk text, so the keyword index scores every manual's copy of the procedure alike.

Question 14, continued

- B** Rerank the top 150 retrieved chunks with a dedicated reranking model and pass the best 20 of them to Claude.
The reranker reads the same chunk text, which never names the pump, so it cannot tell the HX-40 procedure from the identical ones.
- D** Split the manuals into 200-token chunks with a 50-token overlap so that each embedding covers a single procedure step.
Smaller chunks carry even less of the manual, so the model name is still missing and the steps look even more alike.

Exam tip: When the right chunk is indexed but loses to look-alikes, check whether the chunk text names what users search for. If it does not, add that context to the chunk.

Source: anthropic.com/news/contextual-retrieval

Question 15

- B** Keep running the eval on every change as a regression suite whose pass rate should stay near full.
- E** Build a new capability set from tasks the agent currently fails, so that it starts at a low pass rate.

Why B and E are right

Running the old eval on every change catches backsliding, because a drop from a near-full pass rate signals a break. Near-full scores leave no room to see a gain on that set. A new set built from current failures starts low and shows whether a prompt or model change helps. One suite guards what works and the other measures progress.

Why the other options are wrong

- A** Keep comparing prompt variants on the capability eval, adding more trials per task to resolve small score differences.
Nearly every task already passes, so prompt variants land within a point or two and gains cannot show, however many trials run.
- C** Retire the eval and rely on production monitoring of user-reported failures and draft acceptance rates.
Production signals arrive after users are affected, and dropping the suite throws away a cheap check before release.
- D** Raise the grading strictness on the agent's existing tasks so that fewer of them pass and the scores spread.
Stricter grading of the same tasks rejects correct solutions, so the scores spread because the grader got harsher.

Exam tip: Capability evals should start at a low pass rate. When one saturates, move it to regression and build a harder set.

Source: anthropic.com/engineering/demystifying-evals-for-ai-agents

Question 16

- B** Move both tokens to a proxy outside the container that adds them to the agent's outbound GitHub and registry calls.

Why B is right

The README's instructions ran with whatever sat in the agent's environment, including both tokens. With a proxy outside the agent's boundary adding credentials to its GitHub and registry calls, the agent never sees the secrets. Anthropic's deployment guidance recommends this proxy pattern for agents that process untrusted content. Both hosts use HTTPS, so the proxy must terminate TLS or sit behind custom tools that call it, because a plain HTTP_PROXY setting only sees an encrypted tunnel.

Why the other options are wrong

- A** Issue both tokens fresh for each job with a one-hour lifetime, and revoke them as soon as the container shuts down.
The token was published while it was still valid, and anyone reading the pull request could use it within the hour. The agent still holds the secret.
- C** Scan each commit message and request body leaving the container for token patterns, and block any match.
The README can tell the agent to encode or split the token, so the scanner sees no pattern, and the secret is still within reach.
- D** Run each job in a microVM instead of a container so the agent stays isolated from the host kernel.
A microVM guards the host kernel, but this leak left through the agent's own permitted pull request.

Exam tip: If an agent reads untrusted content, keep credentials outside its environment. Short lifetimes and output scanning still leave the secret in its hands.

Source: code.claude.com/docs/en/agent-sdk/secure-deployment

Question 17

- A** Log each answer's retrieved chunk IDs, prompt version and model in its trace, then compare the unhelpful answers by layer.

Why A is right

The unhelpful share rose in a month when the knowledge base was re-chunked and the system prompt changed twice. The app stores only the question, the answer and the rating, so nobody can tell which change hurt. With chunk IDs, prompt version and model on each trace, the team can see whether a bad answer drew on poor passages or got good passages and still answered badly. Every future release then becomes traceable the same way.

Why the other options are wrong

- B** Add a weekly sample of the unhelpful answers to the system prompt as examples of responses Claude should avoid.
It changes the prompt a third time before anyone knows the prompt is at fault, and examples of bad answers cannot restore passages lost in re-chunking.

Question 17, continued

- (C) Score each rated conversation nightly with an LLM grader that gives every answer an overall helpfulness score from 1 to 5.
An overall helpfulness score restates the drop the ratings already show and still cannot separate retrieval failures from prompt failures.
- (D) Re-embed the re-chunked knowledge base with a stronger embedding model and run the month's questions against it again.
It assumes re-chunking caused the drop and stacks a third change on two unexplained ones, which makes the next trend even harder to read.

Exam tip: When several layers change at once, fix attribution first. Log which chunks, prompt version and model produced each answer before you change anything else.

Source: [Official exam page, Anthropic Partner Academy](#)

Question 18

- (D) Load a dated index of the record's documents and give the agent a tool that fetches one by its ID.

Why D is right

Long, noisy context degrades recall, which is how terms from superseded amendments get quoted. A dated index is small and lets the agent pick the contract in force or the latest invoice. The tool then loads only those documents. The first reply no longer waits on 150,000 tokens.

Why the other options are wrong

- (A) Keep the full record in context behind a cache breakpoint so that later turns read it from the cache. Caching speeds up later turns, but the first reply still processes the whole record, and the superseded amendments stay in context.
- (B) Have Claude summarize the full record at session start and load that summary in the record's place. A summary still needs a pass over the full record before the first reply, and it paraphrases contract terms the agent needs word for word.
- (C) Embed the record's documents and attach the ten chunks closest to each user message before replying. A superseded amendment reads almost exactly like the one in force, so similarity search keeps sending it to Claude.

Exam tip: When most tasks need a few known documents, give the agent an index and a fetch tool instead of loading the whole record.

Source: anthropic.com/engineering/effective-context-engineering-for-ai-agents

Question 19

- B** Restate each design's error rate as adjuster rework hours per month, and set those beside each design's monthly model cost.

Why B is right

At 40,000 claims a month, the Sonnet design makes about 360 errors, or 120 rework hours. The Haiku design makes about 960 errors, or 320 hours. The cheaper design therefore costs about 200 extra adjuster hours a month. Set beside what Haiku saves on the monthly model bill, the gap becomes a choice the director can compare and fund.

Why the other options are wrong

- A** Add each design's confidence interval and the size of the graded set to the chart, so the gap reads as real.
Each interval already spans less than 1 percentage point, so the gap is not in doubt, and intervals still speak in percentages.
- C** Show ten claims the Haiku design extracted wrongly beside the Sonnet design's correct output, so the rework becomes concrete.
Ten examples show what an error looks like but not how many there will be, and the choice turns on their volume and cost.
- D** Explain how the 5,000 graded claims were sampled and scored, so the director can trust both designs' accuracy figures.
Nobody questioned how the figures were measured, so a method walk-through answers a concern the director has not raised.

Exam tip: Translate model metrics into the units the decision-maker budgets in, such as hours, money or cases.

Source: [Official exam page](#), Anthropic Partner Academy

Question 20

- C** Rewrite the rule in plain words saying why dose changes need a pharmacist and what stays answerable.

Why C is right

Taken literally, 'NEVER provide dosage information' covers reading back the dose printed on a label, so the rule is too broad as well as too forceful. The newer model is simply following it as written. Anthropic advises giving the reason behind an instruction, because Claude can generalise from the explanation. Once the rule says dose changes need a pharmacist, Claude can read a label or list strengths and still refer dose changes.

Why the other options are wrong

- A** Add an exception rule beneath it that exempts label doses and product strengths from the dosage ban. The absolute NEVER and its exceptions now conflict, and the rule still gives no reason to generalise from, so the next harmless dose question it does not list is referred too.
- B** Move the dosage rule out of the system prompt and into each user turn, just before the question. Placement changes how prominent the rule is, not what it covers, so label readings are still referred.

Question 20, continued

- Ⓓ Raise the effort level so Claude reasons longer about whether each dose question falls under the rule. The written rule covers all dosage information, so longer reasoning still concludes that a label question falls under it.

Exam tip: When a newer model over-applies a rule, rewrite the rule in plain words with its reason rather than adding exceptions beneath it.

Source: platform.claude.com/docs/en/build-with-claude/prompt-engineering/claude-prom...

SAVE THIS PAGE

CCAR-P exam tips

- 1 Enforce business rules in code at the tool boundary. Schemas check shape and prompts give guidance, but neither checks entitlement.
- 2 If an agent reads untrusted content, keep credentials outside its environment. Short lifetimes and output scanning still leave the secret in its hands.
- 3 An approval screen should show what will run, taken from the tool call itself, not the model's description of it.
- 4 For agent evals, grade the outcome in the environment rather than the sequence of tool calls.
- 5 Read the requirement. 'Correct every time' points to pass^k, while 'at least one success in k tries' points to pass@k.
- 6 Translate model metrics into the units the decision-maker budgets in, such as hours, money or cases.
- 7 In discovery, settle vague requirements and data constraints before you benchmark models or draft SLAs.
- 8 When instructions work for everyone but one person, check what loaded in that person's session before changing the instructions or the model.



Good luck on exam day

Keep practising with our CCAR-P question set. There are more free questions on CertMadeEasy.com and in the app.

[More free CCAR-P questions →](#)

