

PRACTICE QUESTIONS

Anthropic

CCAR-F

Claude Certified Architect - Foundations

SHARED BY [u/Curious-Cloud-5664](#) on aiexamcerts

EDITION Free Set 2

QUESTIONS 10

SHARED October 18, 2026

EXAM Anthropic CCAR-F, Claude Certified Architect - Foundations

Shared by u/Curious-Cloud-5664 on aiexamcerts

10 practice questions for the Anthropic CCAR-F exam, Claude Certified Architect - Foundations, each followed by its answer and a short explanation. More free sets and the full practice exam:

aiexamcerts.com/anthropic/ccar-f. Spotted a mistake? Email support@aiexamcerts.com with the exam code CCAR-F and the question number.

1. A synthesis subagent keeps attempting web searches it was never meant to run. What is the best fix?
 - A Restrict its tool set to the tools its role needs, and route any cross-role request through the coordinator
 - B Give it every available tool so that it has more options when a search seems necessary to finish the task
 - C Add a line to its system prompt telling it never to think about searching, since that stops the unwanted behavior
 - D Increase its iteration cap so that it reaches the end of its task before it ever has the chance to start searching

Answer: A

Agents with tools outside their specialization tend to misuse them, so scoping each subagent to its role's tools addresses the cause. Adding every tool, or relying only on a prompt line, leaves the misuse possible. Raising an iteration cap does not prevent the wrong tool from being used.

2. A skill's allowed-tools frontmatter restricts which tools the skill can use while it runs.
 - A True
 - B False

Answer: A (True)

The allowed-tools field in a skill's SKILL.md frontmatter limits the tools available during that skill's execution. It is useful for keeping a skill from taking destructive actions. The statement is true.

3. Which tool_choice setting forces the model to call one specific named tool first?
 - A auto, which lets the model decide whether to answer in plain text or to call a tool
 - B any, which requires some tool call but lets the model choose which of the available tools to use
 - C {"type": "tool", "name": "extract_metadata"}, which requires that exact tool to be called
 - D A tool name written into the system prompt, which the model reads as a hint about the order of its steps

Answer: C

A forced tool choice names the exact tool the model must call, which is how to guarantee an extraction runs before enrichment. The auto setting can return text, and any requires a tool call without choosing the specific one. A name written in the system prompt is only a hint and is not enforced.

4. Few-shot examples are most useful for which situation?
- A Replacing a schema when the output must be machine-parseable, without any validation step being added to the pipeline at all
 - B Showing how ambiguous cases should be handled, with reasoning that explains why one action was chosen over the plausible alternatives
 - C Forcing the model to call a tool on every turn of a conversation, regardless of whether a tool is actually needed
 - D Reducing the cost of batch processing by shortening every prompt in the batch to the smallest possible size

Answer: B

Few-shot examples work well for demonstrating judgment in ambiguous cases, including the reasoning behind a choice. They do not replace a schema or validation, and they do not force tool calls. Shortening prompts for cost is a different optimization from few-shot prompting.

5. Which two practices help a coordinator avoid redundant or incomplete research? (Choose two.)
- A Partition the research scope so that each subagent receives distinct subtopics or source types
 - B Send the same broad query to every subagent so that the results can be cross-checked against each other
 - C Let each subagent decide its own scope with no coordinator input, so that the scope adapts to what it finds
 - D Evaluate the synthesis output for gaps and re-delegate targeted queries until coverage is sufficient
 - E Stop after the first synthesis pass to minimize latency, regardless of whether any gaps were found in the output

Answer: A, D

Partitioning scope reduces duplicated work, and evaluating the synthesis for gaps with targeted re-delegation improves coverage. Sending the same broad query to every subagent duplicates effort, and letting each subagent set its own scope abandons the coordinator's role. Stopping at the first pass ignores known gaps.

6. Where should the key findings summary go in a long aggregated input so that the model is less likely to miss it?
- A Buried in the middle of the input, between two long result sections, so that it sits in context with the findings it summarizes
 - B Repeated inside every section, so that no part of the input is ever positioned in a middle where it could be overlooked
 - C Written as a single sentence with no headers, since section headers add tokens that the model is known to ignore
 - D At the beginning of the aggregated input, with the detailed results organized under explicit section headers

Answer: D

Models process the beginning and end of long inputs more reliably than the middle, so the key findings belong at the start, with explicit headers guiding the detail. Placing the summary in the middle is the risky position the objective warns about. Repetition and removing headers do not address the position effect.

7. Skills in `.claude/skills/` are defined in a `SKILL.md` file, and its _____ frontmatter option runs the skill in an isolated sub-agent context.

Answer: context: fork (also accepted: `context:fork`, `fork`)

The `context: fork` frontmatter option isolates a skill's output from the main conversation, which helps when the skill produces verbose work. Other options such as `allowed-tools` and `argument-hint` configure different aspects of the skill. The blank accepts `context: fork`, `context: fork`, or `fork`.

8. Which situation most favors a fresh session with an injected summary rather than resuming an earlier session?
- A The prior tool results are stale because the underlying files have changed since those results were read
 - B The team wants every earlier reasoning step to stay visible for an audit that is scheduled to begin next month
 - C The earlier session was given a name, so resuming it by that name would be ambiguous and could load the wrong session
 - D The task is a single-file bug fix with a clear stack trace, which means no earlier context has any bearing on it

Answer: A

When earlier tool results no longer reflect the code, resuming can mislead the agent, so a fresh session with a summary of valid findings is more reliable. Keeping reasoning for an audit is not what session resumption is for. A single clear bug fix is better handled by direct execution, which is a separate decision.

9. A strict JSON schema used with tool use prevents semantic errors, such as line items that do not sum to the stated total.
- A True
 - B False

Answer: B (False)

A strict schema eliminates syntax errors, but it does not check that values are consistent with each other. Totals that do not match their line items are semantic errors and need explicit validation. The statement is false.

10. Which error metadata best helps an agent avoid wasted retries on a business rule violation?
- A isRetryable set to true on every failure, so that the agent keeps trying until the call eventually succeeds
 - B The raw database exception with no category, so that the agent can classify the problem itself on the next turn
 - C isRetryable set to false, with a customer-friendly explanation of the policy that was violated
 - D An HTTP 200 status with the error text hidden in an optional field that the agent is unlikely to read at any point

Answer: C

A non-retryable flag plus a customer-friendly explanation stops the agent from repeating a call that policy will never allow, and lets it explain the outcome. Marking every failure retryable causes wasted attempts, and raw exceptions or hidden errors leave the agent to guess. Structured metadata is what prevents the guessing.