

PRACTICE QUESTIONS

Anthropic

CCDV-F

Claude Certified Developer - Foundations

SHARED BY [u/Silver-Kettle-9325](#) on aiexamcerts

EDITION Free Set 1

QUESTIONS 10

SHARED October 18, 2026

EXAM Anthropic CCDV-F, Claude Certified Developer - Foundations

Shared by [u/Silver-Kettle-9325](#) on aiexamcerts

10 practice questions for the Anthropic CCDV-F exam, Claude Certified Developer - Foundations, each followed by its answer and a short explanation. More free sets and the full practice exam: aiexamcerts.com/anthropic/ccdv-f. Spotted a mistake? Email support@aiexamcerts.com with the exam code CCDV-F and the question number.

1. What is the role of a coordinator in a manager and supervisor hierarchy?
 - A Execute every tool itself, so that subagents never run any tool calls at any point in the workflow
 - B Decompose the task, delegate parts to subagents, and combine their results into one answer for the user
 - C Store every subagent's output in a database and never pass any result back to the user or to the other agents
 - D Choose the model's temperature for each subagent and leave every other decision in the workflow to the subagents

Answer: B

A coordinator breaks the work into parts, assigns them, and aggregates the results. Running every tool itself or leaving subagents with no routing defeats the purpose of the hierarchy. Setting temperature is a configuration detail, not the coordinator's role.

2. In Claude Code, the file that holds project-wide instructions and is committed so the whole team shares it is named _____.

Answer: CLAUDE.md (also accepted: CLAUDE.md file)

A project-level CLAUDE.md is committed to version control, so every developer and every session in that project receives the same standing instructions. A user-level file applies only to one person. The blank accepts CLAUDE.md, CLAUDE.md file, or cLaude.md.

3. Why does a long conversation eventually become more expensive and less reliable?
 - A The model charges per character the user types, so a long conversation accumulates the cost of every keystroke in the chat
 - B The API deletes older messages automatically, which forces the model to guess what was said earlier in the chat
 - C Long conversations lower the temperature of the model, so responses become more repetitive and less useful over time
 - D Each turn resends the growing context, and the context window limits how much earlier material the model can use at once

Answer: D

Each request carries the accumulated history, so input tokens grow with the conversation, and the context window bounds what the model can attend to. Billing is per token, not per keystroke. Automatic deletion and temperature changes are not how the conversation behaves.

4. A requirement says each request must return an answer within one minute. Which API choice follows from that?
- A The synchronous Messages API, since the Message Batches API processing window is not suitable for answers needed within a minute
 - B The Message Batches API, since its discount is the main priority whenever a minute of latency is an acceptable requirement
 - C A nightly export of all requests, since results are delivered together and the customer can read them the next morning
 - D A file upload per request, since uploaded files are always answered faster than inline messages in every situation

Answer: A

A per-request time requirement calls for the synchronous Messages API, because batches are asynchronous within a window of up to 24 hours. Choosing batches for speed ignores the requirement. A nightly export and file uploads do not meet a one-minute answer.

5. Which two practices help debug a Claude application that gives wrong answers in production? (Choose two.)
- A Review traces of the full request and response for failing cases to find where the failure starts
 - B Change the model and the prompt at the same time, so that any improvement is picked up as quickly as possible
 - C Delete the logs so that old failures do not confuse the new investigation into the current behavior
 - D Isolate whether the problem is in the integration code or in the model's output, using a minimal reproduction
 - E Assume the model is at fault and stop checking the integration layer code, since the model is the usual source

Answer: A, D

Traces of the full request and response show where a failure begins, and a minimal reproduction separates integration bugs from model behavior. Changing several things at once hides the cause, and deleting logs destroys evidence. Assuming the model is always at fault skips the integration layer that often causes errors.

6. Which tool implementation practice helps the model recover from a failed tool call?
- A Return an empty object with no error text, so that the model assumes the call worked and continues the task without interruption
 - B Crash the application process so that the model is forced to start a new session with fresh context on the next request
 - C Return a clear error message that says what went wrong and which input to change, so the model can correct the call and retry
 - D Return only the raw HTTP headers, since the model parses headers more reliably than it parses any ordinary message text

Answer: C

A clear, actionable error lets the model explain the problem or make a corrected call. Returning an empty success hides the failure, and crashing the process or returning headers only gives the model nothing useful to act on. Recovery depends on informative errors.

7. The Messages API returns responses as plain text, so a developer does not need to parse any JSON to read the reply.
- A True
 - B False

Answer: B (False)

The Messages API returns responses in a structured JSON format with content blocks and metadata such as `stop_reason`, so developers parse JSON to read them. Plain text is only the content inside those blocks. The statement is false.

8. Prompt caching is most useful when the same long prompt prefix is _____ across many requests.

Answer: reused (also accepted: repeated, reused across requests)

Caching pays off when a large, unchanging prefix is reused by many requests, so it is processed once and read cheaply afterward. Content that changes on every call gets little benefit. The blank accepts reused or repeated.

9. Which description of the Claude Agent SDK fits best?

- A** A hosted spreadsheet product that stores Claude outputs as formulas so that they can be analyzed later on
- B** A toolkit for building agents with Claude, covering the agent loop, tool use, hooks, and session handling
- C** A browser extension that replaces the standard chat interface with a fixed set of pre-written answers for users
- D** A database engine that stores prompts and retrieves the nearest stored prompt for each new question a user asks

Answer: B

The Claude Agent SDK provides the building blocks for agents, including the agent loop, tools, hooks, and session management. The other descriptions name products or tools that the SDK is not. Recognizing its scope is the point of this question.

10. A nightly job must produce results by the next morning and should minimize cost. Which API fits?

- A** A synchronous call per document running in parallel, which completes faster and is therefore the cheapest approach overall
- B** A streaming session that stays open all night, so that partial results can be read as soon as they arrive during the run
- C** The largest model on every request, since larger models reduce the total number of requests the job has to send to finish
- D** The Message Batches API, which processes requests asynchronously within a window of up to 24 hours at a reduced cost

Answer: D

The Message Batches API suits non-urgent, high-volume work at reduced cost, and the window fits an overnight job. Parallel synchronous calls cost more per request, and streaming does not reduce cost. Using the largest model increases cost rather than reducing it.