

PRACTICE QUESTIONS

Anthropic

CCDV-F

Claude Certified Developer - Foundations

SHARED BY [u/Happy-Bison-4304](#) on aiexamcerts

EDITION Free Set 2

QUESTIONS 10

SHARED October 18, 2026

EXAM Anthropic CCDV-F, Claude Certified Developer - Foundations

Shared by [u/Happy-Bison-4304](#) 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 main benefit of few-shot prompting?

- A It shows the model the pattern you want through worked examples, which often improves the consistency of format and judgment
- B It makes the model fully deterministic, so that each request returns exactly the same output every time it is run
- C It removes the need for a system prompt, since the examples replace every instruction the application would otherwise give
- D It lowers the context window size so that each request costs less to send and receive from the API

Answer: A

Worked examples show the desired pattern, which tends to make output more consistent in format and in handling ambiguous cases. Few-shot prompting does not make output deterministic, and examples add tokens rather than shrinking the window. Examples complement instructions rather than replacing them.

2. CLAUDE.md files can exist at the user, project, and subdirectory levels, and the instructions from these levels apply according to where Claude is working.

- A True
- B False

Answer: A (True)

Claude Code reads CLAUDE.md files from the user level, the project level, and subdirectories, so instructions depend on the location where work is happening. This hierarchy lets teams share project rules while keeping personal preferences separate. The statement is true.

3. Which practice is best for managing prompts as an application evolves?
- A Keep only the latest prompt in the chat history and overwrite older versions, so that the repository stays small and easy to browse
 - B Store each prompt inside the model's system message, so that no separate version control of prompts is needed at all
 - C Version prompts alongside the code, and record which prompt version produced each output so that changes can be traced and rolled back
 - D Rename the prompt every time it changes, so that no two prompts ever share a name in any environment the team uses

Answer: C

Versioning prompts with the code and recording which version produced each output makes changes reviewable and reversible. Overwriting older prompts destroys the history needed for comparison. Storing prompts in a system message or renaming them does not provide traceability.

4. When would extended thinking be worth its extra cost and latency?
- A On a simple lookup that returns one field from a database and needs a reply in a few milliseconds at most
 - B On a hard multi-step problem where reasoning quality matters more than a fast reply
 - C On every request by default, since thinking is always free and never changes the answer the model gives the user
 - D On tasks that require no reasoning at all, such as copying a value from one line of text to another line of the document

Answer: B

Extended thinking spends more compute on hard problems, which is worth it when quality matters more than speed. It adds latency and cost, which a simple lookup or a copy task does not need. Using it by default for every request wastes budget.

5. Which two statements about MCP are accurate? (Choose two.)
- A MCP servers run inside the model and cannot be deployed separately from the application that uses them
 - B An MCP server can expose tools, resources, and prompts to a Claude application
 - C MCP is a proprietary format that only one vendor's model is able to read, which limits it to that vendor
 - D MCP servers cannot expose data, only actions, so resources are not part of the protocol at all
 - E stdio is one transport that an MCP server can use to communicate with its client

Answer: B, E

MCP servers expose tools, resources, and prompts, and stdio is one of the transports used between a client and a server. MCP servers run separately from the model, and the protocol is not limited to a single vendor. Resources are an explicit part of the protocol, so the claim that they are absent is wrong.

6. What is the purpose of the agent loop?

- A Send one request and end immediately, since looping adds cost without changing the answer the user receives at all
- B Run the same tool on every iteration regardless of what the model asks for, so that the results stay consistent
- C Keep the conversation open forever so that the user can add new messages whenever they choose during the session
- D Repeatedly call the model, run the tools it requests, feed the results back, and stop when the model signals that it is done

Answer: D

The agent loop alternates between model calls and tool execution, feeding results back so the model can take the next step, and ends when the model signals completion. A single request cannot use tool results, and running the same tool regardless of the request ignores the model's decisions. An endless session is not a loop with a defined end.

7. The _____ prompt is where standing instructions for the whole application belong, while each user's request goes in the user turn.

Answer: system

The system prompt sets the standing behavior for the application, and user messages carry the individual requests. Placing instructions in the right component helps the model keep them in effect across turns. The blank accepts system or System.

8. A web page a user submitted contains hidden text that tells the model to reveal its system prompt. What is the most effective defense?

- A Treat retrieved content as untrusted data, keep it separate from instructions, and use guardrails so injected text cannot trigger sensitive actions
- B Raise the temperature so that the model becomes less predictable and therefore less likely to follow the injected text reliably
- C Ask users in the system prompt not to include malicious instructions, since the model will then ignore any such text it finds
- D Switch to the largest model available, because bigger models follow instructions more faithfully and resist injection better

Answer: A

Prompt injection is countered by isolating untrusted content from trusted instructions and enforcing guardrails so that injected text cannot trigger sensitive actions. Temperature does not address injection, and a polite request is not an enforceable control. A larger model that follows instructions well can be more susceptible, not less.

9. A project-level `settings.json` file can hold Claude Code configuration, such as permissions and environment settings, and it can be committed so the team shares it.
- A True
 - B False

Answer: A (True)

Claude Code reads `settings.json` files at several levels, and a project-level file can be committed so that the team shares permissions and environment configuration. Personal settings belong at the user level. The statement is true.

10. A team wants to forecast monthly spend on a Claude feature. What should it track?
- A The number of words in the marketing copy for the feature, since copy length is what determines the cost of each call
 - B The number of developers who have viewed the documentation, since documentation views are what drive API usage
 - C Token usage per request, split into input and output and multiplied by the rates for the chosen model
 - D The temperature setting across requests, since a higher temperature always means higher billing for the same work

Answer: C

Billing follows input and output tokens at the rates for the model in use, so tracking token counts per request is the basis of a cost forecast. Marketing copy, documentation views, and temperature do not determine what the API charges. Measuring usage is what makes the forecast accurate.