KTest

高品質　より良いサービス

練習問題

https://www.ktest.jp

1年間無料アップデート

Exam : **CCDV-F**

Title : Claude Certified Developer - Foundations

Version : DEMO

1.You are designing a Claude application that will require structured JSON output for downstream processing. The output schema is well-defined, and downstream systems will reject malformed JSON.
A. Structure the prompt to request output in a schema that is described in plain English, with downstream systems parsing whatever shape Claude produces.
B. Define a clear schema and structure the prompt to request output in that schema, with downstream systems handling any validation needed.
C. Define a clear schema, structure the prompt to request output in that schema, and validate Claude's output against the schema before passing it downstream.
D. Avoid structured output and use free-form text everywhere instead, on the grounds that free-form text is more flexible and handles edge cases better than structured schemas.
Answer: C

2.Your Claude application has multi-step workflows where each step's output is needed only briefly before the agent moves on. The cumulative tool output is filling the context window with content that is no longer relevant.
How would you handle the accumulating tool output?
A. Apply tool output pruning to remove tool outputs that are no longer needed by later steps in the workflow.
B. Apply prompt caching to the accumulated tool outputs so the application does not re-pay for the older content on each subsequent step.
C. Switch to a smaller Claude model that processes context more efficiently and treat any quality loss as a tradeoff for the cost reduction.
D. Keep every tool output in the context indefinitely so the agent has the full record of every step it has executed during the workflow.
Answer: A

3.A teammate has asked you to explain the difference between context engineering and prompt engineering. They have heard the terms used interchangeably and are unsure how each applies to a Claude application that processes long-running multi-step tasks.
How would you describe the distinction?
A. Prompt engineering focuses on the model's response, while context engineering focuses on the user's input across many sessions in a long-running multi-step Claude application.
B. Prompt engineering is the older term for prompt design, while context engineering is the newer term that has replaced it in modern Claude applications across the industry.
C. Prompt engineering shapes individual prompts for specific outputs, while context engineering manages how content flows across turns and steps and takes steps to keep relevant state visible.
D. Prompt engineering and context engineering each address content the team gives Claude, but the team can group them under a single workflow because the practices use overlapping techniques.
Answer: C

4.You are designing a multi-step Claude workflow where some steps must reason without seeing the full prior conversation history. The team wants to keep specific context isolated to specific steps.
The context engineering technique you would use is...
A. Augmenting the context with all available content at every step so each step has access to the entire

prior history of the workflow during its reasoning.
B. Using a single global prompt that applies to every step in the workflow no matter what each step is reasoning about during its run.
C. Context isolation through subagents or multi-step agentic workflows that scope each step's context to only what the step needs.
D. Embedding the full prior history in each step regardless of whether the step needs the prior history for its reasoning.
Answer: C

5.Your Claude application produces good responses for typical inputs but struggles with edge cases. You have several labeled examples of edge-case inputs and the desired response for each. You want to use these examples to improve the model's handling of edge cases.
What is the best way to use these examples?
A. Embed the examples in a database for the model to find during inference.
B. Add the labeled edge-case examples to the prompt as few-shot examples so the model can learn the pattern.
C. Train a custom model on the edge-case examples and deploy that custom model in place of the team's current Claude integration.
D. Tell users to avoid submitting the edge-case inputs to the application by adding warnings in the application's user interface.
Answer: B