

PRACTICE QUESTIONS

NVIDIA

NCP-AAI

NVIDIA-Certified Professional: Agentic AI

SHARED BY [u/steady_switch_33](#) on aiexamcerts

EDITION	Free Set 1
QUESTIONS	10
SHARED	October 15, 2026
EXAM	NVIDIA NCP-AAI, NVIDIA-Certified Professional: Agentic AI

Shared by [u/steady_switch_33](#) on aiexamcerts

10 practice questions for the NVIDIA NCP-AAI exam, NVIDIA-Certified Professional: Agentic AI, each followed by its answer and a short explanation. More free sets and the full practice exam:

www.aiexamcerts.com/nvidia/ncp-aii. Spotted a mistake? Email support@aiexamcerts.com with the exam code NCP-AAI and the question number.

-
1. Which deployment approach best suits an agent's LLM endpoint that must serve many concurrent users on NVIDIA GPUs?
- A A single laptop running the model with a sleep timer
 - B NVIDIA NIM with a serving stack that supports batching and horizontal scaling behind a load balancer
 - C Hard-coding one GPU to one user session
 - D Running each request on a fresh container that loads the model from remote object storage each time it is called

Answer: B

NIM provides optimized inference containers, and batching plus horizontal scaling let the endpoint handle concurrency efficiently. Cold loading per request adds large latency, which is the tempting but costly pattern.

-
2. _____ memory stores the outcomes of past tasks so the agent can reuse successful plans on similar future tasks.

Answer: Episodic (also accepted: Experience)

Episodic memory records specific past episodes such as previous tasks and outcomes for later reuse. Semantic memory holds general facts instead, which is the closest alternative and therefore a tempting but different answer.

-
3. A single-agent design is failing because one prompt must handle planning, retrieval, code execution and customer messaging. What is the most sound first change?
- A Increase the temperature so the single agent explores more options
 - B Split responsibilities into specialized agents or roles with explicit handoff contracts
 - C Remove the tool list so the model focuses on writing answers as plain text with no structure
 - D Double the context window and paste every document into one prompt

Answer: B

Decomposing the work into roles with defined inputs and outputs makes each step testable and limits prompt conflicts. Raising temperature increases variance rather than capability, which is why it is a tempting but weak fix.

4. A tool that changes data can be retried safely only if the tool is idempotent or accepts a deduplication key.
- A True
 - B False

Answer: A (True)

Retries after a timeout can run a side effect twice unless the tool recognizes the repeated request. Idempotency keys make the retry safe because the second call returns the first result.

5. Which two statements about agent state management are correct? (Choose two.)
- A Session state should be stored outside the process so any worker can resume a conversation
 - B Checkpointing the task state after each step allows resumption after a crash
 - C Keeping state only in a Python global variable is a safe default for production traffic
 - D Each request should reset all state so the agent cannot learn from previous steps
 - E State must always be kept inside the model weights

Answer: A, B

External, durable session storage lets stateless workers serve any turn, and per-step checkpoints let a crashed run resume. Process-global variables break as soon as there is more than one worker, which is why that option is tempting but wrong.

6. What is the main benefit of trajectory-level evaluation for an agent?
- A It replaces all end-to-end tests with step-level checks only
 - B It inspects each step's tool choice, arguments and results, showing why a failure happened
 - C It only measures GPU utilization and memory use during each run
 - D It removes the need for logging because the model always reports its own decisions faithfully

Answer: B

Trajectory evaluation checks the sequence of decisions and results, so teams can locate the failing step. End-to-end scores alone show failure without the cause, which is why trajectory checks complement them.

7. Which statement about a shared scratchpad or blackboard in multi-agent systems is most accurate?
- A It removes the need for any agent to report its results to the supervisor agent
 - B It guarantees that all agents agree on every fact stored in the shared state after each update
 - C A GPU memory-sharing feature of the CUDA driver for processes on one node
 - D It gives agents a common state to read and write, but needs ownership rules against overwrites

Answer: D

A shared state simplifies coordination, but concurrent updates need ownership, locking or versioning to avoid silent overwrites. Agreement is not guaranteed by sharing state, which is why that claim is a tempting error.

8. _____ is the practice of deliberately trying to make an agent fail or misbehave before release, so the weaknesses can be fixed.

Answer: Red teaming (also accepted: Red-teaming, Adversarial testing)

Red teaming probes the system with attacks and edge cases to find weaknesses before users do. Regular regression testing is a related activity, but it checks known behavior rather than actively searching for new failures.

9. A long-running agent starts losing early instructions as its conversation grows. Which strategy is the most sound?
- A Summarize older turns into a compact state and keep critical constraints pinned in the prompt
 - B Delete the system prompt to make room
 - C Increase temperature so older text is remembered
 - D Repeat the full transcript at every turn without any limit, including all raw tool outputs and errors

Answer: A

Summaries and pinned constraints preserve what matters while staying within the window. Repeating the full transcript grows the cost until it breaks the window, which is the tempting but unsustainable approach.

10. What does the application do after a model emits a function call in a standard tool-calling loop?
- A** Sends the arguments back to the model without executing anything
 - B** Retrains the model on the arguments before continuing
 - C** Stops the conversation and asks the user to write the code
 - D** Validates the arguments, executes the function, and returns the result to the model as a tool message

Answer: D

The model proposes the call, but the application is responsible for validating and executing it, then feeding the result back as a tool message. Returning arguments without execution is a plausible misunderstanding, since the model never runs code itself.