📘 MCQs: Hooks (RunHooks & AgentHooks) – Fundamentals of Agentic AI

**1. What are hooks in Agentic AI?**

A) Retry loops for failed calls  
B) Functions or callbacks that run at specific lifecycle points of an agent or run  
C) System prompts  
D) Token counters

**Answer:** B  
**Explanation:** Hooks are **customizable callbacks** that let developers monitor, modify, or extend agent behavior.

**2. What do RunHooks primarily operate on?**

A) Individual spans  
B) The lifecycle of an **entire run**  
C) Static prompts  
D) Tokens

**Answer:** B  
**Explanation:** RunHooks attach to the **execution lifecycle of a full agent run**.

**3. What do AgentHooks primarily operate on?**

A) The lifecycle of the **agent itself** (reasoning steps, tool calls, decisions)  
B) Retry handlers only  
C) System prompts  
D) Post-processing

**Answer:** A  
**Explanation:** AgentHooks provide visibility into **agent decision-making and actions**.

**4. Which of these is an example of a RunHook?**

A) on\_run\_start  
B) on\_agent\_think  
C) on\_tool\_call  
D) on\_output\_parse

**Answer:** A  
**Explanation:** on\_run\_start is a RunHook that fires **when a run begins**.

**5. Which of these is an example of an AgentHook?**

A) on\_agent\_action  
B) on\_run\_end  
C) on\_run\_retry  
D) on\_output\_filter

**Answer:** A  
**Explanation:** on\_agent\_action is an AgentHook triggered when the agent **chooses an action/tool**.

**6. Why are hooks useful?**

A) To enforce guardrails only  
B) To add **custom monitoring, logging, and intervention** points  
C) To reduce token counts  
D) To disable retries

**Answer:** B  
**Explanation:** Hooks let developers **observe and control** agent lifecycles.

**7. Which hook type is better for system-wide observability?**

A) RunHooks  
B) AgentHooks  
C) Guardrails  
D) Retry loops

**Answer:** A  
**Explanation:** RunHooks track **full run-level events** across the system.

**8. Which hook type is better for fine-grained decision tracking?**

A) RunHooks  
B) AgentHooks  
C) Traces  
D) Guardrails

**Answer:** B  
**Explanation:** AgentHooks track **agent reasoning, tool use, and outputs**.

**9. When does on\_run\_end fire?**

A) When agent selects an action  
B) When the run finishes successfully or with error  
C) Before any tool call  
D) At model initialization

**Answer:** B  
**Explanation:** on\_run\_end captures **final run completion** state.

**10. Which of the following is a valid use case of AgentHooks?**

A) Debugging why an agent chose a tool  
B) Counting total runs per session  
C) Resetting retry logic  
D) Changing temperature

**Answer:** A  
**Explanation:** AgentHooks expose **agent reasoning steps** for debugging.

**11. Which of the following is a valid use case of RunHooks?**

A) Adding global logging at the start and end of each run  
B) Inspecting agent step-by-step reasoning  
C) Blocking specific tool calls  
D) Enforcing retry limits

**Answer:** A  
**Explanation:** RunHooks apply **at the entire run level**, useful for global logging.

**12. What happens if a hook raises an error?**

A) Execution always stops  
B) It may propagate, or can be caught depending on implementation  
C) Errors are ignored  
D) Hook disables itself

**Answer:** B  
**Explanation:** Hook errors can **stop runs** or be **safely caught** by the framework.

**13. Which hook type would best capture metrics like latency per run?**

A) RunHooks  
B) AgentHooks  
C) Guardrails  
D) Spans

**Answer:** A  
**Explanation:** RunHooks record **run-level metrics** like duration.

**14. Which hook type would best capture a specific tool call’s parameters?**

A) RunHooks  
B) AgentHooks  
C) RetryHooks  
D) Guardrails

**Answer:** B  
**Explanation:** AgentHooks can **inspect and log tool call inputs/outputs**.

**15. Which lifecycle phase is unique to RunHooks but not AgentHooks?**

A) on\_run\_start/on\_run\_end  
B) on\_agent\_action  
C) on\_tool\_call  
D) on\_agent\_think

**Answer:** A  
**Explanation:** RunHooks cover **start and end of the whole run**.

**16. Which lifecycle phase is unique to AgentHooks but not RunHooks?**

A) on\_agent\_think/on\_agent\_action  
B) on\_run\_start  
C) on\_run\_end  
D) on\_retry

**Answer:** A  
**Explanation:** AgentHooks focus on **decision-making steps**.

**17. How do hooks relate to tracing?**

A) Hooks generate structured trace events like spans  
B) Hooks disable tracing  
C) Hooks replace tracing  
D) Hooks only apply after tracing

**Answer:** A  
**Explanation:** Hooks often **feed data into tracing systems** for observability.

**18. What’s a best practice for using hooks in production?**

A) Avoid them entirely  
B) Use hooks for monitoring, logging, and injecting safety controls  
C) Only log errors, ignore normal flow  
D) Disable guardrails when hooks are active

**Answer:** B  
**Explanation:** Hooks add **observability and safety** in production systems.

**19. Which hook type could implement a tripwire-style safety check during agent reasoning?**

A) AgentHooks  
B) RunHooks  
C) Retry logic  
D) Token guard

**Answer:** A  
**Explanation:** AgentHooks let developers **inspect reasoning mid-execution**.

**20. Which hook type could implement a global retry strategy after a failed run?**

A) RunHooks  
B) AgentHooks  
C) Guardrails  
D) ToolHooks

**Answer:** A  
**Explanation:** RunHooks can enforce **retry/fallback logic at run-level**.