📘 MCQs: Tool Calls & Error Handling – Fundamentals of Agentic AI

**1. What does a tool call mean in Agentic AI?**

A) When the agent executes free text  
B) When the agent invokes an external function, API, or service  
C) When the system logs errors  
D) When the agent changes temperature

**Answer:** B  
**Explanation:** Tool calls happen when an **agent uses an external resource** (e.g., API, database, function).

**2. Why are tool calls essential in Agentic AI?**

A) They reduce token usage  
B) They allow agents to perform **real-world actions** beyond text generation  
C) They replace system prompts  
D) They disable error handling

**Answer:** B  
**Explanation:** Tool calls let agents **go beyond language** by interacting with external systems.

**3. Which SDK feature wraps Python functions for tool use?**

A) AgentRunner  
B) function\_tool  
C) BaseModel  
D) RunConfig

**Answer:** B  
**Explanation:** function\_tool converts Python functions into **tools callable by agents**.

**4. What is a common default output when a tool call succeeds?**

A) Free text  
B) Structured data (validated output)  
C) Empty string  
D) Error message

**Answer:** B  
**Explanation:** Tool calls typically return **structured/validated outputs** using schemas.

**5. Which of the following is a risk of tool calls?**

A) Token overflow  
B) Invalid inputs causing runtime errors  
C) Automatic alignment failures  
D) Too much creativity

**Answer:** B  
**Explanation:** Tool calls may fail if agents provide **invalid inputs** (e.g., wrong types).

**6. What is the first step in tool call error handling?**

A) Logging  
B) Retrying with corrected inputs  
C) Default fallback response  
D) Returning to user

**Answer:** A  
**Explanation:** Errors are first **logged** for debugging and traceability.

**7. Which parameter defines the schema for tool inputs/outputs?**

A) output\_type  
B) function\_tool  
C) callback  
D) RunConfig

**Answer:** A  
**Explanation:** output\_type enforces **structured input/output validation** for tool calls.

**8. If a tool call fails due to invalid schema, what happens?**

A) Execution halts completely  
B) Validation error is raised and agent may retry  
C) Output defaults to None  
D) Tool is disabled permanently

**Answer:** B  
**Explanation:** Schema validation errors trigger **agent retries or fallbacks**.

**9. Which mechanism allows agents to recover gracefully after tool errors?**

A) Guardrails  
B) Safe fallbacks  
C) Callbacks  
D) Token sampling

**Answer:** B  
**Explanation:** Fallbacks provide **safe defaults** if tool calls fail.

**10. What does retry logic mean in tool execution?**

A) Re-running the agent from scratch  
B) Automatically attempting the tool call again with corrected parameters  
C) Ignoring errors  
D) Skipping validation

**Answer:** B  
**Explanation:** Retry logic helps agents **fix errors and re-call tools automatically**.

**11. Why is error handling critical in Agentic AI?**

A) To reduce creativity  
B) To prevent system crashes and ensure safe, reliable outputs  
C) To block token sampling  
D) To disable callbacks

**Answer:** B  
**Explanation:** Without error handling, tool failures could break the agent.

**12. Which error type occurs when the tool is unavailable?**

A) Validation error  
B) Connection error  
C) Schema error  
D) Retry overflow

**Answer:** B  
**Explanation:** **Connection errors** occur if APIs or external tools are unreachable.

**13. What should agents do when a tool is not found?**

A) Retry infinitely  
B) Return a structured error message or fallback response  
C) Crash the system  
D) Output free text only

**Answer:** B  
**Explanation:** Best practice is **graceful error return**, not failure.

**14. Which mechanism prevents unsafe tool usage?**

A) Callbacks  
B) Guardrails & system messages  
C) Retry loops  
D) Context resets

**Answer:** B  
**Explanation:** **Guardrails** ensure agents don’t misuse tools (e.g., with sensitive data).

**15. In execution, how are multiple tool calls often handled?**

A) Sequentially or in parallel, depending on configuration  
B) Only sequentially  
C) Only in parallel  
D) Random order

**Answer:** A  
**Explanation:** Tools can be executed **sequentially (chain)** or **parallel (batch)**.

**16. If a tool call produces unexpected output type, what happens?**

A) Execution continues without checks  
B) Pydantic/validation raises an error  
C) The output is ignored silently  
D) The agent shuts down

**Answer:** B  
**Explanation:** Schema validation enforces **output type correctness**.

**17. What is the default agent behavior after a failed tool call?**

A) Retry with corrected reasoning  
B) Stop execution permanently  
C) Return free text  
D) Disable the tool

**Answer:** A  
**Explanation:** Agents typically **retry with corrected inputs**.

**18. Which logging feature is useful for debugging tool errors?**

A) AgentHooks  
B) Token counters  
C) Memory buffer  
D) System prompts

**Answer:** A  
**Explanation:** AgentHooks allow **custom logging and debugging** of tool calls/errors.

**19. Which of the following is a real-world example of tool call error handling?**

A) Payment API timeout → return safe fallback "Payment could not be processed"  
B) Tool call → image generation  
C) Prompt formatting  
D) Agent refusing unsafe question

**Answer:** A  
**Explanation:** **Graceful fallback responses** are part of robust error handling.

**20. Best practice for tool calls in production Agentic AI is:**

A) Disable validation for faster runs  
B) Always define schemas, retries, and safe fallbacks  
C) Avoid using tools  
D) Use only free text

**Answer:** B  
**Explanation:** Best practice is to define **clear schemas, retry strategies, and fallbacks** for reliability.