📘 MCQs: Dynamic Instructions & Context Objects – Fundamentals of Agentic AI

**1. What are dynamic instructions in Agentic AI?**

A) Static prompts given at the start  
B) Instructions that adapt during runtime based on agent state or user input  
C) Debug logs  
D) Retry strategies

**Answer:** B  
**Explanation:** Dynamic instructions **change at runtime** depending on conversation history, environment, or context.

**2. Why are dynamic instructions important?**

A) They reduce token size  
B) They allow agents to **adapt behavior flexibly** to user needs and situations  
C) They block unsafe prompts  
D) They replace system messages

**Answer:** B  
**Explanation:** Without dynamic instructions, agents would remain **rigid and less intelligent**.

**3. What are context objects in Agentic AI?**

A) Objects storing structured state information (user, environment, history)  
B) Token counters  
C) Retry logs  
D) JSON error messages

**Answer:** A  
**Explanation:** Context objects hold **structured data** about the conversation/session for reasoning.

**4. Which of the following best describes context objects?**

A) Static system prompts  
B) Memory containers passed to agents  
C) Logging tools  
D) Retry loops

**Answer:** B  
**Explanation:** Context objects act as **memory-like inputs** to guide agent decisions.

**5. Example of dynamic instruction?**

A) “Always say hello.”  
B) “If user asks for weather, use today’s date and location from context.”  
C) “Translate text to French only.”  
D) “Log every response.”

**Answer:** B  
**Explanation:** This is dynamic since it **uses runtime context** (date, location).

**6. How are context objects commonly represented in SDKs?**

A) As raw strings  
B) As Python dictionaries or structured classes  
C) As plain text logs  
D) As retry functions

**Answer:** B  
**Explanation:** Context objects are usually **structured models (dicts, Pydantic classes, dataclasses)**.

**7. Which advantage do dynamic instructions provide?**

A) Smaller tokens  
B) Personalization, adaptivity, and task-awareness  
C) Always deterministic responses  
D) Disabling tool calls

**Answer:** B  
**Explanation:** They enable **personalized and adaptive agent behavior**.

**8. If a user changes their preference mid-conversation, how do dynamic instructions help?**

A) Agent ignores it  
B) Agent updates behavior immediately using updated context  
C) Agent resets itself  
D) Tool calls are disabled

**Answer:** B  
**Explanation:** Dynamic instructions let the agent **adapt on the fly**.

**9. Which component updates context objects during runtime?**

A) Function tools  
B) Runner/agent orchestration  
C) Guardrails  
D) Temperature parameter

**Answer:** B  
**Explanation:** The **runner or orchestration layer** updates context as tasks proceed.

**10. What is a risk of overly dynamic instructions?**

A) Reduced memory use  
B) Confusion, inconsistencies, or prompt injection vulnerabilities  
C) Faster inference  
D) Safer system messages

**Answer:** B  
**Explanation:** If uncontrolled, dynamic instructions can make agents **unreliable or unsafe**.

**11. Which type of data is often stored in context objects?**

A) User preferences, environment state, past tool outputs  
B) Token count only  
C) Logging timestamps  
D) Model parameters

**Answer:** A  
**Explanation:** Context stores **structured memory** for agent reasoning.

**12. How do context objects improve tool usage?**

A) By reducing retries  
B) By providing **inputs (like user ID, location, or API keys)** automatically to tools  
C) By disabling tool calls  
D) By formatting logs

**Answer:** B  
**Explanation:** Context ensures tools get the **right parameters dynamically**.

**13. Which SDK feature is often paired with context objects for validation?**

A) Callbacks  
B) Pydantic models  
C) Logging hooks  
D) Temperature

**Answer:** B  
**Explanation:** Pydantic ensures **context data is valid and structured**.

**14. What happens if a context object is missing required fields?**

A) Agent ignores the error  
B) Validation error is raised, preventing unsafe execution  
C) Default values replace them automatically  
D) Conversation restarts

**Answer:** B  
**Explanation:** Missing data triggers **validation errors** for safety.

**15. Which is a practical application of dynamic instructions?**

A) A travel assistant adapting to new destination requests mid-conversation  
B) Fixed Q&A system  
C) Token limiter  
D) Retry handler

**Answer:** A  
**Explanation:** Travel assistants need **adaptive dynamic instructions**.

**16. In error handling, how can context objects be useful?**

A) By storing previous error states and retry counts  
B) By disabling retries  
C) By logging only tokens  
D) By ignoring errors

**Answer:** A  
**Explanation:** Context helps track **execution history**, improving retries.

**17. Which feature enables safe merging of dynamic instructions with static prompts?**

A) Guardrails and structured system messages  
B) Temperature  
C) Retry fallback  
D) Top-k sampling

**Answer:** A  
**Explanation:** Guardrails ensure **safe integration** of dynamic instructions.

**18. Which type of dynamic instruction supports role adaptation?**

A) "You are always a teacher."  
B) "Switch between teacher and coder mode depending on context object role field."  
C) "Never translate text."  
D) "Always log requests."

**Answer:** B  
**Explanation:** This instruction **adapts role dynamically**.

**19. How can context objects help with multi-turn conversations?**

A) By discarding history  
B) By storing dialogue history, preferences, and tool outputs  
C) By resetting model state  
D) By disabling tools

**Answer:** B  
**Explanation:** Context maintains **conversation continuity**.

**20. Best practice for dynamic instructions & context objects in production?**

A) Avoid validation for speed  
B) Always validate, update safely, and guard against injection  
C) Keep everything static  
D) Disable tools

**Answer:** B  
**Explanation:** The best practice is **validated, secure, adaptive usage** of context + instructions.