🐍 Python / Agentic AI Exam – Agents (Basic)

100 Questions & Answers

Definition & Core Concepts

1.

Q: What is an Agent in Agentic AI?
A: An entity that uses a model + configuration + tools to process inputs and generate outputs.

2.

Q: Which 3 main parts typically define an Agent?
A: Model, configuration, and tools.

3.

Q: True or False: An Agent can only use one tool at a time.
A: False – Agents can use multiple tools.

4.

Q: What makes an Agent “intelligent”?
A: Its ability to reason with context, decide when to use tools, and adapt outputs.

5.

Q: Which component allows an Agent to interact with external APIs or functions?
A: Tools.

6.

Q: What role does “context” play for an Agent?
A: It provides history and additional information to improve reasoning.

7.

Q: Is an Agent tied to a single task or multi-tasking?
A: Multi-tasking – Agents can handle multiple tasks with reasoning.

8.

Q: What is the difference between an Agent and a plain LLM call?
A: An Agent is structured (configured with tools/context/rules), while a plain LLM call is unstructured text prediction.

9.

Q: Which module typically runs and controls Agents?
A: Runner.

10.

Q: True or False: Agents can hand off tasks to other agents.
A: True.

Agent Lifecycle

11.

Q: What is the first step when creating an Agent?
A: Define its model and configuration.

12.

Q: After defining, what step comes next for an Agent?
A: Attach tools.

13.

Q: What is the final step before execution of an Agent?
A: Run/initialize with Runner or execution controller.

14.

Q: Which function allows an Agent to start processing?
A: agent.run() or via Runner.

15.

Q: Can an Agent persist state across runs?
A: Yes, if designed with memory/context.

Agent vs Other Components

16.

Q: Difference between Agent and Tool?
A: Agent = orchestrator/decision-maker; Tool = external function/API used by Agent.

17.

Q: Difference between Agent and Runner?
A: Agent = logic; Runner = execution controller.

18.

Q: Which manages flow: Agent or Runner?
A: Runner manages execution flow; Agent manages reasoning.

19.

Q: Which is higher-level: Agent or Tool?
A: Agent.

20.

Q: Do Agents replace Tools?
A: No, they depend on Tools.

Agent Configuration

21.

Q: Which parameter defines model reasoning randomness?
A: Temperature.

22.

Q: Which config controls selection of tokens?
A: top_p / top_k.

23.

Q: Which config prevents repetition?
A: Penalties.

24.

Q: Which config defines tool selection?
A: tool_choice.

25.

Q: Can an Agent work without explicit configuration?
A: Yes, but performance may degrade.

26.

Q: True or False: Config is optional.
A: True, defaults exist.

27.

Q: Which part of config ensures safety?
A: Guardrails (basic/medium topic).

28.

Q: Which config decides delegation?
A: Handoff.

29.

Q: Which config ensures multiple tools can be chained?
A: tool_choice + runner control.

30.

Q: Where is execution limit defined?
A: Runner (max_turns).

Agent Input & Output

31.

Q: What type of input does an Agent take?
A: Natural language or structured input.

32.

Q: What type of output does an Agent give?
A: Natural language or structured (JSON/text).

33.

Q: True or False: Agents can only output text.
A: False – can output structured data.

34.

Q: What happens if an Agent has no tool access?
A: It falls back to reasoning with only the model.

35.

Q: Can Agents handle streaming input?
A: Yes, with streaming enabled (medium level).

Agent Examples

36.

Q: Example: A travel Agent that books flights uses what?
A: Tools (API) + reasoning.

37.

Q: A coding Agent that debugs uses what tools?
A: Code executor, search, debugger.

38.

Q: A trading Agent that buys crypto depends on?
A: Trading API tool.

39.

Q: A student Q&A Agent mainly uses?
A: Context + model reasoning.

40.

Q: Which Agent type handles multiple domains?
A: Generalist Agent.

Decision-making

41.

Q: How do Agents decide when to use a tool?
A: Based on model reasoning + tool_choice.

42.

Q: Which step decides final output?
A: Post-reasoning after tool calls.

43.

Q: Can Agents decline requests?
A: Yes, if safety constraints/guardrails apply.

44.

Q: What if tool fails?
A: Agent retries or returns error (depends on config).

45.

Q: Who controls retries?
A: Runner or error handler.

Agent Roles

46.

Q: What is a Specialist Agent?
A: An Agent focused on one domain/task.

47.

Q: What is a Generalist Agent?
A: An Agent handling multiple domains.

48.

Q: Which type is more efficient?
A: Specialist Agent (domain optimized).

49.

Q: Which type is more flexible?
A: Generalist Agent.

50.

Q: True or False: Generalist always performs better.

A: False.

Agent Collaboration

51.

Q: How do multiple Agents collaborate?
A: Through handoff functions.

52.

Q: What is Agent-to-Agent communication called?
A: Handoff / Delegation.

53.

Q: Which controls Agent collaboration?
A: Runner + handoff.

54.

Q: Can one Agent call another as a tool?
A: Yes (Agent as Tool).

55.

Q: Example of collaboration?
A: Research Agent delegating to Math Agent.

Agent Failures

56.

Q: What is a failure in an Agent?
A: Inability to complete a task.

57.

Q: What happens on tool failure?
A: Error handling or fallback.

58.

Q: Which config helps avoid crashes?
A: failure_error_function.

59.

Q: Can Agents self-correct?
A: Yes, by re-prompting or retrying.

60.

Q: Who handles max retry attempts?
A: Runner.

Agent Memory & Context

61.

Q: What is short-term memory in Agents?
A: Context of current run.

62.

Q: What is long-term memory?
A: Persisted storage across sessions.

63.

Q: Which memory allows Q&A conversation continuity?
A: Short-term (context window).

64.

Q: Which memory allows recall of user preferences?
A: Long-term memory.

65.

Q: True or False: Context window is infinite.
A: False – limited by model.

Agent Execution

66.

Q: Who starts execution?
A: Runner or agent.run().

67.

Q: What is max_turns in Runner?
A: Limit of reasoning steps.

68.

Q: Why limit max_turns?
A: To avoid infinite loops.

69.

Q: Which part defines stop conditions?
A: Runner.

70.

Q: Can execution be paused?
A: Yes, with control hooks.

Agent Safety

71.

Q: What ensures safe outputs?
A: Guardrails.

72.

Q: What prevents harmful actions?
A: Config constraints + tool restrictions.

73.

Q: True or False: Agents are inherently safe.
A: False – must be controlled.

74.

Q: Which config disables unsafe tools?
A: is_enabled = False.

75.

Q: Which prevents context leakage?
A: Context filters.

Agent Scalability

76.

Q: Can multiple Agents run in parallel?
A: Yes.

77.

Q: Who orchestrates multiple Agents?
A: Orchestrator or multi-agent runner.

78.

Q: True or False: Scaling Agents reduces performance.
A: False – scaling increases reach but may cost more.

79.

Q: What is load balancing in multi-agents?
A: Distribution of tasks.

80.

Q: Which helps scale?
A: Orchestrating Multiple Agents.

Agent Practical

81.

Q: Example code: agent = Agent(model="gpt", tools=[search]) □ What is created?
A: An Agent with GPT model and search tool.

82.

Q: agent.run("Find weather") □ What happens?
A: Agent decides to call weather tool.

83.

Q: If tool_choice is disabled, what happens?
A: Agent may only reason textually.

84.

Q: If temperature=0, how does Agent behave?
A: Deterministic responses.

85.

Q: If temperature=1, how does Agent behave?

A: More creative/random.

Agent Use Cases

86.

Q: Which industry uses trading Agents?
A: Finance.

87.

Q: Which industry uses medical Agents?
A: Healthcare.

88.

Q: Which uses customer support Agents?
A: E-commerce.

89.

Q: Which uses research Agents?
A: Academia.

90.

Q: Which uses legal Agents?
A: Law firms.

Miscellaneous

91.

Q: Can Agents be cloned?
A: Yes (advanced topic).

92.

Q: What is tool_use_behavior?
A: Defines how Agent selects tools.

93.

Q: Which advanced config resets tools?
A: Reset_tool_choice.

94.

Q: Which allows input validation?
A: input_filter.

95.

Q: Which defines structured outputs?
A: structured JSON schema.

96.

Q: Which logs Agent flow?
A: Tracing.

97.

Q: Which streams outputs live?
A: Streaming.

98.

Q: Can an Agent act as a tool for another?
A: Yes (Agent as Tool).

99.

Q: Who defines custom_output_extractor?
A: Advanced Agent setup.

100.

Q: What is the key role of an Agent in AI systems?
A: To orchestrate reasoning, tool use, and generate intelligent responses.

 That’s 100 Basic-level Questions & Answers only on Agents.

Do you want me to move next on “Agent Configuration” (another 100 Q&A) or should I merge all Basic section topics into one big 300 Q&A set?