□ Python / Agentic AI Exam – Agent Configuration (Basic)

100 Questions & Answers

Intro to Configuration

1.

Q: What is Agent Configuration?
A: The setup of parameters that control how an Agent behaves, reasons, and uses tools.

2.

Q: Which 3 main areas are covered by configuration?
A: Model settings, tool usage, and execution rules.

3.

Q: Why is configuration important?
A: It customizes Agent behavior for specific tasks and performance.

4.

Q: True or False: An Agent can run without configuration.
A: True – it runs with defaults, but less optimized.

5.

Q: Which module stores configuration in most frameworks?
A: Agent’s config object or Runner settings.

Model Settings

6.

Q: Which config controls randomness of outputs?
A: Temperature.

7.

Q: What does top_p control?
A: Probability distribution sampling (nucleus sampling).

8.

Q: What does top_k control?
A: The number of top tokens considered.

9.

Q: Which config prevents repetition?
A: Penalties (frequency, presence penalties).

10.

Q: Which config decides creativity level?
A: Temperature.

11.

Q: What happens when temperature = 0?
A: Deterministic outputs.

12.

Q: What happens when temperature = 1?
A: More diverse outputs.

13.

Q: If top_p = 1, what does that mean?
A: All tokens are considered (no restriction).

14.

Q: If top_k = 1, what does that mean?
A: Only the most likely token is chosen.

15.

Q: Which two configs work together for sampling?
A: top_p and top_k.

Tool Choice Config

16.

Q: What does tool_choice define?
A: Whether the Agent can select tools and how.

17.

Q: If tool_choice = “auto”, what happens?
A: Agent decides automatically which tool to use.

18.

Q: If tool_choice = “none”, what happens?
A: Agent does not use any tool.

19.

Q: If tool_choice = “required”, what happens?
A: Agent must use a tool to answer.

20.

Q: Which config enables/disables specific tools?
A: is_enabled.

21.

Q: Can multiple tools be enabled at once?

A: Yes.

22.

Q: Which config lets an Agent re-select tools mid-run?
A: reset_tool_choice.

23.

Q: What happens if a disabled tool is called?
A: Agent error or fallback.

24.

Q: Who decides final tool execution?
A: Runner executes after Agent’s request.

25.

Q: True or False: Tool configs are global only.
A: False – they can be per-Agent.

Execution Control

26.

Q: What is max_turns in config?
A: Maximum reasoning steps before stopping.

27.

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

28.

Q: Which config sets timeout for execution?
A: Execution timeout.

29.

Q: What happens if max_turns exceeded?
A: Runner stops the Agent.

30.

Q: Which config controls retries?
A: retry policy.

31.

Q: Which config logs each step?
A: Tracing/logging.

32.

Q: Which config ensures deterministic runs?
A: seed + temperature=0.

33.

Q: Which config handles fallback models?
A: failover_model.

34.

Q: What does strict_mode do?
A: Enforces strict schema/tool use.

35.

Q: Can execution be interrupted mid-run?
A: Yes, via Runner control.

Context Handling

36.

Q: Which config decides how much context is kept?

A: Context window.

37.

Q: What is context_filter used for?
A: To remove unwanted/unsafe input.

38.

Q: What does memory_enabled control?
A: Whether Agent uses memory.

39.

Q: Which config controls conversation history?
A: keep_history.

40.

Q: If keep_history = False, what happens?
A: No past conversation is stored.

41.

Q: Which config ensures context doesn’t exceed limits?
A: truncation_policy.

42.

Q: Which config hides sensitive inputs?
A: redaction_filter.

43.

Q: Which config sets priority for context items?
A: context_priority.

44.

Q: Can context filtering prevent data leaks?
A: Yes.

45.

Q: What if context window exceeds max length?
A: Model truncates earlier content.

Handoff Config

46.

Q: What is handoff in config?
A: Delegating tasks to another Agent.

47.

Q: Which config defines the handoff function?
A: on_handoff.

48.

Q: What is input_type in handoff config?
A: The format expected for delegated input.

49.

Q: What is input_filter in handoff config?
A: Validates or restricts inputs before handoff.

50.

Q: Which config decides if delegation is allowed?
A: is_enabled for handoff.

51.

Q: What happens if on_handoff is missing?
A: Default handoff behavior or error.

52.

Q: Can one Agent handoff to multiple Agents?
A: Yes.

53.

Q: Which config controls failure in handoff?
A: failure_error_function.

54.

Q: Which config defines output transformation?
A: custom_output_extractor.

55.

Q: Example: Agent A calls Agent B Who decides input format?
A: input_type config.

Safety & Guardrails

56.

Q: Which config ensures safety constraints?
A: Guardrails.

57.

Q: Which config prevents unsafe tool calls?
A: is_enabled = False.

58.

Q: Which config defines max token output?
A: max_tokens.

59.

Q: Which config prevents excessive API use?
A: rate_limit.

60.

Q: What does stop_sequences config do?
A: Stops generation at certain phrases.

61.

Q: Which config blocks banned words?
A: content_filter.

62.

Q: Which config ensures compliance policies?
A: policy_enforcer.

63.

Q: Which config ensures logging of harmful attempts?
A: safety_log.

64.

Q: Which config masks sensitive data?
A: redaction.

65.

Q: True or False: Guardrails are optional.
A: False – strongly recommended.

Error Handling

66.

Q: Which config defines custom error messages?
A: failure_error_function.

67.

Q: What happens if no error config set?
A: Default error behavior.

68.

Q: Which config retries after error?
A: retry policy.

69.

Q: Which config logs error events?
A: error_logger.

70.

Q: Which config falls back to another model?
A: failover_model.

71.

Q: Which config suppresses errors from user view?
A: silent_fail.

72.

Q: Which config escalates errors to admins?
A: escalation_policy.

73.

Q: Which config triggers re-prompt after failure?
A: retry_on_failure.

74.

Q: Which config stops Agent after repeated errors?
A: max_retries.

75.

Q: True or False: Agents cannot recover from failure.
A: False – recovery possible.

Advanced Config

76.

Q: What does name_override do?
A: Overrides tool or Agent name in config.

77.

Q: What does description_override do?
A: Customizes tool/Agent description.

78.

Q: Which config controls structured JSON output?
A: structured_output.

79.

Q: Which config forces schema validation?
A: strict_schema.

80.

Q: Which config controls streaming tokens?
A: streaming_enabled.

81.

Q: Which config defines orchestration rules?
A: orchestration_policy.

82.

Q: Which config allows cloning of Agents?
A: clone_config.

83.

Q: Which config determines output type?
A: output_type.

84.

Q: Which config sets log detail level?
A: log_level.

85.

Q: Which config controls tool_use_behavior?
A: tool_use_behavior param.

Practical Config Scenarios

86.

Q: Config: temperature=0.0 □ How does Agent behave?
A: Predictable, deterministic.

87.

Q: Config: temperature=0.9 □ How does Agent behave?
A: Creative, random.

88.

Q: Config: top_k=5 □ What does it do?
A: Considers 5 most likely tokens.

89.

Q: Config: tool_choice="auto" □ What happens?
A: Agent decides when to use tools.

90.

Q: Config: is_enabled=False □ What happens?
A: Tool or feature disabled.

91.

Q: Config: max_turns=3 □ What happens?
A: Agent reasoning limited to 3 steps.

92.

Q: Config: stop_sequences=["END"] □ What happens?
A: Generation halts when “END” appears.

93.

Q: Config: retry=2 □ What happens?
A: Agent retries failed step twice.

94.

Q: Config: structured_output=True □ What happens?
A: Outputs follow JSON schema.

95.

Q: Config: reset_tool_choice=True □ What happens?
A: Agent can switch tool mid-task.

Summary & Edge Cases

96.

Q: Which config makes Agents flexible with tools?
A: tool_choice.

97.

Q: Which config makes Agents predictable?
A: temperature=0.

98.

Q: Which config makes Agents robust to errors?
A: failure_error_function + retries.

99.

Q: Which config enables multiple Agents collaboration?
A: handoff configs.

100.

Q: What is the ultimate purpose of configuration?
A: To shape Agent’s reasoning, tool use, safety, and outputs.