DUMPSARENA

Claude Certified Architect - Foundations

Anthropic CCA-F

Version Demo

Total Demo Questions: 10

Total Premium Questions: 149

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

QUESTION NO: 1

You're building a security scanning workflow. When engineers need to locate all occurrences of a dangerous function like eval() across a large codebase, which tool should your agent use for content search?

A. Use Grep to search for the pattern "eval(" across all files in the codebase.

B. Use Glob with a pattern like **/*eval to find files, then Read each matching file.

C. Use Bash to run ls -R | grep eval to recursively list files containing eval.

D. Read the project's main entry file and follow import statements to trace where eval might be used.

ANSWER: A

Explanation:

Use Grep to search for the pattern "eval(" across all files in the codebase. Grep is the appropriate tool for content search because it searches the contents of files and returns matching lines along with their file paths and line locations. This gives an agent a direct, scalable way to identify every candidate use of a security-sensitive function throughout a repository, including uses in files that are not reachable from a main entry point or that appear in tests, scripts, configuration-adjacent source files, and legacy modules. Searching for the specific call-like pattern `eval(` also reduces irrelevant matches compared with looking only for the word `eval`, such as comments or variable names, while still providing a useful initial security-review inventory. In Claude Code, the Grep tool is designed to search file contents using regular expressions, and its results can then be inspected with targeted reads before remediation or further analysis. This follows Anthropic's guidance to use specialized tools for their intended operation: use file-content search to find text in files, then use file-reading tools to examine the relevant surrounding code. See the Claude Code tool documentation and Anthropic's tool use documentation.

QUESTION NO: 2

Your multi-agent research pipeline crashed after processing 12 of 28 documents. The web search agent had identified relevant sources, the document analyzer had partially completed extraction, and the synthesizer had begun pattern identification. You need to resume processing without repeating work or losing fidelity of prior findings. What state management approach best balances information fidelity with context efficiency when restoring agent state?

A. Index all agent outputs in a shared vector store. When resuming, each agent queries the store using semantic search to retrieve relevant prior findings.

B. Have each agent maintain its own persistent state file and reload it independently at the start of each session.

C. Have each agent persist a structured export to a known location. On resume, the coordinator loads the manifest and injects relevant state into agent prompts.

D. Persist the coordinator's conversation log containing all task delegations and responses, providing this to agents when resuming.

ANSWER: C

Explanation:

Have each agent persist a structured export to a known location. On resume, the coordinator loads the manifest and injects relevant state into agent prompts. This approach preserves durable, high-fidelity intermediate artifacts while keeping restored prompt context focused on the specific information each agent needs. A structured export can record document identifiers, processing status, extracted facts and citations, source metadata, synthesis progress, versions, and validation results. The coordinator can use a manifest as the authoritative recovery record: it identifies completed work, locates artifacts, determines which documents remain, and supplies only relevant state to the web-search, analysis, or synthesis task being resumed. This supports idempotent execution, so completed documents are not unnecessarily reprocessed, and enables reliable checkpointing after meaningful stages of the workflow. It also avoids treating a full conversational history as the application database. Anthropic recommends designing agent systems with clear orchestration, explicit task state, and context engineering so agents receive the right information at the right time rather than indiscriminately accumulating context. Structured artifacts and selective context loading make the workflow more inspectable, reproducible, and efficient as its

document set grows. See Anthropic's guidance on prompt engineering and context and its Building Effective Agents engineering guidance.

QUESTION NO: 3

Your extraction pipeline processes restaurant menus and must output structured JSON with fields for item names, descriptions, prices, and dietary tags. Some menus use inconsistent formatting—prices as "$12" vs "12.00", dietary info as icons vs text. What's the most reliable approach?

A. Extract data as-is and normalize formats in post-processing code after Claude returns.

B. Define a strict output schema and include format normalization rules in your prompt.

C. Use separate extraction calls for each field to ensure consistent handling of each type.

D. Request multiple extraction attempts per document and select the most common format.

ANSWER: B

Explanation:

Define a strict output schema and include format normalization rules in your prompt. This is the most reliable approach because it makes the required output contract explicit at the point where Claude interprets the source menu. A schema establishes the expected JSON fields and types, while normalization instructions define one canonical representation for variations in the input. For example, the prompt can require prices to be numeric values with exactly two decimal places and instruct Claude to translate recognized vegetarian, vegan, gluten-free, or other dietary icons into a fixed allowed list of tag strings. It should also specify how to represent unavailable or ambiguous values, such as using `null` rather than inventing data. Claude supports structured outputs through JSON Schema, which can ensure the response follows the requested structure and reduce downstream parsing and validation complexity. Clear, specific instructions and examples for edge cases are particularly useful in document-extraction workflows where visual and textual conventions vary between sources. The application should still validate the returned JSON and business rules before storing results, but schema-constrained extraction with normalization requirements provides a consistent, efficient primary mechanism for handling inconsistent menu formatting. See Anthropic's Structured outputs documentation and its guidance on clear, direct prompting.

QUESTION NO: 4

You've configured the system so that all four subagents have access to the complete set of 18 tools. During testing, agents frequently call tools outside their specialization—the synthesis agent attempts web searches, and the report generator tries to analyze documents. What is the primary cause of this poor tool selection behavior?

A. Choosing from 18 tools instead of 4-5 relevant ones increases decision complexity beyond reliable selection thresholds.

B. The agents' role descriptions in their system prompts conflict with having access to tools outside that role.

C. The tool definitions consume too much context window space, leaving insufficient room for task content.

D. The coordinator cannot track which capabilities each subagent has, leading to misrouted tasks.

ANSWER: A

Explanation:

Choosing from 18 tools instead of 4-5 relevant ones increases decision complexity beyond reliable selection thresholds. In a multi-agent architecture, each subagent should generally receive only the tools necessary for its defined responsibility. Giving every agent the entire tool catalog expands the action space it must evaluate for every turn, makes tool descriptions compete for attention, and weakens the practical connection between the agent's role and the operations it can perform. As the number of available tools rises, near-overlapping capabilities and less relevant tools can become plausible selections even when the system prompt assigns a narrow specialization. Tool access is therefore an important form of capability boundary, not merely a convenience or security control. Anthropic recommends designing tools with clear purposes and descriptions, and using prompt and system design to make the intended tool-selection behavior explicit. Restricting each

subagent to a small, role-relevant subset reduces ambiguity and helps produce more predictable, reliable routing of work. For example, a synthesis agent can be limited to tools for combining and formatting already-provided findings, while web-search and document-analysis tools remain available only to the specialist agents responsible for those tasks. See Anthropic's tool use documentation and prompt engineering guidance.

QUESTION NO: 5

In production, you observe that simple fact-checking queries (e.g., "What year was the Paris Climate Agreement signed?") traverse all four subagents sequentially, consuming 40+ seconds and significant tokens per query.

Complex comparative research benefits from the full pipeline. Your query distribution is diverse and evolving as users discover new applications. What's the most effective approach to optimize for varying query complexity?

A. Have the coordinator analyze each query and dynamically decide which subagents to invoke based on its assessment of query requirements.

B. Train a query complexity classifier on labeled historical data to predict optimal subagent combinations, retraining periodically as query patterns evolve.

C. Create a fast-path for factual questions that bypasses subagents entirely, routing all other queries through the complete pipeline to ensure research thorough

D. Implement pattern-based routing that categorizes queries by structure (single-fact vs. comparative vs. analytical) and maps each category to a predefined combination.

ANSWER: A

Explanation:

Have the coordinator analyze each query and dynamically decide which subagents to invoke based on its assessment of query requirements. This is the most effective approach because the coordinator can make routing decisions using the actual intent, scope, ambiguity, and research requirements of each incoming request. A simple, well-bounded factual question can be answered with a minimal path, while a query requiring source comparison, synthesis, or multiple perspectives can receive the full multi-agent workflow. This preserves the quality benefits of specialization without imposing the latency and token cost of every agent on every request.

Dynamic coordination is particularly appropriate when the query distribution is diverse and changes over time. Rather than relying on a fixed definition of "simple" or on a model trained against a potentially stale set of labeled historical examples, the coordinator can inspect the request in context and select an appropriate plan at runtime. Anthropic describes multi-agent systems as using a lead agent to delegate work to specialized subagents, with orchestration tailored to the task. The coordinator should also be prompted to use the smallest sufficient set of agents and to escalate only when the task warrants additional investigation. See Anthropic's multi-agent research system and the prompt-engineering guidance.

QUESTION NO: 6

Your system must extract event details from calendar invitations and output JSON that strictly conforms to a schema with fields for title, date, time, location, and attendees. Downstream reject any malformed or non-conformant JSON. What approach provides the most reliable schema compliance?

A. Define a tool with your target schema as input parameters and have Claude call it with the extracted data.

B. Pre-fill Claude's response with an opening brace to force JSON output, then complete and parse the response.

C. Append instructions like "Output only valid JSON matching the schema exactly" and implement retry logic to re-prompt when JSON parsing fails.

D. Include detailed JSON formatting instructions and the target schema in your prompt, then parse Claude's text response as JSON.

ANSWER: A

Explanation:

Define a tool with your target schema as input parameters and have Claude call it with the extracted data. Tool use supplies Claude with an explicit, machine-readable input schema, and the returned tool-use block contains structured arguments rather than relying on unconstrained prose generation. For a calendar-extraction workflow, define properties such as `title`, `date`, `time`, `location`, and `attendees`, including their expected types, required fields, nested attendee structure, and any relevant constraints. Claude then produces values as tool input, which the application can validate before it passes the result to downstream services. This is the appropriate design when correctness of a structured payload is a hard integration requirement rather than merely a preferred response format.

Anthropic's tool-use interface is specifically intended for cases where Claude needs to return structured data that an application can consume programmatically. A well-defined JSON Schema makes the extraction contract explicit and reduces ambiguity around field names and value shapes. The implementation should still validate tool arguments at the application boundary and handle missing or uncertain invitation details according to the schema's design, such as allowing nullable fields where appropriate. See Anthropic's tool use documentation and its JSON Schema guidance for tools.

QUESTION NO: 7

A user is expanding the research system beyond its single web search agent by adding specialized data sources. They add a financial API agent that returns structured JSON with revenue, margins, and growth rates; a news monitoring agent that returns prose summaries of recent developments; and a patent analysis agent that returns structured lists of technology areas. The synthesis agent combines these into executive briefings. Currently, it converts everything to bullet points, causing financial comparisons to lose tabular clarity and news summaries to lose narrative flow. What change would most improve briefing quality?

A. Update the synthesis agent to render each content type appropriately—financial data as tables, news as prose

B. Standardize all subagent outputs to JSON with fields for claim, evidence, source, and confidence

C. Add a format conversion layer between subagents and synthesis that transforms all outputs to a common intermediate representation

D. Standardize all subagent outputs to prose summaries with inline citations

ANSWER: A

Explanation:

Update the synthesis agent to render each content type appropriately—financial data as tables, news as prose is correct because the synthesis layer should preserve and use the most meaningful presentation format for each kind of evidence. Financial metrics such as revenue, margins, and growth rates are best presented in tables because tables make values, periods, and company or segment comparisons quickly scannable for executive readers. In contrast, recent news developments require prose to retain chronology, causal context, nuance, and the relationships among events. Patent-analysis results can likewise be rendered as structured lists or concise thematic sections when that best supports the briefing's goal. This approach allows specialized agents to provide information in formats suited to their source domain while giving the synthesizing Claude explicit responsibility for selecting an appropriate final representation for the audience and task. Anthropic guidance on multi-agent systems emphasizes using specialized subagents and a lead agent that coordinates their outputs into a coherent final response. The final agent should make deliberate decisions about organization and presentation rather than flattening all evidence into one format. See Anthropic's prompt engineering overview and its multi-agent research system engineering article.

QUESTION NO: 8

Your documents (query) tool returns results as "Found 3 documents: Q2 Budget Proposal, Q2 Budget Forecast, Annual Review". You want the agent to document (4, multi) and doc (24, multi). What return format would best enable these multi-step workflows?

A. URLs that users can click to open the document in their browser.

B. Structured data containing document IDs and metadata for each result.

C. A JSON array of document titles extracted from the search results.

D. More detailed human-readable descriptions including the size and authors.

ANSWER: B

Explanation:

Structured data containing document IDs and metadata for each result is the best format because multi-step agent workflows need stable, machine-readable identifiers that can be passed directly into later tool calls. A display title alone may be ambiguous, may change over time, and does not necessarily provide the exact reference required to retrieve a specific document. Including an ID for every search result lets the model reliably select the intended documents and invoke a follow-up operation—such as retrieving documents 4 and 24—without attempting to infer an identifier from prose. Useful metadata, such as title, source, document type, timestamps, access scope, or relevance score, provides enough context for the model to choose appropriately while preserving the canonical ID for execution. This also makes tool outputs easier to validate, parse, filter, and chain across calls. Anthropic's guidance for tool use emphasizes designing tools with clear input and output schemas so Claude can use their results reliably, including returning structured information that supports subsequent actions. See the Anthropic tool use documentation and the tool-use implementation guidance.

QUESTION NO: 9

Your extraction pipeline processes invoices and extracts line items, subtotals, tax amounts, and grand totals. During evaluation, you discover that in 18% of extractions, the sum of extracted line item amounts doesn't match the extracted grand total—sometimes due to OCR errors in the source document, sometimes due to extraction mistakes by the model. Downstream accounting systems reject records with mismatched totals. What's the most effective approach to improve extraction reliability?

A. Add a "calculated total" field where the model sums extracted line items alongside a "stated_total" field. Flag records for human review when values differ.

B. Extract line items and totals independently, then use a separate validation model to reconcile discrepancies by determining which extracted values are most likely correct.

C. Add few-shot examples demonstrating invoices where extracted line items sum correctly to the stated total, encouraging the model to produce mathematically consistent extractions.

D. Implement post-processing that automatically adjusts line item amounts proportionally when their sum doesn't match the stated total.

ANSWER: A

Explanation:

Adding a "calculated total" field where the model sums extracted line items alongside a "stated_total" field, then flagging differences for human review, is the appropriate reliability pattern because it preserves both the document's reported value and an independently derived consistency check. Invoice totals are a business-critical invariant: downstream systems need a clear, auditable signal that the itemized amounts reconcile with the stated grand total before accepting a record. The calculated value should ideally be computed deterministically in application code from the extracted structured line items, while the stated value is captured from the invoice. Comparing them makes OCR defects, missing or duplicated rows, decimal-placement errors, and extraction mistakes visible without silently changing financial data. Records that reconcile can proceed automatically; records that do not can be routed to an exception workflow with the original source available for review. This approach also provides measurable evaluation data, allowing the team to track reconciliation rates and improve prompts, document preprocessing, or review thresholds over time. Anthropic recommends structured outputs and validation for dependable production workflows; see the Structured outputs documentation and the guidance on increasing consistency.

QUESTION NO: 10

Production reviews reveal inconsistent handling of uncertainty in final reports. Sometimes conflicting subagent findings are synthesized into a single confident statement (losing nuance), while other times reports over-hedge with excessive

qualifications (becoming unhelpful). When the web search agent returns "industry analysts estimate $50B market size (methodology varies)" and the document analysis agent returns "peer-reviewed study estimates $35B (±$7B, 95% CI)," the coordinator either picks one arbitrarily or produces vague statements like "the market may be $35B-$50B depending on factors." What systematic approach best addresses this?

A. Implement a confidence calibration layer that normalizes subagent uncertainty expressions to standardized probability scores (0.0-1.0), then weight-average findings by their calibrated confidence.

B. Instruct the synthesis agent to structure reports with explicit sections distinguishing well-established findings from contested ones, preserving original source characterizations and methodological context.

C. Configure subagents to only report findings meeting a high-confidence threshold, filtering uncertain information before it reaches the coordinator.

D. Add a verification subagent that cross-references findings across sources, only passing claims to synthesis that are corroborated by at least two independent sources.

ANSWER: B

Explanation:

Instruct the synthesis agent to structure reports with explicit sections distinguishing well-established findings from contested ones, preserving original source characterizations and methodological context. This approach directly supports faithful synthesis: the final report can identify that the peer-reviewed estimate includes a defined confidence interval and describe the analyst estimate with its stated methodological variability, rather than converting both into an artificial single number or an unsupported probability calculation. The coordinator should make uncertainty legible and useful by attributing claims to their sources, retaining relevant qualifiers, and clearly separating evidence that is convergent from evidence that remains in tension. This gives readers the context needed to assess the result while still producing an actionable report.

Anthropic’s guidance emphasizes grounding responses in provided information and communicating limitations appropriately. For agentic systems, the synthesizing agent should preserve provenance and distinguish what is supported by available evidence from what needs further validation. A structured presentation of established, uncertain, and contested findings is therefore a robust reporting pattern, especially when sources differ in methodology, precision, or authority. It avoids inventing a false consensus while ensuring uncertainty does not obscure the substantive evidence. See Anthropic’s prompt engineering guidance and tool-use documentation.