TESTPASSPORT

QUESTIONS D'ENTRAÎNEMENT

UN MEILLEUR ACCOMPAGNEMENT

Exam : **CCAR-F**

Title : Claude Certified Architect - Foundations (CCAR-F)

Version : DEMO

1.You are integrating Claude Code into your Continuous Integration/Continuous Deployment (CI/CD) pipeline. The system runs automated code reviews, generates test cases, and provides feedback on pull requests. You need to design prompts that provide actionable feedback and minimize false positives. The automated review consistently flags patterns your team uses intentionally—force-unwrapping optionals in test files, using large coordinator classes that follow your established architecture, and importing internally maintained modules marked as deprecated in the public SDK. Developers dismiss approximately 30% of all findings as project-specific false positives.
Which approach prevents the model from generating these findings in the first place by supplying the project' s conventions as persistent context during every review?
A. Document the team's accepted patterns and intentional conventions in the project's CLAUDE.md file so the model receives this context during every review.
B. Configure the review to analyze only the changed lines in the diff without the surrounding file context, reducing the amount of code the model evaluates.
C. Build post-processing keyword filters that suppress findings containing terms such as "force unwrap," "large class," or "deprecated import" before results reach developers.
D. Have developers add inline suppression comments at flagged lines and preprocess diffs to exclude suppressed lines before sending code to the model.
Answer: A
Explanation:
Option A supplies the missing project context before Claude evaluates the code. Anthropic's CLAUDE.md documentation defines project-level files as persistent, version-controlled instructions for coding standards, architecture, workflows, and conventions. The described exceptions are stable repository facts: force-unwrapping is permitted in tests, coordinator classes follow an intentional pattern, and an internally maintained dependency remains approved despite its public deprecation status. Recording these facts concisely allows every review session to interpret the patterns correctly.
Option B removes surrounding evidence and would make architectural and cross-file judgments less reliable.
Option C hides findings after generation and can suppress genuine bugs containing the same keywords.
Option D adds noise to source files and distributes policy across individual call sites instead of documenting it centrally. The CLAUDE.md entries should be specific and scoped—for example, limiting the force-unwrapping exception to test paths rather than approving it globally. Anthropic notes that shorter, precise instructions are followed more consistently, so the file should state the accepted convention, its scope, and any conditions that would still make the pattern reportable.

2.You are building a multi-agent research system using the Claude Agent SDK. A coordinator agent delegates to specialized subagents: one searches the web, one analyzes documents, one synthesizes findings, and one generates reports. The system researches topics and produces comprehensive, cited reports.
A user expands the research system beyond its original web-search agent by adding specialized data sources. A financial API agent returns structured JSON containing revenue, margins, and growth rates. A news-monitoring agent returns prose summaries of recent developments. A patent-analysis agent returns structured lists of technology areas. The synthesis agent combines these results into executive briefings. Currently, it converts everything into bullet points, causing financial comparisons to lose tabular clarity and news summaries to lose their narrative flow.

What change would most improve briefing quality?
A. Standardize all subagent outputs as prose summaries with inline citations.
B. Add a format-conversion layer that transforms every subagent output into a common intermediate representation.
C. Update the synthesis agent to render each content type appropriately—for example, financial data as tables, news as prose, and patent areas as structured lists.
D. Standardize all subagent outputs as JSON containing claim, evidence, source, and confidence fields.
Answer: C
Explanation:
Option C preserves the information structure that makes each source useful. Financial metrics share comparable fields and therefore benefit from rows, columns, aligned units, and reporting periods. News findings require connected prose to preserve chronology and causal relationships, while patent technology areas are naturally represented as categorized lists. Anthropic's output-consistency guidance recommends specifying the exact output format needed for the task rather than relying on an unspecified default. Anthropic' s discussion of its multi-agent research system also recognizes specialized output stages for reports, structured data, and visualizations because specialist prompts can produce better results than generic coordinator processing.
Option A destroys the comparative structure of numerical data.
Option D can provide a useful provenance contract internally but does not determine how the executive briefing should present heterogeneous content.
Option B risks creating a lowest-common-denominator representation that discards source-specific advantages. The synthesis contract should preserve normalized facts and provenance internally while directing the report generator to select presentation forms according to the content's semantic structure and the executive reader's needs.

3.You are building a structured data extraction system using Claude. The system extracts information from unstructured documents, validates the output using JavaScript Object Notation (JSON) schemas, and maintains high accuracy. It must handle edge cases gracefully and integrate with downstream systems.
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 systems reject any malformed or non-conformant JSON.
What approach provides the most reliable schema compliance?
A. Pre-fill Claude's response with an opening brace to force JSON output, then complete and parse the response.
B. Append instructions like "Output only valid JSON matching the schema exactly" and implement retry logic to re-prompt when JSON parsing fails.
C. Define a tool with your target schema as input parameters and have Claude call it with the extracted data.
D. Include detailed JSON formatting instructions and the target schema in your prompt, then parse Claude' s text response as JSON.
Answer: C
Explanation:
A tool definition converts the desired extraction structure into an explicit machine-readable contract.

Claude returns the event information inside a tool_use block, with the tool arguments corresponding to the properties defined by the tool's input_schema. Anthropic specifies that custom tool parameters are described using JSON Schema, allowing the application to extract the structured arguments directly rather than attempting to recover JSON from ordinary prose. For current implementations, adding strict: true to the tool definition provides guaranteed conformance of tool-call inputs to the declared schema. (https://docs.anthropic.com/en /docs/agents-and-tools/tool-use/implement-tool-use)
Options A, B, and D remain prompt-based formatting techniques. They may improve the probability of valid JSON, but none creates the same schema-enforced interface. Prefilling an opening brace constrains the beginning of the response without guaranteeing valid field names, required properties, or data types. Retry logic detects failures only after generation and adds latency. Detailed formatting instructions can still produce malformed or structurally incorrect output.
Anthropic now also provides Structured Outputs for direct, schema-validated JSON responses. Within the options presented, however, a schema-defined tool is the only approach that establishes an explicit structured-output boundary rather than relying primarily on text-generation compliance. (https://docs.anthropic.com/en /docs/test-and-evaluate/strengthen-guardrails/increase-consistency)
Official references/topics: Tool Definitions, JSON Schema Input Contracts, Strict Tool Use, Structured Outputs.

4.You have 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. The agents' role descriptions in their system prompts conflict with having access to tools outside those roles.
B. The tool definitions consume too much context-window space, leaving insufficient room for task content.
C. The coordinator cannot track which capabilities each subagent has, leading to misrouted tasks.
D. Choosing from 18 tools instead of four or five relevant tools increases decision complexity beyond reliable selection thresholds.
Answer: A
Explanation:
Option A identifies the direct configuration contradiction. Each system prompt defines a specialized responsibility, but the corresponding tool allowlist presents unrelated actions as available capabilities. The synthesis agent may therefore interpret web searching as an acceptable way to fill evidence gaps, while the report generator may perform document analysis instead of limiting itself to report production. Anthropic's subagent documentation defines subagents as focused workers with their own prompts and specific tool access. Its tool-definition guidance also recommends reducing selection ambiguity by presenting clear, relevant tools. The appropriate correction is least-capability configuration: expose search tools to the search agent, document tools to the analyzer, synthesis utilities to the synthesizer, and formatting or output tools to the report generator.
Option B may affect token efficiency but does not explain the role-specific misuse pattern.
Option C concerns delegation, whereas the improper calls occur after successful delegation.
Option D is tempting, but Anthropic does not define four or five tools as a universal reliability threshold; 18 tools alone does not prove threshold failure. The decisive defect is misalignment between declared

roles and permitted capabilities.

5.You are building a structured data extraction system using Claude. The system extracts information from unstructured documents, validates the output using JSON schemas, and maintains high accuracy. It must handle edge cases gracefully and integrate with downstream systems.
After implementing tool use with strict schema definitions, JSON syntax errors are eliminated, but 5% of extractions still contain empty arrays or null values for required fields such as citations and methodology. Spot-checking reveals that the source documents contain this information, but in varied formats—inline citations versus bibliographies, and methodology sections versus details embedded in introductions.
What is the most effective way to address these failures?
A. Implement retry logic that resends requests when validation detects empty required fields.
B. Add few-shot examples demonstrating extractions from documents with varied structures, showing how to identify citations in different formats and locate methodology details across section types.
C. Build a regex-based post-processing layer that scans source documents for citation patterns and methodology keywords, populating empty fields when the model fails to extract them.
D. Modify the schema to make citations and methodology optional, and flag incomplete records for manual review instead of failing validation.
Answer: B
Explanation:
Option B targets the remaining failure mode: semantic recognition across heterogeneous document structures. Strict schemas eliminate malformed JSON and can guarantee that tool inputs conform to declared types, but they cannot force Claude to locate evidence that appears under unfamiliar headings or in atypical sections. Anthropic’s prompting guidance says that a few relevant, diverse, structured examples are among the most reliable ways to improve accuracy and consistency. Examples should therefore show inline citations, reference lists, numbered bibliographies, methodology sections, and methods embedded in introductions, each paired with the correct extracted structure. This teaches the intended evidence-location and granularity rules rather than merely repeating the same request.
Option A retries an unchanged prompt and can reproduce the same omission.
Option C introduces brittle regex rules that may miss nonstandard citations and mistake keyword mentions for methodology content.
Option D suppresses validation failures by weakening the contract, but it does not improve extraction and would convert recoverable omissions into incomplete records. The examples should be drawn from real failure cases, evaluated on a held-out set, and expanded when monitoring reveals new layouts. Schema constraints and few-shot coverage solve different layers of reliability and should be used together.