Anthropic.CCA-F.v2026-09-24.q153

Exam Code:	CCA-F
Exam Name:	Claude Certified Architect Foundations (CCA-F)
Certification Provider:	Anthropic
Free Question Number:	153
Version:	v2026-09-24
# of views:	104
# of Questions views:	1666
https://www.freecram.net/torrent/Anthropic.CCA-F.v2026-09-24.q153.html	

NEW QUESTION: 1
Which of the following statements correctly describe best practices for few-shot prompting? Choose 2 correct answers.
A. Few-shot examples are highly effective for demonstrating ambiguous-case handling.
B. Few-shot examples should be avoided when extracting data from documents with highly varied structures.
C. Every few-shot example provided should use a completely different output format to show flexibility.
D. The examples provided should maintain identical formatting to establish a clear pattern for the model to follow.
Answer: (SHOW ANSWER)
Best practices dictate that few-shot examples should demonstrate ambiguous-case handling and edge cases, and all examples must maintain identical formatting to establish a reliable pattern. Few-shot examples are actually highly recommended for documents with varied structures.

NEW QUESTION: 2
In a large organization, skill naming conflicts can occur. If a personal skill, a project-level skill, a plugin skill, and an enterprise managed skill all share the identical name, what is the correct priority order (from highest to lowest) that Claude Code uses to resolve the conflict?
A. Personal > Enterprise > Project > Plugins
B. Enterprise > Personal > Project > Plugins
C. Project > Enterprise > Personal > Plugins
D. Plugins > Project > Personal > Enterprise
Answer: (SHOW ANSWER)
Claude Code resolves skill name conflicts following a strict priority hierarchy: Enterprise (managed settings) take highest priority, followed by Personal skills, Project skills, and finally Plugins at the lowest priority.

NEW QUESTION: 3
When designing JSON schemas and interfaces for an agent's tools, which of the following practices represent recommended tool design principles? Choose 2 correct answers.
A. Keeping tool descriptions under 10 words to minimize context window token consumption.
B. Expanding tool descriptions to explicitly document what happens with empty inputs or invalid boundary data.
C. Splitting a generic document analysis tool into distinct, purpose-built extract_data and verify _ claim tools.
D. Combining all related backend API endpoints into a single monolithic tool to simplify the allowedTools array.

Answer: (SHOW ANSWER)
Recommended tool design principles include providing highly detailed descriptions (including edge cases and empty input handling) and splitting generic or monolithic tools into specialized, purpose-specific tools with clear input/output contracts. Minimal descriptions and monolithic tools severely degrade the LLM's ability to select and configure tools reliably.

NEW QUESTION: 4
When authoring a complex skill, relying on a massive 2000-line SKILL. md file wastes context window tokens. Which of the following describe the recommended "progressive disclosure" architectural pattern? Choose 2 correct answers.
A. Keep the SKILL .md file under 500 lines and link to supporting reference files that Claude reads only when needed.
B. Store executable validation code in a scripts/ directory and instruct Claude to run them rather than read their contents.
C. Use the fork_session command explicitly on line 1 of the markdown to split the file structure.
D. Configure an automatic /compact command to execute every 10 turns within the skill definition.
Answer: A,B (LEAVE A REPLY)
Progressive disclosure in skill architecture involves keeping the main SKILL.md file small (under 500 lines) and linking to supporting files (in directories like references/ and scripts!) that Claude reads or executes only when relevant. Instructing Claude to run scripts rather than read them prevents the script's raw source code from consuming context tokens.

NEW QUESTION: 5
A developer at builds an MCP server to standardize repetitive actions for the engineering team. They want to provide ready-to-use message sequences with specific formatting instructions that bypass the need for users to write them from scratch. Which MCP component fulfills this requirement?
A. MCP Subagents
B. MCP Prompts
C. MCP Tools
D. MCP Resources
Answer: (SHOW ANSWER)
MCP Prompts define a set of pre-built user and assistant messages that clients can request. They act as high-quality, tested templates that give Claude better starting instructions for specific tasks than a user might write on their own.

NEW QUESTION: 6
During the final synthesis phase, the system receives conflicting statistical data from two different subagents: one derived from a verified financial database, and another extracted from an outdated PDF. How should the synthesis agent handle this conflict?
A. Arbitrarily select the value from the financial database since structured databases are inherently more accurate.
B. Average the two conflicting values together to provide a balanced estimate in the final report.
C. Track information provenance by preserving both values, their sources, confidence levels, and timestamps, allowing informed resolution.
D. Halt the synthesis process and prompt the user to manually select the correct statistical value.
Answer: (SHOW ANSWER)
Tracking information provenance (source, confidence, timestamp) is essential in multi-agent synthesis. Conflicting data should be annotated with its source attribution rather than being arbitrarily selected or averaged, allowing the system or human reviewer to resolve the conflict based on reliability.

NEW QUESTION: 7
When designing the role of the coordinator agent in a multi-agent system, which of the following practices are recommended? Choose 2 correct answers.
A. Routing all inter-subagent communication through the coordinator to maintain observability and consistent error handling.
B. Analyzing query requirements and dynamically selecting which subagents to invoke rather than always routing through the full pipeline.

C. Forcing all queries through a flat architecture pipeline to maximize cross-agent state sharing.
D. Decomposing broad research topics into overly narrow subtasks to save context tokens at the risk of coverage gaps.
Answer: (SHOW ANSWER)
In a hub-and-spoke architecture, the coordinator routes all inter-subagent communication to maintain observability, consistent error handling, and controlled information flow. It should also dynamically select which subagents to invoke based on query complexity. Overly narrow decomposition and flat architectures are anti-patterns.

NEW QUESTION: 8
You are designing a prompt to extract structured data from unstructured documents. You decide to use tool_use with a strict JSON schem a. What specific guarantee does this approach provide?
A. It guarantees both structural compliance and the semantic correctness of the extracted data.
B. It eliminates JSON syntax errors and guarantees structural compliance, but does not prevent semantic errors.
C. It guarantees the model will not hallucinate missing information.
D. It eliminates the need for validation-retry loops.
Answer: (SHOW ANSWER)
Using tool_use with JSON schemas guarantees structural compliance (all required fields present, valid types, syntax correct) but does NOT guarantee semantic correctness (the actual values may still be inaccurate or hallucinated). You still need semantic validation rules.

NEW QUESTION: 9
When designing an agent to interact with a backend database, it is critical to distinguish between access failures and valid empty results.
Which of the following statements correctly describe handling these scenarios? Choose 2 correct answers.
A. An access failure (e.g., database timeout) should return isError: true so the agent knows the check was not actually performed.
B. An access failure should silently return an empty array C] to smoothly transition to alternative tools.
C. A valid empty result (searched successfully, but no matches) should return isError: false.
D. Both access failures and empty results should throw fatal exceptions to the coordinator to halt execution.
Answer: (SHOW ANSWER)
It is critical to distinguish between 'could not check' (access failure, isError: true) and 'checked and found nothing' (empty result, isError: false). Silently returning an empty array for an access failure misleads the agent into concluding the data does not exist.

NEW QUESTION: 10
To optimize a structured data extraction pipeline, you want to route only the most uncertain documents to your limited pool of human reviewers. Which method provides the most reliable calibration for this routing?
Freecram.net
A. Relying on the model's self-reported confidence score in the conversational text output.
B. Routing any document that triggers a validation-retry loop more than once.
C. Having the model output field-level confidence scores, and calibrating those thresholds using a labeled validation set.
D. Escalating documents based solely on the aggregate accuracy metric of the entire pipeline.
Answer: C (LEAVE A REPLY)
For effective human-in-the-loop routing, models should output structured, field-level confidence scores that are then calibrated against labeled validation sets. Self-reported conversational confidence is unreliable, and aggregate metrics mask specific field-level or document- type failures.

NEW QUESTION: 11

When the coordinator agent delegates a subtask to a specialized document analysis subagent, what is the best practice for passing context to ensure optimal performance?
A. Share the full coordinator conversation history with the subagent so it has the maximum possible context.
B. Pass ONLY the explicit context and findings relevant to that specific subagent's assigned task.
C. Grant the subagent access to the coordinator's scratchpad file using a Read tool.
D. Use the /compact command to summarize the coordinator's history before injecting it into the subagent's context.
Answer: (SHOW ANSWER)
Subagents do not automatically inherit parent context and should not receive the full conversation history, as this causes context pollution. The correct approach is to explicitly pass only the specific, relevant context and parameters needed for the subagent's assigned task.

NEW QUESTION: 12
In a hub-and-spoke multi-agent architecture, what is the best practice for passing context from the coordinator agent down to a specialized subagent?
A. Share the entire coordinator conversation history to ensure the subagent has maximum context visibility.
B. Pass only the explicit context, findings, and parameters specifically relevant to that subagent's assigned subtask.
C. Grant the subagent access to a shared global state object where the coordinator writes its findings.
D. Rely on automatic context inheritance, as subagents inherently share the coordinator's memory.
Answer: (SHOW ANSWER)
Context isolation is a core principle of multi-agent orchestration. Subagents do not automatically inherit parent context and should not receive the full conversation history, as this causes context pollution and wastes tokens. They should only be passed explicit, relevant context specific to their task.

NEW QUESTION: 13
In a multi-agent architecture, what is the best practice for handing off context from a coordinator agent to a specialized subagent?
A. Share the entire coordinator conversation history to guarantee maximum context.
B. Grant the subagent access to a shared global state database where the coordinator writes its findings.
C. Pass only the explicit context, findings, and parameters specifically relevant to the subagent's assigned subtask.
D. The subagent automatically inherits the coordinator's context window.
Answer: C (LEAVE A REPLY)
Context isolation is a fundamental principle of multi-agent architecture. Subagents do not automatically inherit parent context. Sharing the full coordinator history wastes tokens and introduces noise. You should explicitly pass only the specific, relevant context needed for the subtask.

NEW QUESTION: 14
When implementing structured data extraction for a application, what does utilizing tool_use with a strict JSON schema explicitly guarantee?
A. Both structural compliance and semantic correctness of the extracted data.
B. Structural compliance, ensuring valid JSON syntax and required fields, but not semantic accuracy.
C. Elimination of all hallucinations and inaccurate data extraction.
D. Automatic retry loops if the extracted values fail business logic validation.
Answer: (SHOW ANSWER)
Using tool_use with JSON schemas guarantees structural compliance (all required fields present, correct types, valid enum values), which eliminates JSON syntax errors. However, it does not guarantee semantic correctness, as the model may still extract wrong values or place them in the incorrect fields.

NEW QUESTION: 15

When building a multi-agent research pipeline using the Claude Agent SDK, which of the following practices represent architectural anti- patterns that must be avoided? Choose 2 correct answers.
A. Silently returning empty result sets for failed API lookups instead of reporting access failures.
B. Using a flat architecture where all agents share a global state and full conversation history.
C. Emitting multiple Task tool calls in a single coordinator response to execute subagents in parallel.
D. Requiring subagents to output structured claim-source mappings to preserve attribution.
Answer: (SHOW ANSWER)
Silently returning empty results for failures masks the error and misleads the agent, while flat architectures with global state sharing lead to context exhaustion and pollute subagent focus. Emitting multiple Task calls for parallel execution and requiring claim-source mappings are recommended best practices, not anti-patterns.

NEW QUESTION: 16
When integrating Claude Code into continuous integration and delivery pipelines, which of the following practices are highly recommended for reliability and automation? Choose 2 correct answers.
A. Executing Claude Code using the -p flag to ensure the process runs non-interactively.
B. Having the same session that generated the code perform the final review to ensure context consistency.
C. Isolating the code generator session from the code reviewer session to eliminate confirmation bias.
D. Using the Message Batches API to execute blocking pre-merge pull request checks to save costs.
Answer: (SHOW ANSWER)
In CI/CD pipelines, using the -p (or --print) flag is essential to prevent interactive hangs. Furthermore, isolating generator and reviewer sessions guarantees an unbiased review. Same-session reviews create confirmation bias, and the Batch API is too slow for blocking pre- merge checks.

Valid CCA-F Dumps shared by EduDump.com for Helping Passing CCA-F Exam! EduDump.com now offer the **newest CCA-F exam dumps**, the EduDump.com CCA-F exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com CCA-F dumps with Test Engine here:
https://www.edudump.com/exams/Anthropic/CCA-F/premium/ (**112** Q&As Dumps, **35%OFF** **Special Discount Code:** **freecram**)

NEW QUESTION: 17
Which of the following statements represent correct principles of multi-agent orchestration design? Choose 2 correct answers.
A. Parallel execution of subagents is achieved by emitting multiple Task tool calls in a single coordinator response.
B. A flat architecture where all agents share a global state is generally preferred over a hub-and-spoke architecture.
C. The coordinator should route all inter-subagent communication to maintain observability and consistent error handling.
D. Subagents should automatically inherit the coordinator's memory across invocations to save token costs.
Answer: (SHOW ANSWER)
Parallel execution is properly implemented by having the coordinator emit multiple Task tool calls in a single turn. Furthermore, in a hub-and-spoke architecture, the coordinator routes all inter-agent communication, providing observability and controlled information flow. Flat architectures with global state and automatic memory inheritance are anti-patterns leading to context exhaustion.

NEW QUESTION: 18
A properly structured SKILL.md file relies on YAML frontmatter metadata to inform Claude of its capabilities. Which of the following are REQUIRED fields in the skill frontmatter? Choose 2 correct answers.

A. name
B. description
C. allowed-tools
D. model
Answer: (SHOW ANSWER)
The 'name' and 'description' fields are the only strictly required fields in the SKILL.md frontmatter. Claude relies on the description for semantic matching to determine when to trigger the skill. Options like allowed-tools and model are powerful, but optional.

NEW QUESTION: 19
Which tool must be explicitly included in a coordinator agent's configuration to enable spawning subagents?
A. SubAgent
B. Fork
C. Task
D. Spawn
Answer: (SHOW ANSWER)
The Task tool is the designated mechanism used for spawning subagents. A critical configuration requirement is that the coordinator's allowedTools array must explicitly include the 'Task' tool to enable subagent creation.

NEW QUESTION: 20
When planning complex multi-flle architectural changes, such as a monolith-to-microservices migration, why is it recommended to use plan mode rather than direct execution?
Freedam.net
A. Plan mode automatically generates missing unit tests for legacy code before executing.
B. Plan mode utilizes the Message Batches API to reduce token costs by 50 percent.
C. Plan mode restricts Claude to using only read-only tools to explore the codebase, design a strategy, and get developer approval before making costly modifications.
D. Plan mode overrides the token limit context window by executing across multiple continuous sessions automatically.
Answer: (SHOW ANSWER)
Plan mode is designed for complex tasks requiring architectural decisions. It leverages read-only tools to explore and outline a plan, allowing developers to review the strategy before committing to file modifications. Direct execution is better suited for simple, well-defined fixes.

NEW QUESTION: 21
You are analyzing a massive legacy codebase and want to explore an alternative, potentially destructive refactoring approach without polluting your main session's context window. Which session management feature should you use?
A. - resume
B. fork session
C. / compact
D. / clear
Answer: (SHOW ANSWER)
The fork_session feature creates a branched session for exploration that inherits the existing context but keeps subsequent interactions isolated. This allows you to explore divergent approaches safely without exploration noise polluting your main conversation context.

NEW QUESTION: 22

When designing an independent code review instance to evaluate generated code at quicktechie.com, which specific context must be omitted to ensure an objective review?
A. The project's CLAUDE .md standards configuration.
B. The reasoning context and conversation history from the generation session.
C. The underlying JSON schemas of the affected components.
D. The list of available MCP tools.
Answer: (SHOW ANSWER)
An independent review instance must lack the prior reasoning context and conversation history of the generator. Supplying a fresh context prevents the reviewer from adopting the same flawed assumptions or confirmation bias that led to any generated bugs.

NEW QUESTION: 23
While working in Claude Code, you press Shift + Tab to enter Plan Mode. Which subagent is typically associated with isolating the verbose discovery phase to prevent context window exhaustion during planning?
A. The Review subagent
B. The Write subagent
C. The Explore subagent
D. The Plan-Executor subagent
Answer: (SHOW ANSWER)
The Explore subagent is used for isolating verbose discovery output and returning focused summaries to preserve the main conversation's context window during the research and planning phase.

NEW QUESTION: 24
In a coordinator-subagent research system, a web search subagent times out while attempting to retrieve information on a complex topic. How should this failure be propagated back to the coordinator agent?
A. The subagent should catch the timeout and return an empty JSON array L] marked as successful so the workflow continues smoothly.
B. Propagate the timeout exception directly to a top-level handler to immediately terminate the entire research workflow.
C. Implement automatic retry logic indefinitely within the subagent, returning a generic search unavailable status if it ultimately fails.
D. Return structured error context to the coordinator, including the failure type, the attempted query, any partial results, and potential alternative approaches.
Answer: (SHOW ANSWER)
Structured error propagation is critical. Returning structured error context (failure type, what was attempted, partial results) enables the coordinator to make an intelligent recovery decision. Silently suppressing the error by returning an empty result is a severe anti-pattern, and returning generic messages hides context needed for recovery.

NEW QUESTION: 25
To ensure that a model consistently classifies the severity of identified code issues, what is the best prompt engineering technique?
A. Define explicit severity criteria with concrete code examples for each severity level.
B. Instruct the model to self-report a confidence score from 1-10 for each finding.
C. Allow the model to dynamically generate severity labels based on context.
D. Lower the max_tokens limit so the model focuses only on critical bugs.
Answer: (SHOW ANSWER)
Providing explicit severity criteria paired with concrete code examples for each level bounds the model's behavior, leading to consistent severity classification and preventing subjective miscategorization.

NEW QUESTION: 26
A customer support agent must never process a refund exceeding $500 without explicit manager approval. What is the most robust and deterministic method to enforce this business rule?
A. Include a capitalized warning in the system prompt explicitly forbidding refunds over $500.
B. Provide negative few-shot examples in the prompt showing the agent successfully refusing high-value refunds.
C. Implement a PostToolUse hook that programmatically intercepts the refund tool call and blocks execution if the amount exceeds 500.
D. Set the tool choice parameter to 'auto' so the model can dynamically assess the financial risk.
Answer: C (LEAVE A REPLY)
For critical business rules requiring deterministic compliance, programmatic enforcement via Agent SDK hooks is required. Prompt- based guidance and few-shot examples are probabilistic and can occasionally be ignored by the LLM. Hooks run as code and provide 100% reliable enforcement.

NEW QUESTION: 27
An automated code review pipeline is configured to analyze pull requests modifying dozens of files. To avoid attention dilution and contradictory findings, what is the recommended task decomposition strategy?
A. Analyze all files simultaneously using a model with a maximum context window to ensure complete cross-file visibility.
B. Split the review into focused per-file local analysis passes, followed by a separate integration-focused pass examining cross-file data flow.
C. Execute three independent review passes on the entire pull request and only flag issues that appear in at least two runs.
D. Instruct the model to perform a single-pass review but explicitly prioritize security vulnerabilities over style inconsistencies.
Answer: B (LEAVE A REPLY)
For large code reviews, processing many files at once leads to attention dilution, missing bugs, and inconsistent depth. The best architectural strategy is to split the review into per-file local analysis passes to catch syntax/logic bugs, followed by a separate cross-file integration pass to evaluate data flow.

NEW QUESTION: 28
To optimize a CI/CD pipeline, your manager wants to use the Message Batches API to reduce costs by 50%. The pipeline includes a real-time, blocking pre-merge code review, and a separate nightly technical debt audit. How should you apply the Batch API?
A. Use the Batch API for both workflows to maximize cost savings, utilizing status polling for the pre-merge check.
B. Use synchronous API calls for both workflows, as the Batch API does not support structured JSON schema outputs.
C. Use the Batch API strictly for the nightly technical debt audit, and retain synchronous real-time calls for the blocking pre-merge review.
D. Switch both to the Batch API, but implement a timeout fallback that switches to synchronous calls if the batch exceeds 5 minutes.
Answer: (SHOW ANSWER)
The Message Batches API offers significant cost savings but processes within a 24-hour window with no latency guarantees. It is perfect for non-blocking, latency-tolerant workloads like nightly audits, but highly inappropriate for blocking pre-merge checks where developers need immediate feedback.

NEW QUESTION: 29
What is the only reliable and architecturally correct signal in the Claude API to determine whether an agentic loop should continue executing tools or terminate?
A. Parsing the assistant's natural language text output for completion keywords like 'task complete'.
B. Inspecting the stop_reason field of the response for tool_use versus end_turn.
C. Monitoring the total token count and exiting the loop when it reaches the context window limit.
D. Setting a hard maximum iteration cap of 5 loops to prevent infinite execution.
Answer: (SHOW ANSWER)
The stop_reason field is the only reliable signal for controlling the agentic loop. 'tool_use' indicates the loop must continue to execute tools, while 'end_turn' indicates Claude has finished reasoning and the loop should exit. Parsing natural language or setting arbitrary iteration caps are heavily tested anti-patterns.

NEW QUESTION: 30

When orchestrating an agentic workflow using the Claude Agent SDK, you must enforce a strict business rule: the agent is never allowed to issue refunds over $500. What is the most robust and deterministic method to implement this constraint?

Freecram.net

A. Add a capitalized warning in the system prompt: NEVER process a refund exceeding $500.

B. Implement a PostToolUse hook that intercepts the process_refund tool inputs, checks the amount, and programmatically blocks execution if it exceeds 500.

C. Configure the agent's tool_choice parameter to auto so it can decide when a refund is inappropriate.

D. Provide few-shot examples showing the agent refusing to process a $600 refund.

Answer: B (LEAVE A REPLY)

For critical business rules requiring deterministic compliance, programmatic enforcement via Agent SDK hooks (such as PreToolUse or PostToolUse logic) is the only 100% reliable method. Prompt-based guidance and few-shot examples are probabilistic and can occasionally be ignored by the LLM.

NEW QUESTION: 31

How should field-level confidence scores from a model be utilized to optimize a human-in-the-loop workflow?

A. By trusting the model's self-reported conversational confidence score to bypass human review.

B. By calibrating the scores using labeled validation sets to effectively route review attention to low-confidence extractions.

C. By using them to automatically increase the max_tokens limit on subsequent runs.

D. By setting tool_choice: 'any" whenever confidence drops below 0.5.

Answer: (SHOW ANSWER)

To optimize human review, models should output structured field-level confidence scores that are rigorously calibrated against labeled validation sets. This ensures review thresholds are accurate, allowing high-risk or low-confidence findings to be correctly routed to human

Valid CCA-F Dumps shared by EduDump.com for Helping Passing CCA-F Exam! EduDump.com now offer the **newest CCA-F exam dumps**, the EduDump.com CCA-F exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com CCA-F dumps with Test Engine here:

https://www.edudump.com/exams/Anthropic/CCA-F/premium/ (**112** Q&As Dumps, **35%OFF** **Special Discount Code:** **freecram**)

NEW QUESTION: 32

A CI pipeline uses Claude to generate missing test cases for new pull requests. However, it frequently suggests test scenarios that are already covered elsewhere. How should this be resolved?

A. Set the model's temperature to 0.0 to ensure deterministic test generation.

B. Provide existing test files in the context so the model can avoid suggesting duplicate scenarios.

C. Use a PostToolUse hook to deduplicate the generated tests against the existing codebase.

D. Instruct the model to only generate tests for functions with zero coverage.

Answer: B (LEAVE A REPLY)

To prevent Claude from generating duplicate or overlapping tests, you must provide the existing test files in the context window. This ensures test generation adds net-new value and accurately supplements the current test suite.

NEW QUESTION: 33

What is the recommended way to handle a subagent timeout during a complex research task to enable intelligent recovery?
A. Return structured error context including failure type, attempted query, and partial results.
B. Catch the timeout and return an empty result set marked as successful.
C. Terminate the entire research workflow and notify the user.
D. Return a generic string {"error": "Search unavailable"} to the coordinator.
Answer: (SHOW ANSWER)
Structured error context gives the coordinator the information it needs to make intelligent recovery decisions (e.g., retry with a modified query, try an alternative approach, or proceed with partial results). Returning a generic status or silently marking an empty result as successful prevents informed recovery.

NEW QUESTION: 34
Which iterative refinement strategy involves writing a test suite covering expected behavior and edge cases before implementation, then sharing subsequent test failures with Claude to guide progressive code improvements?
A. The Interview Pattern
B. Test-Driven Iteration
C. Multi-Pass Review
D. Prompt Chaining
Answer: (SHOW ANSWER)
Test-driven iteration gives Claude a concrete, verifiable goal: writing test suites covering requirements before implementation, executing them, and sharing the specific test failures with Claude to guide progressive improvement until tests pass.

NEW QUESTION: 35
You are assigned to restructure a monolithic application into microservices, requiring architectural decisions and changes across 45+ files. Which execution approach in Claude Code is recommended?
A. Start with direct execution and make changes incrementally, letting the implementation reveal the natural boundaries.
B. Use direct execution with a massive upfront prompt detailing how every service should be structured.
C. Enter plan mode to explore the codebase and design an implementation strategy before executing changes.
D. Write a static Prompt Chain that forces the agent to read and refactor each file sequentially.
Answer: (SHOW ANSWER)
Plan mode is specifically designed for complex, multi-file tasks involving architectural decisions. It enables safe codebase exploration and design before committing to file modifications, preventing costly rework. Direct execution is intended for simple, well-scoped tasks.

NEW QUESTION: 36
When building a conversation loop to handle multi-turn tool calling, what is the only reliable signal to determine when Claude has finished requesting tools and the loop should terminate?
A. The model's self-reported confidence score reaches 0.9.
B. The assistant text content includes the phrase 'task complete'.
C. The token count reaches the predefined limit.
D. The stop_reason field of the API response equals end_turn.
Answer: (SHOW ANSWER)
The stop_ reason field is the only deterministic signal for controlling the agentic loop. tool_use execute tools, while end_turn indicates the loop must continue to indicates Claude has finished reasoning and the loop should exit. Parsing natural language is an anti-pattern.

NEW QUESTION: 37
In a multi-pass review architecture, how can you effectively automate the routing of identified findings for human review calibration?
A. Run verification passes where the model outputs field-level confidence scores alongside findings to enable calibrated review routing.
B. Rely strictly on the model's self-reported conversational confidence score in the output.
C. Escalate any finding that flags more than 5 lines of code.
D. Use the Message Batches API to delay findings until the end-of-week manual review.
Answer: (SHOW ANSWER)
To optimize human-in-the-loop workflows, models should output structured, field-level confidence scores that are calibrated against labeled validation sets. This allows reliable routing of low-confidence or high-risk findings for human review.

NEW QUESTION: 38
You want to ensure specific testing conventions are automatically applied whenever Claude generates or edits a test file (e.g., *. test. tsx). These test files are spread across dozens of different subdirectories. What is the most maintainable configuration approach?
A. Create a rule file in . claude/rules/ with YAML frontmatter specifying paths: test .tsx"].
B. Place a separate CLAUDE .md file in every single subdirectory that contains test files.
C. Consolidate all testing conventions in the root CLAUDE. md file under a general testing header and rely on semantic inference.
D. Create a custom slash command that manually injects testing rules when invoked during a session.
Answer: (SHOW ANSWER)
Using .claude/rules/ files with YAML frontmatter paths fields containing glob patterns (like *P.test.tsx) is the recommended approach for conditional rule activation across scattered files. This ensures rules load only when editing matching files, saving context window tokens and eliminating the need to duplicate CLAUDE.md files in every directory.

NEW QUESTION: 39
You are tasked with using Claude Code to add comprehensive unit tests to an unfamiliar legacy codebase at quicktechie.com. Which approach represents the recommended decomposition strategy for this open-ended task?
A. Start by writing tests for the first file in the directory structure and proceed sequentially through the codebase.
B. Request Claude to automatically generate a monolithic test suite in a single prompt execution.
C. First map the project structure, identify high-impact areas, and create a prioritized plan that adapts as dependencies are discovered.
D. Use a fixed prompt chaining pipeline that executes npm test after generating each line of code.
Answer: (SHOW ANSWER)
For open-ended and complex tasks like adding tests to a legacy codebase, the best decomposition strategy involves first mapping the structure and identifying high-impact areas, then creating a prioritized plan that can dynamically adapt as the agent discovers dependencies during implementation.

NEW QUESTION: 40
An automated extraction pipeline reports 96% aggregate accuracy, but downstream users complain about frequent errors on complex contracts. What monitoring methodology resolves this visibility gap?
Freeram.net
A. Increasing the overall sampling size randomly to capture a broader dataset.
B. Tracking accuracy by specific document type and field (stratified metrics).
C. Switching entirely to the Message Batches API for asynchronous processing.
D. Requiring a 100% human-in-the-loop review for all documents.
Answer: (SHOW ANSWER)

Aggregate accuracy metrics can mask severe per-document-type failures (e.g., contracts failing frequently while simple receipts succeed nearly 100% of the time). Tracking accuracy using stratified metrics (per document type and field) reveals these hidden failures.

NEW QUESTION: 41
After executing a requested tool in an agentic loop, how must the application handle the follow-up request to ensure the model can accurately reason about the next action?
A. Pass only the new tool result to save token space in the context window.
B. Pass the tool result inside an assistant message equipped with an is_error flag.
C. Append a user message containing a tool_result block with the matching tool_use_Id, and include the full prior conversation history.
D. Use the /compact command to summarize the tool output before returning it to the loop.
Answer: C (LEAVE A REPLY)
Because Claude is stateless, the application must provide the entire conversation history. When returning a tool result, it must be appended as a user message containing a tool_result block that explicitly includes the specific tool_use_id matching the original request.

NEW QUESTION: 42
Which prompt strategy is most effective at reducing false positives in an automated code review system?
A. Instruct the model to only report high-confidence findings.
B. Provide explicit, measurable criteria like "flag functions exceeding 50 lines".
C. Ask the model to "find all code issues" to ensure nothing is missed.
D. Use a temperature of 0.0 to ensure deterministic outputs.
Answer: (SHOW ANSWER)
Explicit, measurable criteria enable automated validation of output quality, reduce ambiguity, and narrow the scope, directly reducing false positives. Asking for "all code issues" is too vague, and self-reported confidence is unreliable.

NEW QUESTION: 43
The configuration hierarchy in Claude Code resolves settings from multiple levels. Which of the following are valid configuration layers where CLAUDE.md files can be placed? Choose 2 correct answers.
A. User-level (e.g., 4. claude/CLAUDE. md) for personal preferences.
B. Global system-level (e.g., /etc/c1aude/CLAI_JDE. md) for machine-wide mandates.
C. Directory-level (e.g., src/api/CLAUDE .md) for module-scoped conventions.
D. Cloud-level (e.g., s3 : //config/CLAUDE .md) for distributed synchronization.
Answer: (SHOW ANSWER)
The three valid configuration layers in Claude Code are User-level (-/.claude/CLAUDE.md), Project-level (.claude/CLAUDE.md), and Directory-level (scoped to specific subdirectories). Global system-level and cloud-level configs are not standard mechanisms.

NEW QUESTION: 44
An MCP tool processes financial refunds. A policy violation occurs because the requested amount exceeds the user's maximum allowed limit. What is the most architecturally sound way the tool should respond?
A. Return a generic "Operation failed" string to avoid leaking internal policy limits.
B. Throw a fatal system exception to halt execution.
C. Return structured error metadata with isError: true, a human-readable context description of the policy violation, and isRetryable: false.
D. Return an empty response payload so the agent can seamlessly try a different tool.

Answer: (SHOW ANSWER)
Business logic and policy errors require structured error context including isError: true, an explanation of the violation, and isRetryable: false. This allows the agent to communicate the appropriate denial reason to the user rather than blindly guessing or uselessly retrying.

NEW QUESTION: 45
A quicktechie.com team wants to run a nightly code quality audit across the entire codebase using Claude Code to reduce API costs.
Which approach is most cost-effective and architecturally sound?
A. Run claude -p synchronously for each file in a loop.
B. Use the Message Batches API for non-urgent batch processing to achieve 50 percent cost savings.
C. Run multiple Claude Code instances in parallel.
D. Schedule a single long-running Claude Code session overnight.
Answer: (SHOW ANSWER)
The Message Batches API processes requests within a 24-hour window and offers a 50% cost savings compared to synchronous requests. It is ideal for latency-tolerant, non-urgent workloads like overnight technical debt audits.

NEW QUESTION: 46
When generating complex mock data setups for integration tests, natural language descriptions often produce inconsistent structural results from Claude. What is the most effective technique to fix this?
A. Use the /compact command before requesting the test generation.
B. Provide 2-3 concrete input/output examples demonstrating the exact expected transformation.
C. Increase the max_tokens parameter to give the model more space to reason.
D. Ask Claude to evaluate its own test generation output before returning it.
Answer: (SHOW ANSWER)
Concrete input/output (few-shot) examples are the most effective way to communicate expected transformations when prose descriptions are interpreted inconsistently. They demonstrate the exact structural formatting required for complex data setups.

Valid CCA-F Dumps shared by EduDump.com for Helping Passing CCA-F Exam! EduDump.com now offer the **newest CCA-F exam dumps**, the EduDump.com CCA-F exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com CCA-F dumps with Test Engine here:
https://www.edudump.com/exams/Anthropic/CCA-F/premium/ (**112** Q&As Dumps, **35%OFF** **Special Discount Code:** **freecram**)

NEW QUESTION: 47
Before automating the extraction of high-confidence fields in a quicktechie.com processing system, what validation step is crucial?
A. Validating the JSON schema syntax using a PreToolUse hook.
B. Validating accuracy segmented by document type and specific field to verify consistent performance across all segments.
C. Confirming consensus through 3 separate monolithic review passes.
D. Checking that the model's self-reported sentiment analysis is positive.
Answer: (SHOW ANSWER)
Before reducing human review and automating high-confidence extractions, it is crucial to validate accuracy segmented by both document type and specific field. This ensures that performance is actually consistent across all segments, rather than relying on misleading aggregate averages.

NEW QUESTION: 48
How should subagents handle transient failures before propagating errors to the coordinator?
A. Implement local error recovery within the subagent and only propagate errors that cannot be resolved locally.
B. Immediately propagate all failures to ensure the coordinator maintains full observability.
C. Silently drop the failed query and proceed with other data sources without notifying the coordinator.
D. Use a PreToolUse hook to block the error from reaching the coordinator entirely.
Answer: (SHOW ANSWER)
Subagents should attempt to implement local recovery for transient failures first. If they cannot resolve the issue, they should then propagate the error to the coordinator along with what was attempted and any partial results obtained.

NEW QUESTION: 49
When a tool lookup (e.g., get_customer) returns multiple matches, what is the most reliable way for the agent to proceed?
A. Automatically escalate to a human agent.
B. Select the most recently active account using a heuristic.
C. Instruct the agent to ask the user for additional identifiers for clarification.
D. Return a structured error and terminate the workflow.
Answer: (SHOW ANSWER)
When multiple matches are found, relying on heuristics to guess the correct user can lead to security and billing risks. The correct reliability decision is to instruct the agent to ask the user for additional clarifying identifiers.

NEW QUESTION: 50
During a multi-source research workflow, what is the primary risk to information provenance when intermediate findings are compressed using standard summarization?
Freecram.net
A. The model's extended thinking budget is exhausted before synthesis can occur.
B. Source attribution is silently lost unless structured claim-source mappings are explicitly required and preserved.
C. The context window automatically truncates the middle sections of the summary.
D. It triggers a PreToolUse hook that blocks the synthesis agent from reading the data.
Answer: (SHOW ANSWER)
A major risk in multi-source synthesis is that source attribution is lost during summarization steps when findings are compressed. To prevent this, architects must require structured claim-source mappings that downstream agents are instructed to explicitly preserve through the synthesis phase.

NEW QUESTION: 51
An agent at quicktechie.com needs to process diverse data and has been given access to 18 different tools. You notice the agent frequently misroutes requests and selects the wrong tool. What is the most architecturally sound fix?
A. Combine all 18 tools into a single monolithic tool with 20+ optional parameters to simplify the list.
B. Reduce the tool count to 4-5 per agent by distributing the tools across specialized subagents.
C. Set tool_choice: "any" so the agent can parallelize the execution of all 18 tools simultaneously.
D. Switch to a higher-tier model with a larger context window so it can comprehend all 18 schemas perfectly.
Answer: (SHOW ANSWER)

Giving a single agent access to too many tools (e.g., 18 instead of the optimal 4-5) degrades tool selection reliability because it increases decision complexity and ambiguity. The best practice is to group related tools and distribute them across specialized subagents (e.g., Customer Agent, Order Agent).

NEW QUESTION: 52
A PreToolUse hook is configured to intercept and block unauthorized file reads based on restricted file paths. How does the hook properly communicate the specific blocking reason back to Claude so the model can adapt?
A. By silently returning an empty JSON array [ ] to the context window.
B. By returning an exit code of 2 and writing the explanation to standard error, which is fed back as context.
C. By completely terminating the session and initiating a new one with a revised system prompt.
D. By overriding the context window entirely with an 'access denied' templated message.
Answer: (SHOW ANSWER)
In Claude Code and Agent SDK hooks, returning an exit code of 2 (or explicitly returning a 'blocked' flag with a reason string) and providing an explanation feeds the exact error context back to Claude. This allows the model to understand why the action was blocked and adjust its next steps accordingly.

NEW QUESTION: 53
A large pull request modifies 15 files. A single-pass review produces superficial comments and misses obvious cross-file bugs. How should you restructure the automated review architecture? Choose 2 correct answers.
A. Split into focused passes: analyze each file individually for local issues.
B. Run a separate integration-focused pass examining cross-file data flow.
C. Switch to a higher-tier model with a larger context window to give all 15 files adequate attention in one pass.
D. Run three independent full reviews and require consensus before posting comments.
Answer: (SHOW ANSWER)
Splitting large code reviews into per-file local analysis passes (to catch syntax/local logic bugs) plus a separate cross-file integration pass prevents attention dilution. Relying solely on larger context windows or requiring consensus across full monolithic runs are anti-patterns.

NEW QUESTION: 54
A application features a tool designed to fetch a user's profile. During execution, the tool fails to connect to the database due to a timeout. Why is silently returning an empty array [ ] in this scenario considered a catastrophic anti-pattern?
A. It prevents the use of few-shot prompting in the next iteration.
B. It causes the agent to misinterpret an access failure (could not check) as a valid empty result, wrongly concluding the user does not exist.
C. It forces the model to ignore all subsequent tool calls in the session.
D. It causes the Claude API to throw a 400 Bad Request error due to missing required schema fields.
Answer: (SHOW ANSWER)
Access failures (e.g., database down, timeout) must be explicitly distinguished from empty results (e.g., checked database, found zero matches). Silently returning an empty array for an access failure misleads the agent into believing the data does not exist, leading to completely incorrect assumptions.

NEW QUESTION: 55
To locate all test files written in TypeScript within a specific directory subtree (e.g., **/*. test. tsx), which built-in tool is architecturally correct to use?
A. Grep
B. Glob
C. Bash

D. Read
Answer: B (LEAVE A REPLY)
The Glob tool is specifically designed to find files matching naming or path patterns (e.g., finding all test files via globbing). Grep is used for searching file contents, and using Bash (e.g., via the 'find' command) is an anti-pattern when a purpose-built tool like Glob exists.

NEW QUESTION: 56
Which of the following is an explicit limitation of using the Message Batches API?
A. It cannot process explicit JSON schemas for structured output.
B. It does not support multi-turn tool calling within a single request.
C. It requires a dedicated local MCP server to execute.
D. It severely restricts the maximum context window capacity.
Answer: (SHOW ANSWER)
The Message Batches API does not support multi-turn tool calling within a single request. You cannot execute tools mid-request and return results dynamically back to the model within the same asynchronous batch execution.

NEW QUESTION: 57
In which scenario is it architecturally more reliable to start a fresh session injected with a structured summary rather than using - - resume to continue an existing session?
A. When the prior tool results in the existing session's context window have become completely stale.
B. When you want to maintain the exact reasoning chain from the previous generation steps.
C. When the target codebase has not undergone any structural changes.
D. When executing a fast, single-file typo correction.
Answer: (SHOW ANSWER)
While -- resume is excellent for continuing recent work, if significant time has passed or the environment has changed, prior tool results become stale. Starting a fresh session and injecting a structured summary of findings is more reliable than resuming a session filled with outdated API responses or file contents.

NEW QUESTION: 58
When processing a batch of 10,000 documents via the Message Batches API, 45 requests fail due to context window limits. What is the recommended failure handling strategy?
A. Resubmit the entire batch of 10,000 requests to guarantee correct ordering.
B. Resubmit only the failed documents (identified by their custom_id) after modifying them, such as chunking oversized documents.
C. Switch immediately to the synchronous API to retry the failed documents.
D. Ignore the failures, as the API inherently guarantees a 99% success SLA.
Answer: (SHOW ANSWER)
Batch failures should be handled by identifying the specific failed requests using their custom_id, modifying them appropriately (e.g., chunking oversized documents to fit context limits), and resubmitting only those failed items.

NEW QUESTION: 59
When providing Claude with a 50-page document, you notice it accurately recalls details from the first and last pages but misses a critical instruction buried on page 25. What is this phenomenon called?
A. The "lost in the middle" effect
B. Context truncation
C. Progressive summarization

D. Attention dilution
Answer: (SHOW ANSWER)
The "lost in the middle" effect describes how models reliably process information at the beginning and end of long inputs but may omit findings or instructions buried in middle sections.

NEW QUESTION: 60
A developer needs to migrate a monolithic application into microservices, requiring changes across 45+ files, updates to module dependencies, and new architectural boundaries. Which execution approach in Claude Code is recommended for this task?
Freecram.net
A. Use Direct Execution to incrementally modify files one by one until the migration is complete.
B. Use the /compact command heavily to fit all 45 files into the context window for a single-pass Direct Execution.
C. Use Plan mode to explore the codebase, map dependencies, and design an implementation strategy before committing to file modifications.
D. Write a static Prompt Chain that forces the agent to read, refactor, and save each file sequentially.
Answer: (SHOW ANSWER)
Plan mode is specifically designed for complex, multi-file tasks involving architectural decisions and multiple valid approaches. It allows the agent to explore and outline a plan using read-only tools before executing changes, reducing costly rework. Direct execution is for simple, well-scoped changes.

NEW QUESTION: 61
A developer at quicktechie.com is assigned a task to migrate a core utility library that is imported by over 45 different files across the repository. Which execution approach should the developer take?
A. Use Direct Execution to make incremental changes and let the implementation naturally reveal dependency boundaries.
B. Use Plan Mode to explore the codebase, understand dependencies, and design an architectural approach before committing to file modifications.
C. Use a custom slash command designed to bypass context limits.
D. Start with Direct Execution, and manually switch to Plan Mode only if unexpected compilation errors occur.
Answer: (SHOW ANSWER)
Plan mode is specifically designed for complex tasks involving large-scale changes, architectural decisions, and multi-file modifications (such as a library migration affecting 45+ files). It enables safe codebase exploration before committing to costly rework.

Valid CCA-F Dumps shared by EduDump.com for Helping Passing CCA-F Exam! EduDump.com now offer the **newest CCA-F exam dumps**, the EduDump.com CCA-F exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com CCA-F dumps with Test Engine here:
https://www.edudump.com/exams/Anthropic/CCA-F/premium/ (**112** Q&As Dumps, **35%OFF** **Special Discount Code:** **freecram**)

NEW QUESTION: 62
A subagent attempts to query an external database but encounters a timeout. How should this error be propagated back to the coordinator agent?
A. The subagent should silently return an empty JSON array to allow the workflow to proceed smoothly.
B. Return a generic 'Operation failed' string to prevent leaking infrastructure details.
C. Return structured error context, including the failure type (access failure), what was attempted, and partial results.
D. Throw a fatal exception to immediately terminate the entire research workflow.
Answer: (SHOW ANSWER)

Structured error propagation is critical. Returning structured error context allows the coordinator to distinguish between an 'access failure' (could not check) and an 'empty result' (checked and found nothing) and make an intelligent recovery decision. Silently returning empty results for failures is a catastrophic anti-pattern.

NEW QUESTION: 63
Production logs indicate that Claude is frequently misrouting requests to an analyze_content tool when the analyze_document tool should have been selected. Both tools currently have simple, one-sentence descriptions. What is the most effective architectural fix?
A. Provide negative few-shot examples in the system prompt instructing Claude not to confuse the two tools.
B. Rename the tools to eliminate functional overlap and expand descriptions to explicitly state boundaries explaining when to use each.
C. Use a PreToolUse hook to programmatically redirect analyze_content calls to analyze_document.
D. Increase the model's temperature parameter to encourage more diverse tool selection.
Answer: (SHOW ANSWER)
Tool descriptions are the primary mechanism models use for selection. Ambiguous or overlapping descriptions (like analyze_content vs analyze_document) cause misrouting. The architectural solution is to rename tools clearly and expand descriptions to define explicit boundaries and edge cases explaining when to use one versus the other.

NEW QUESTION: 64
When implementing a self-evaluation loop for structured data extraction, your agent reports 96% aggregate accuracy, but downstream users complain about frequent errors on complex contracts. What monitoring methodology resolves this evaluation visibility gap?
A. Tracking accuracy by specific document type and field segment (stratified metrics).
B. Increasing the overall sampling size randomly to capture a broader dataset.
C. Switching entirely to the Message Batches API for asynchronous processing.
D. Executing the evaluation passes inside a PreToolUse hook.
Answer: (SHOW ANSWER)
Aggregate accuracy metrics can mask severe per-document-type failures (e.g., contracts failing frequently while simple receipts succeed nearly 100% of the time). Tracking accuracy using stratified metrics (per document type and field) reveals these hidden failures before automating high-confidence extractions.

NEW QUESTION: 65
Your CI pipeline must consume Claude's code review feedback programmatically to post inline comments on a GitHub pull request. How should you enforce the exact structural shape of the output?
A. Use regex to parse the unstructured markdown text output returned by Claude.
B. Add - -output-format json and supply the required structure via the --json-schema CLI flag.
C. Inject a PostToolUse hook that formats the text output into a JSON object.
D. Provide at least 10 few-shot examples of valid JSON structures in the system prompt.
Answer: (SHOW ANSWER)
To ensure machine-parseable, structured output in automated pipelines, Claude Code must be executed with the -output-format json flag combined with the --json-schema flag to strictly enforce the shape of the returned data.

NEW QUESTION: 66
Your codebase has test files spread throughout multiple directories alongside the components they test (e.g., Button. test .tsx next to Button. tsx). How can you ensure Claude automatically applies specific testing conventions whenever it generates or edits a test file?
A. Create a rule file in . claude/rules/ with YAML frontmatter specifying paths: . test.tsx"].
B. Consolidate all testing conventions in the root CLAUDE .md file and rely on Claude's inference.

C. Place a separate CLAUDE .md file in every subdirectory containing test files.
D. Define an MCP server resource to fetch testing standards dynamically.
Answer: (SHOW ANSWER)
Using . claude/rules/ files with glob patterns in the paths field allows conventions to be automatically and conditionally applied based on file paths. This ensures the testing rules load deterministically whenever matching test files are generated or edited.

NEW QUESTION: 67
In an automated CI/CD code review workflow, why is it considered an architectural anti-pattern to have the same Claude session that generated the code also review its own changes?
A. It causes the token limit of the context window to exceed maximum capacity.
B. The reviewer retains the reasoning context from the generation phase, creating confirmation bias.
C. The Message Batches API does not support synchronous self-review workflows.
D. Tools invoked in the first pass are permanently locked and cannot be reused in the second pass.
Answer: (SHOW ANSWER)
Same-session self-review is ineffective because the model retains its original reasoning context and assumptions in the conversation history, creating a blind spot or confirmation bias. Using an independent review instance with fresh context evaluates the code objectively.

NEW QUESTION: 68
A quicktechie.com enterprise organization deploys a managed skill named code- review. A developer also creates a personal skill in ~/. claude/ skills/ with the exact same name code-review.
When the developer requests a review, which skill takes priority?
A. The personal skill overrides the enterprise skill because user-level scope takes precedence.
B. The enterprise managed skill takes highest priority and strictly overrides the personal skill.
C. Claude automatically merges the instructions from both skills into a single execution context.
D. A validation error is thrown, and neither skill loads until the naming conflict is resolved.
Answer: (SHOW ANSWER)
Claude Code enforces a strict priority hierarchy for skill conflicts: Enterprise (managed settings) > Personal > Project > Plugins. The enterprise managed skill takes the highest priority, ensuring that mandatory organizational standards cannot be bypassed by personal configurations.

NEW QUESTION: 69
How should a quicktechie.com application handle verbose tool outputs (e.g., an order lookup returning 40+ fields when only 3 are relevant) to optimize the context window?
A. Pass the full output and rely on the model's semantic focus to ignore irrelevant fields.
B. Trim the tool outputs programmatically to include only the return-relevant fields before they accumulate in the context history.
C. Use a PreToolUse hook to block the API response.
D. Require the agent to run a web search to summarize the JSON object.
Answer: (SHOW ANSWER)
Tool results accumulate in context and consume tokens disproportionately to their relevance. Best practice dictates trimming verbose tool outputs to only the relevant fields before they are appended to the context history, preventing rapid context degradation.

NEW QUESTION: 70
Which of the following represent anti-patterns when designing escalation and ambiguity resolution logic for Claude? Choose 2 correct answers.

Freecram.net

A. Automatically escalating a case whenever the model's self-reported confidence score drops below a specific threshold.
B. Escalating immediately when a customer explicitly demands to speak to a human representative.
C. Routing a case to a human agent purely because the user uses highly negative or frustrated language.
D. Escalating a transaction when it exceeds a programmatic business threshold enforced via a PostToolUse hook.
Answer: (SHOW ANSWER)
Self-reported confidence scores and sentiment-based routing are explicitly identified as escalation anti-patterns. Valid triggers include explicit human requests, programmatic business thresholds, capability limits, and policy gaps.

NEW QUESTION: 71
Your agent achieves a low first-contact resolution rate because it attempts to autonomously handle complex policy exceptions, while simultaneously escalating straightforward tasks. What is the most effective approach to improve this escalation calibration?
A. Deploy a separate classifier model to predict which requests need escalation before the main agent begins.
B. Require the agent to self-report a confidence score (1-10) and route to humans when confidence falls below 7.
C. Implement sentiment analysis to detect frustration levels and automatically escalate when they are high.
D. Add explicit escalation criteria to the system prompt with few-shot examples demonstrating when to escalate versus resolve autonomously.
Answer: (SHOW ANSWER)

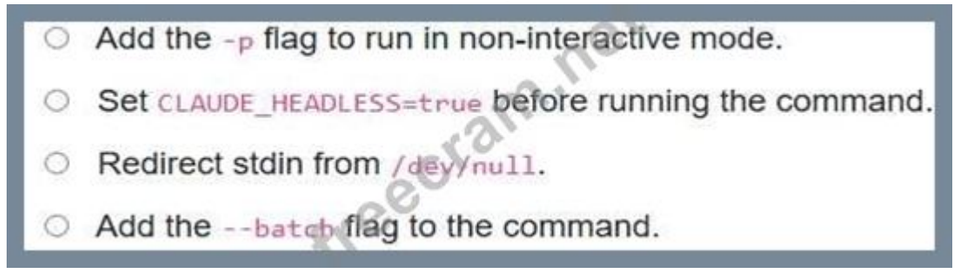
NEW QUESTION: 72
When applying few-shot prompting to improve extraction quality for ambiguous scenarios, what is the optimal number of examples to provide?
A. 1 example to establish the JSON format.
B. 2 to 4 examples, covering standard and edge cases.
C. 8 to 10 examples to cover all possible variations.
D. 15 or more examples to thoroughly fine-tune the model's reasoning.
Answer: (SHOW ANSWER)
The golden rule of few-shot prompting is to provide 2-4 examples. Fewer than 2 does not establish a pattern, while providing more than 6 examples bloats the prompt without adding proportional benefit.

NEW QUESTION: 73
When implementing multi-pass review and self-evaluation architectures, which of the following practices are recommended to ensure reliability and accurate routing? Choose 2 correct answers.
A. Using a second independent Claude instance to review generated code without the generator's reasoning context.
B. Running verification passes where the model self-reports confidence alongside each finding to enable calibrated review routing.
C. Having the same session review its newly generated code to guarantee maximum preservation of architectural context.
D. Forcing consensus across 3 monolithic runs to suppress false positives in large multi-file PRs.
Answer: (SHOW ANSWER)
Independent review instances without prior reasoning context are much more effective at catching subtle issues. Furthermore, running verification passes where the model outputs field-level confidence scores alongside findings enables calibrated review routing when mapped to a labeled validation set. Same-session reviews create confirmation bias, and monolithic consensus runs suppress true positive detection.

NEW QUESTION: 74
When implementing custom subagents to delegate tasks, how do these subagents access and inherit your configured project skills?

A. Custom subagents automatically inherit all skills loaded by the parent coordinator agent.
B. Built-in agents (like Explorer) automatically cascade project skills to newly spawned subagents.
C. Subagents do not automatically see skills; skills must be explicitly listed in the skills : field of the custom subagent's markdown definition.
D. Skills are passed to subagents dynamically by declaring them as parameters in the Task tool call.
Answer: (SHOW ANSWER)
Subagents start with a clean context and do not automatically inherit skills from the main conversation. To use a skill, it must be explicitly defined in the custom subagent's configuration frontmatter using the 'skills' field.

NEW QUESTION: 75
Your CI pipeline script runs claude "Analyze this PR" but the job hangs indefinitely. What is the correct approach to run Claude Code in an automated pipeline?



A. Option A
B. Option B
C. Option C
D. Option D
Answer: (SHOW ANSWER)
The -p (or -print) flag is the documented way to run Claude Code in non-interactive mode. It processes the prompt, outputs the result to stdout, and exits without waiting for user input, which is exactly what CI/CD pipelines require to prevent hanging.

NEW QUESTION: 76
In a Continuous Integration (CI) pipeline, you run a script where a single Claude session generates a block of code and then immediately reviews its own changes in the same session. Why is this considered an architectural anti-pattern?
A. The reviewer retains the reasoning context from the code generation phase, creating confirmation bias and reducing the likelihood of catching subtle issues.
B. It exceeds the maximum token limit for the context window, causing the session to silently truncate the generated code.
C. Same-session reviews trigger rate limiting automatically, breaking the CI/CD pipeline.
D. The Message Batches API does not support multi-turn tool calling within a single request.
Answer: (SHOW ANSWER)
Same-session self-review is an anti-pattern because the model retains its original reasoning context, creating a blind spot. Independent review instances without prior reasoning context are much more effective at objectively evaluating code.

Valid CCA-F Dumps shared by EduDump.com for Helping Passing CCA-F Exam! EduDump.com now offer the **newest CCA-F exam dumps**, the EduDump.com CCA-F exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com CCA-F dumps with Test Engine here:
https://www.edudump.com/exams/Anthropic/CCA-F/premium/ (**112** Q&As Dumps, **35%OFF** **Special Discount Code:** **freecram**)

NEW QUESTION: 77
When configuring Claude Code for a new project, you want to establish team-wide TypeScript strict mode conventions, while keeping your personal Vim keybindings private. What is the correct architectural approach?

- Place all configurations in the project-level `.claude/CLAUDE.md` file.
- Place the TypeScript conventions in the project-level `.claude/CLAUDE.md` file, and your Vim bindings in the user-level `~/.claude/CLAUDE.md` file
- Place the TypeScript conventions in a directory-level `src/api/CLAUDE.md` file, and Vim bindings in `CLAUDE.local.md`.
- Define team standards in `.mcp.json` and personal preferences in the root `CLAUDE.md`.

freecram.net

A. Option A
B. Option B
C. Option C
D. Option D
Answer: (SHOW ANSWER)
Claude Code utilizes a configuration hierarchy. Team-wide standards that should be shared via version control belong in the project- level .claude/CLAUDE.md file. User-level configuration (-/.claude/CLAUDE.md) is for personal preferences and is not shared with the team.

NEW QUESTION: 78
Why is it heavily recommended to use Agent SDK hooks rather than prompt instructions to enforce critical business rules in a application?
A. Hooks inherently consume zero tokens from the context window.
B. Hooks provide deterministic guarantees, whereas prompt instructions rely on probabilistic compliance and carry a non-zero failure rate.
C. Hooks automatically implement strict JSON schema validation for all tool_use calls.
D. Prompt instructions cannot be modified without restarting the entire MCP server architecture.
Answer: (SHOW ANSWER)
The core distinction is that hooks run as programmatic code, providing deterministic, 100% reliable enforcement for business policies. Prompts rely on the LLM's probabilistic pattern matching, meaning the model can and occasionally will ignore critical instructions.

NEW QUESTION: 79
A engineering team wants to share formatting rules, testing guidelines, and preferred libraries across their entire repository. Where is the correct location to configure these team-wide standards in Claude Code?
A. In the user-level configuration file located at ~/. claude/CLAUDE . md.
B. In the project-level configuration file located at . claude/CLAUDE. md or the project root.
C. In a directory-level configuration file located at tests/CLAIJDE .md.
D. In the global .mcp.json file.
Answer: (SHOW ANSWER)
Team-wide standards that should be shared via version control belong in the project-level .claude/CLAUDE.md file. User-level config is for personal, unshared preferences, and directory-level is scoped only to specific modules.

NEW QUESTION: 80
Which of the following describe valid core components and capabilities provided by an MCP server integration? Choose 2 correct answers.

Freecram.net

A. Exposing Tools that allow the agent to execute actions or mutations on external systems.
B. Exposing Resources that provide static or parameterized data catalogs to the client.
C. Automatically training custom embedding models for proprietary semantic search pipelines.
D. Persisting cross-session conversation memory natively inside the MCP client.
Answer: (SHOW ANSWER)
MCP servers expose three primary components to extend capabilities: Tools (executable functions/actions), Resources (fetchable data/catalogs), and Prompts (pre-built templates). They do not train embedding models or natively handle long-term agent memory persistence.

NEW QUESTION: 81
A development team at quicktechie.com wants to enforce strict security standards specifically whenever a developer modifies infrastructure-as-code files. Which YAML frontmatter configuration correctly applies these rules to all Terraform files within the repository?

```
scope: ["*.tf"]

target-files: "terraform/**/*"

paths: ["terraform/**/*"]

apply-to: ["**/*.terraform"]
```

freedram.net

A. Option A
B. Option B
C. Option C
D. Option D
Answer: (SHOW ANSWER)
Path-specific rules in the .claude/rules/ directory use a YAML frontmatter 'paths' field containing glob patterns (e.g., paths: This ensures the rules load conditionally only when editing matching files.

NEW QUESTION: 82
When designing an autonomous customer support agent, you must explicitly program the conditions under which it stops attempting to solve the problem and escalates to a human. Which of the following are valid, objective escalation triggers? Choose 2 correct answers.
A. The agent detects a policy gap where no existing rule covers the customer's specific situation.
B. The agent assesses its own confidence level to be below 0.6 out of 1.0.
C. The customer explicitly demands to speak with a human representative.
D. The customer expresses high frustration and uses negative sentiment words.
Answer: (SHOW ANSWER)
Valid escalation triggers include policy gaps, explicit customer requests for humans, business thresholds being exceeded, and capability limits. Self-reported confidence scores and sentiment analysis are unreliable anti-patterns for escalation decisions.

NEW QUESTION: 83
You are designing a security-sensitive skill that audits code for vulnerabilities. You want to guarantee that this skill can only analyze files and cannot accidentally modify the codebase. How is this enforced?
A. By adding a strong warning in the SKILL.md markdown text: "DO NOT USE WRITE TOOLS".
B. By configuring allowed-tools: Read, Grep, Glob in the skill's YAML frontmatter.
C. By executing the skill inside a stateless Docker container using the Bash tool.

D. By appending a - -read -only flag when invoking the skill via the terminal.
Answer: (SHOW ANSWER)
To strictly restrict which tools Claude can use when a skill is active, you must configure the 'allowed-tools' property in the SKILL.md YAML frontmatter. Relying on prompt-based instructions in the markdown text is probabilistic and unsafe for security guarantees.

NEW QUESTION: 84
To integrate an MCP server providing database tools for a team, you must configure it so that all developers share the integration but do not leak secrets into version control. What is the correct configuration approach?
A. Configure the server in the user-scoped ~/. claude. json file to keep credentials local.
B. Configure the server in the project-scoped .mcp. j son file using environment variable expansion like ${DB_TOKEN}.
C. Hardcode an encrypted API token directly in the .mcp.json file.
D. Store the credentials in an ignored secrets. json file referenced by CLAUDE. md.
Answer: B (LEAVE A REPLY)
Shared team MCP integrations belong in the project-level .mcp.json file. To prevent credential leaks in version control, you must use environment variable expansion (e.g., ${DB_TOKEN}) so secrets remain in the local environment and not in the code.

NEW QUESTION: 85
When passing context between agents in a multi-agent orchestration, why is it highly recommended to use structured data formats?
A. To automatically trigger JSON schema validation by the Claude API.
B. To separate content from metadata (such as source URLs, document names, and page numbers) to preserve proper attribution.
C. To compress the token count of the prompt by 50% using the Message Batches API.
D. To bypass the maximum token limit of the subagent's isolated context window.
Answer: (SHOW ANSWER)
Using structured data formats to separate content from metadata (like source URLs, document names, and publication dates) when passing context between agents ensures that downstream synthesis agents preserve proper source attribution and information provenance.

NEW QUESTION: 86
Your agent achieves a low first-contact resolution rate because it attempts to autonomously handle complex policy exceptions, while simultaneously escalating straightforward tasks. What is the most effective approach to improve this escalation calibration?
A. Add explicit escalation criteria to the system prompt with few-shot examples demonstrating when to escalate versus resolve autonomously.
B. Deploy a separate classifier model to predict which requests need escalation before the main agent begins.
C. Require the agent to self-report a confidence score (1-10) and route to humans when confidence falls below 7.
D. Implement sentiment analysis to detect frustration levels and automatically escalate when they are high.
Answer: (SHOW ANSWER)
Adding explicit escalation criteria with few-shot examples directly addresses unclear decision boundaries. Self-reported confidence scores are poorly calibrated, sentiment analysis does not measure complexity, and a separate classifier is over-engineered before trying prompt optimization.

NEW QUESTION: 87
An AI extraction system processes thousands of financial documents. The dashboard displays an impressive 96% overall aggregate accuracy. However, downstream teams complain about frequent errors. What is the most likely oversight in the review and monitoring design?
A. The system is using prompt chaining instead of dynamic adaptive decomposition.

B. Aggregate accuracy metrics can mask severe per-document-type failures (e.g., invoices failing at 70% while receipts succeed at 99%).
C. The extraction is using JSON schema tool_use which guarantees semantic correctness but fails structural checks.
D. The system lacks a PostToolUse hook to block large financial transactions.
Answer: (SHOW ANSWER)
A heavily tested exam concept is that aggregate accuracy metrics can hide critical failures in specific subsets of data (stratified metrics). Tracking accuracy per document type and field is essential before automating high-confidence extractions.

NEW QUESTION: 88
A customer support agent built with the Claude Agent SDK needs to correctly identify when to resolve a case autonomously versus when to trigger human escalation. Which of the following represents an architectural anti-pattern for triggering escalation?
A. Escalating when the agent detects a policy gap not covered by its existing knowledge base.
B. Escalating immediately when the customer explicitly demands to speak to a human representative.
C. Escalating based on negative customer sentiment or self-reported low model confidence scores.
D. Escalating when a deterministic programmatic hook blocks a business operation that exceeds the agent's permission limits.
Answer: (SHOW ANSWER)
Using sentiment analysis or the LLM's self-reported confidence score for escalation is a heavily tested anti-pattern. Sentiment does not correlate with task complexity (e.g., an angry customer may just need a simple password reset), and model confidence scores are poorly calibrated for routing decisions.

NEW QUESTION: 89
To ensure an agent strictly outputs a specific JSON structure without generating conversational text, and guarantees that a specific tool is invoked, which tool_choice configuration is required?



A. Option A
B. Option B
C. Option C
D. Option D
Answer: C (LEAVE A REPLY)
Forcing a specific tool with tool_choice: {"type": "tool", "name": "..."} ensures a particular tool runs, which guarantees schema compliance for extraction tasks. 'auto' allows the model to decide whether to use a tool, and 'any' forces a tool but allows the model to choose which one.

NEW QUESTION: 90
In a multi-agent orchestration, the synthesis agent occasionally misuses a web search tool instead of synthesizing the provided findings. How should you prevent this cross-role misuse according to the principle of scoped tool access?
A. Implement a PostToolUse hook to silently drop web search results when executed by the synthesis agent.

B. Remove the web search tool from the synthesis agent's allowed tool set entirely.
C. Add a capitalized warning in the system prompt instructing the synthesis agent to ignore web search capabilities.
D. Use the fork session command to isolate the web search tool execution.
Answer: (SHOW ANSWER)
Agents with tools outside their specialization tend to misuse them. The most robust architectural fix for cross-role tool misuse is scoped tool access: explicitly removing the irrelevant tool from that specific subagent's allowed array so it strictly focuses on its assigned role.

NEW QUESTION: 91
You are designing custom MCP tools for a quicktechie.com backend system. Why is it considered an architectural anti-pattern to design a single, monolithic process_data tool with 20 optional parameters for extraction, summarization, and verification?
A. The Claude API strictly limits individual tool schemas to a maximum of 5 parameter properties.
B. Massive generic tools degrade tool selection reliability; they should be split into purpose-specific tools with defined input/output contracts.
C. MCP servers do not support passing complex nested JSON objects required by optional parameters.
D. It triggers automatic prompt caching invalidation upon every execution.
Answer: (SHOW ANSWER)
Splitting generic tools into purpose-specific tools (e.g., extract_data_points, summarize_content, verify_claim) provides defined input/output contracts. Massive generic tools with dozens of optional parameters force the LLM to make overly complex configuration decisions, significantly degrading reliability and increasing hallucination risks.

Valid CCA-F Dumps shared by EduDump.com for Helping Passing CCA-F Exam! EduDump.com now offer the **newest CCA-F exam dumps**, the EduDump.com CCA-F exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com CCA-F dumps with Test Engine here:
https://www.edudump.com/exams/Anthropic/CCA-F/premium/ (**112** Q&As Dumps, **35%OFF** **Special Discount Code:** **freecram**)

NEW QUESTION: 92
A quicktechie.com engineer pauses an active Claude Code investigation session, makes several manual code modifications to the project files in a separate IDE, and then resumes the Claude session. What is the recommended practice to ensure reliable continued analysis?
A. Use the /compact command to clear Claude's memory of the old file states automatically.
B. Explicitly inform the resumed session about the specific file changes for targeted re-analysis rather than requiring full re-exploration.
C. Rely on Claude's background file system watcher to automatically update the context window.
D. Invoke a PreToolUse hook to block Claude from reading any newly modified files.
Answer: B (LEAVE A REPLY)
When resuming sessions after external code modifications, it is critical to explicitly inform the agent about the changes to previously analyzed files. This enables targeted re-analysis, avoiding context degradation that occurs when Claude relies on stale memories of files that have since been changed.

NEW QUESTION: 93
How can an architect split a massive, monolithic CLAUDE. md file to selectively include external standards files based on the current module?
A. By utilizing the #include macro inside the file's YAML frontmatter.
B. By using the @import syntax to reference and pull in external markdown files.
C. By compiling a PreToolUse hook to merge files dynamically before generation.
D. By specifying an allowed-tools array pointing to specific directories.

Answer: B (LEAVE A REPLY)
The @import syntax is used in CLAUDE.md files for referencing external markdown files (e.g., @import ./rules/api-design.md). This enables modular configuration, preventing a single CLAUDE.md file from becoming unmaintainable and bloated.

NEW QUESTION: 94
When designing escalation patterns for an autonomous customer support agent, which of the following represent architectural anti- patterns that should be avoided? Choose 2 correct answers.
A. Relying on the model's self-reported confidence score to determine if human review is needed.
B. Escalating cases immediately when a customer explicitly requests to speak to a human.
C. Escalating cases automatically based on high levels of negative customer sentiment.
D. Escalating cases when the agent detects a policy gap not covered by its instructions.
Answer: (SHOW ANSWER)
Using self-reported model confidence scores and customer sentiment are heavily tested anti-patterns. Model confidence is poorly calibrated for routing, and sentiment does not correlate with task complexity. Explicit human requests and policy gaps are valid escalation triggers.

NEW QUESTION: 95
What is the primary architectural difference between handling a transient error and handling a valid empty result within an MCP tool's response payload?
A. Transient errors use isError: true with a specific errorCategory, while empty results use isError: false and return an empty data array.
B. Transient errors must bypass the MCP client and communicate directly via WebSockets, whereas empty results use standard HTTP.
C. Transient errors require the agent to spawn a subagent for resolution, whereas empty results are handled in the main thread.
D. There is no difference; both scenarios should return a generic HTTP 404 response.
Answer: (SHOW ANSWER)
An empty result signifies the tool executed flawlessly but found nothing, meaning isError must be false. A transient error (like a network timeout) means the tool failed to execute its primary purpose, so it must return isError: true along with contextual metadata like the errorCategory.

NEW QUESTION: 96
When implementing path-specific rules for conditional convention loading in Claude Code, which of the following statements accurately describe the architecture? Choose 2 correct answers.
A. Path-specific rules are defined using YAML frontmatter with a paths array containing glob patterns.
B. Path-specific rule files must be stored exclusively in the user-level ~/. claude/rules/ directory to prevent team-wide conflicts.
C. They are ideal for applying coding conventions to file types (like **/*. test. tsx) that are distributed across many different subdirectories.
D. Path-specific rules automatically execute a PreToolUse hook to format files before they are loaded into context.
Answer: (SHOW ANSWER)
Path-specific rules utilize YAML frontmatter 'paths' arrays with glob patterns (A) and are the recommended approach for applying conventions to files spread across the codebase by type (C). They are typically stored in project-level .claude/rules/, not just user-level, and they do not natively execute hooks.

NEW QUESTION: 97
You are building a classifier to categorize customer reviews, but the model struggles with ambiguous scenarios like sarcasm and mixed sentiments. You decide to use few-shot prompting. What is the recommended best practice for this scenario?
A. Provide exactly 1 example of a standard positive review to establish the JSON format.
B. Provide 8 to 10 examples to ensure every possible sentiment category is covered comprehensively.

C. Provide 2 to 4 examples, ensuring that at least one example specifically addresses an ambiguous edge case like sarcasm.
D. Avoid few-shot prompting and instead use a detailed system prompt explaining the exact definition of sarcasm.
Answer: (SHOW ANSWER)
The optimal number of few-shot examples for ambiguous tasks is 2-4. Providing more than 6 examples bloats the prompt without adding proportional value. It is critical that at least one of these examples covers an edge case or ambiguous scenario to establish the expected reasoning pattern.

NEW QUESTION: 98
In an extended exploration session, which built-in Claude Code command is explicitly used to reduce context usage when the window fills with verbose discovery output?
A. / clear
B. / compact
C. - - resume
D. / init
Answer: (SHOW ANSWER)
The /compact command is specifically designed to compress conversation history and reclaim space in the context window while preserving essential knowledge. /clear completely wipes the context, which is not desirable if you want to retain project knowledge.

NEW QUESTION: 99
When orchestrating complex workflows using the Claude Agent SDK, in which scenario is dynamic adaptive decomposition preferred over fixed sequential prompt chaining?
A. When executing open-ended investigation tasks where intermediate discoveries dictate the next required steps.
B. When performing predictable, multi-aspect code reviews where the sequence of analysis is entirely static.
C. When validating a strictly structured JSON extraction against predefined business rules.
D. When executing a daily technical debt audit across a predefined list of target directories.
Answer: (SHOW ANSWER)
Dynamic adaptive decomposition is the optimal strategy for unpredictable, open-ended tasks (like investigations) where the agent must decide its next steps based on intermediate findings. Prompt chaining (fixed sequential pipelines) is better suited for predictable, linear workflows with static sequences.

NEW QUESTION: 100
You need Claude to output raw JSON directly to the user interface of your application, without any conversational preamble or markdown formatting. Which prompt engineering technique achieves this?
Freecram.net
A. Add "DO NOT USE MARKDOWN" to the end of the user prompt.
B. Combine Assistant Message Prefilling ```json with and use ```as a stop sequence.
C. Set max_tokens to exactly match the length of the JSON string.
D. Provide a 10-shot example list of raw JSON outputs.
Answer: (SHOW ANSWER)
Assistant Message Prefilling with ```json tricks the model into thinking it has already started a code block. Providing sequence immediately terminates generation once the JSON payload completes, resulting in raw JSON without explanatory text.

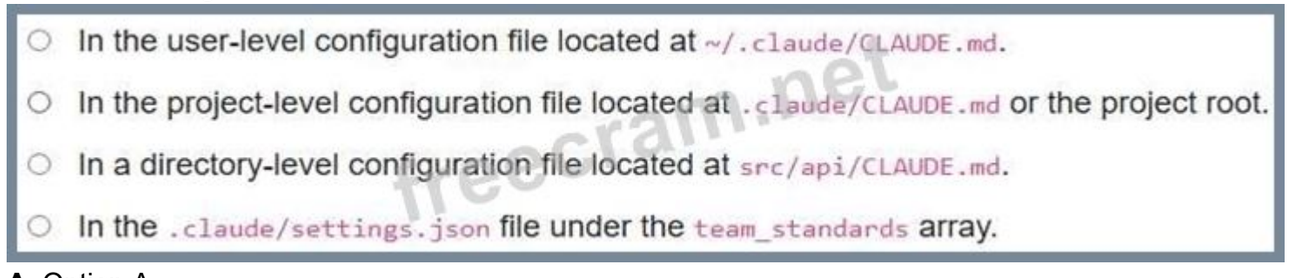
NEW QUESTION: 101
When building the raw tool-handling conversation loop with the API/SDK, Claude may return a ToollJse block. To properly continue the multi-turn agentic loop, what are the requirements for formatting and returning the result of the executed function to Claude? Choose 2 correct answers.
A. The result must be appended to the messages array inside a user role message containing a tool_result block.

B. The tool result block must include a that matches the ID generated by Claude's initial tool request.
C. The result must be passed back inside an assistant role message with an is_error'. false flag.
D. The tool result must strictly be a serialized JSON object; raw text strings will automatically trigger a loop termination.
Answer: (SHOW ANSWER)
To correctly supply a tool result back to Claude in the API/SDK flow, the application must construct a user message containing a content block of type 'tool_result'. This block must explicitly contain the 'tool_use_id' matching the original 'tool_use' request block from the assistant to map the result accurately.

NEW QUESTION: 102
You are reviewing the escalation logs for a customer support agent. The agent currently escalates cases whenever it detects an angry customer tone, even for simple tasks like password resets. What is the architectural flaw in this escalation design?
A. Sentiment does not equal task complexity; the agent should escalate based on objective criteria like policy gaps or explicit human requests, not emotion.
B. The agent should have processed a refund automatically to calm the customer down before escalating.
C. The sentiment threshold is set too low; it should only escalate if the customer uses specific profanity.
D. The agent lacks a PreToolUse hook to normalize the customer's text input before processing.
Answer: (SHOW ANSWER)
Escalating based on sentiment is an anti-pattern. An angry customer with a simple request (like a password reset) does not necessarily need a human agent. Escalation triggers should be based on objective criteria such as policy gaps, capability limits, business thresholds, or an explicit request for a human.

NEW QUESTION: 103
When designing an automated code review system for a large repository, you notice that developers are experiencing 'alert fatigue' because the AI flags too many non-issues and stylistic preferences. What is the most effective prompt engineering technique to reduce these false positives?
A. Instruct the model to be thorough and flag everything suspicious to ensure no bugs are missed.
B. Define explicit, measurable review criteria, such as flag functions exceeding 50 lines of code, instead of relying on vague instructions.
C. Set the model's temperature to 0.0 to ensure deterministic output and eliminate all false positives.
D. Provide at least 15 few-shot examples covering every possible coding style in the repository.
Answer: (SHOW ANSWER)
Explicit, measurable criteria produce consistent, actionable results that build developer trust and reduce false positives. Vague instructions like 'be thorough' lead to over-flagging and alert fatigue. Temperature adjustments do not fix vague criteria, and providing 15 few-shot examples bloats the prompt without proportional benefit.

NEW QUESTION: 104
A development team wants to enforce TypeScript strict mode, specific ESLint rules, and API design standards for all developers working on a shared repository. Where is the correct location to configure these team-wide standards in Claude Code?



A. Option A
B. Option B
C. Option C

D. Option D
Answer: (SHOW ANSWER)
Team-wide standards that should be shared via version control belong in the project-level .claude/CLAUDE.md file. User-level config is for personal preferences (like dark theme or vim bindings) and is not shared. Directory-level config is scoped to specific module folders.

NEW QUESTION: 105
Which of the following strategies effectively manage context degradation in long-running exploratory sessions? Choose 2 correct answers.
A. Continuously summarizing 'case facts' blocks to free up space.
B. Delegating verbose exploration output to subagents to keep the main coordinator's context clean.
C. Persisting key findings and intermediate states in external scratchpad files that survive context resets.
D. Increasing the model's temperature parameter to encourage creative context recovery.
Answer: (SHOW ANSWER)
Delegating verbose exploration to subagents (who return only concise summaries to the main thread) and using external scratchpad files to persist key findings across context boundaries effectively mitigate context degradation. Summarizing 'case facts' defeats their purpose, and temperature adjustments do not fix context limits.

NEW QUESTION: 106
A developer needs to perform a monolith-to-microservices migration spanning 45+ files. Why is Plan Mode recommended over Direct Execution for this task?
A. It uses the Message Batches API to reduce token costs by 50 percent during execution.
B. It enables safe codebase exploration and design to determine architectural approaches before making costly file modifications.
C. It automatically forks the session into a persistent subagent.
D. It bypasses the token limit context window by executing across multiple continuous sessions automatically.
Answer: (SHOW ANSWER)
Plan mode is designed for complex tasks involving large-scale changes, multiple valid approaches, and architectural decisions. It enables safe codebase exploration and design before committing to changes, which prevents costly rework compared to Direct Execution.

Valid CCA-F Dumps shared by EduDump.com for Helping Passing CCA-F Exam! EduDump.com now offer the **newest CCA-F exam dumps**, the EduDump.com CCA-F exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com CCA-F dumps with Test Engine here:
https://www.edudump.com/exams/Anthropic/CCA-F/premium/ (**112** Q&As Dumps, **35%OFF** **Special Discount Code:** **freecram**)

NEW QUESTION: 107
When a get_customer tool lookup returns multiple potential matches for a user, what is the required pattern for ambiguity resolution?
A. Select the most recently active account using a heuristic.
B. Instruct the agent to ask the user for additional identifiers for clarification.
C. Automatically escalate the session to a human supervisor.
D. Return a structured error payload and permanently terminate the loop.
Answer: (SHOW ANSWER)
When multiple matches are found, relying on heuristics to guess the correct user can lead to security and billing risks. The correct reliability decision is instructing the agent to ask the user for additional clarifying identifiers rather than using heuristic selection.

NEW QUESTION: 108
When formatting few-shot examples, what is the most important structural rule to follow to ensure the model produces the desired output?
A. Format consistency: all examples must follow the identical output structure.
B. Randomization: examples should be presented in different formats to increase model adaptability.
C. Conciseness: output examples should be truncated to save tokens.
D. Exclusion: never use XML tags around the examples.
Answer: (SHOW ANSWER)
Format consistency is crucial in few-shot prompting. All examples must follow the identical output structure so the model recognizes the strict pattern it needs to emulate in its final generation.

NEW QUESTION: 109
Which statements correctly describe the relationship between JSON schemas and validation loops in structured extraction? Choose 2 correct answers.
A. tool_use eliminates schema syntax errors, greatly reducing the need for syntax-focused retries.
B. Semantic errors (e.g., values placed in wrong fields) are prevented entirely by strict JSON schemas.
C. Validation loops are necessary to catch semantic errors, such as line items failing to sum to the extracted total.
D. Validation retries are most effective when the required information is completely missing from the document.
Answer: (SHOW ANSWER)
Tool use guarantees structural compliance (eliminating JSON syntax errors), but it does not guarantee semantic correctness. Therefore, validation loops are still absolutely necessary to catch semantic errors like bad mathematical summations or values placed in the wrong fields.

NEW QUESTION: 110
Which of the following scenarios represent appropriate, architecturally sound use cases for Agent SDK hooks? Choose 2 correct answers.
Freecram.net
A. Normalizing heterogeneous numeric status codes from different MCP tools before appending them to the conversation context.
B. Guaranteeing that a model-extracted JSON object contains no semantic reasoning errors.
C. Enforcing the conversational tone, politeness, and persona the agent should adopt.
D. Intercepting outgoing tool calls to deterministically block policy-violating actions, such as unauthorized financial transactions.
Answer: A,D (LEAVE A REPLY)
Hooks are ideal for programmatic interventions like data normalization CPostToolUse*) and policy enforcement/interception ('PreToolUse'PPostToolUse'). Conversational tone is best handled by prompts, and hooks cannot inherently guarantee the semantic accuracy of an LLM's complex reasoning.

NEW QUESTION: 111
During an extended codebase exploration session in Claude Code, context degradation begins to occur. What is the most effective strategy to persist key findings across context boundaries?
A. Use the - -json-schema flag to aggressively compress output.
B. Have agents maintain scratchpad files to record key findings and reference them in subsequent questions.
C. Increase the model's temperature parameter to recall lost context.
D. Continuously re-feed the entire codebase into the prompt every 5 turns.
Answer: B (LEAVE A REPLY)
Context degradation in extended sessions causes models to lose track of earlier details. Using scratchpad files to persist critical intermediate state externally ensures that key findings survive context compression and session boundaries.

NEW QUESTION: 112
You want to expose a static content catalog, such as a large database schema or internal API specification, to an MCP client without requiring Claude to make exploratory tool calls. Which MCP feature is specifically designed for this?
A. MCP Prompts
B. MCP Resources
C. MCP Subagents
D. MCP Hooks
Answer: (SHOW ANSWER)
MCP Resources are used to expose static or templated data catalogs (like documentation, static data, or schemas) that clients can fetch directly to inject into prompts, saving context and eliminating the need for exploratory tool actions.

NEW QUESTION: 113
You have testing conventions that must be uniformly applied, but your test files are distributed alongside source code files (e.g., Button. test. tsx next to Button. tsx). Why is creating a rule file in . claude/rules/ better than modifying the root CLAUDE. md?
A. The root CLAUDE .md cannot process testing frameworks like Jest or Vitest.
B. Consolidating all conventions in the root CLAUDE .md requires Claude to infer which sections apply to which files, making the application of rules probabilistic rather than deterministic.
C. Rules in . claude/rules/ can natively execute shell commands prior to loading, unlike the root configuration.
D. The root CLAUDE .md is strictly reserved for environment variable expansion.
Answer: (SHOW ANSWER)
Consolidating all conventions in a root CLAUDE.md file forces the model to rely on inference to determine which sections apply to the current context. Using .claude/rules/ with glob patterns applies conventions deterministically based on file paths, regardless of directory location.

NEW QUESTION: 114
You are managing context in a long-running Claude Code session that has accumulated a large amount of exploratory tool calls. You want to free up context space without completely losing the memory of what was previously worked on. Which techniques are recommended? Choose 2 correct answers.
A. Use the /compact command to summarize conversation history and reclaim context space.
B. Use the /clear command to erase all history and rely on the model's pre-trained knowledge.
C. Save key findings to a scratchpad file and delegate verbose exploration to subagents.
D. Delete older messages from the conversation array manually to ensure hard truncation.
Answer: (SHOW ANSWER)
The recommended mitigation strategies for context degradation in extended sessions include using the Icompact command to compress history, persisting key state to scratchpad files, and using subagent delegation to keep the coordinator context clean. [clear removes all memory, which contradicts the goal of retaining previous work memory.

NEW QUESTION: 115
A quicktechie.com developer is designing a schema to extract contract terms. Source documents sometimes lack a 'termination_fee' clause. How should the schema be designed to prevent the model from fabricating a value?
A. Add a capitalized warning in the system prompt to avoid hallucination.
B. Design the field as strictly required and let the model infer a standard default.
C. Design the field as optional or nullable, such as ["number", "null"].
D. Set tool_choice. :"auto" so the model can skip the tool entirely.

Answer: (SHOW ANSWER)
Designing schema fields as optional (nullable) when source documents may not contain the information prevents the model from fabricating values to satisfy required fields. If a field is strictly required but the data does not exist, the model may hallucinate a value.

NEW QUESTION: 116
Why is it considered an architectural anti-pattern to configure the exact same Claude session to review its newly generated code?
A. It exceeds the maximum token limit of the context window during the generation phase.
B. The reviewer retains the reasoning context and assumptions from the generation phase, creating confirmation bias.
C. The Message Batches API does not support synchronous self-review workflows.
D. Tools invoked during the first pass are permanently locked and cannot be reused in the review pass.
Answer: (SHOW ANSWER)
Same-session self-review is an anti-pattern because the model retains the context and reasoning from the generation phase, leading to a blind spot or confirmation bias. Independent review instances with fresh contexts are much more effective at objectively evaluating code.

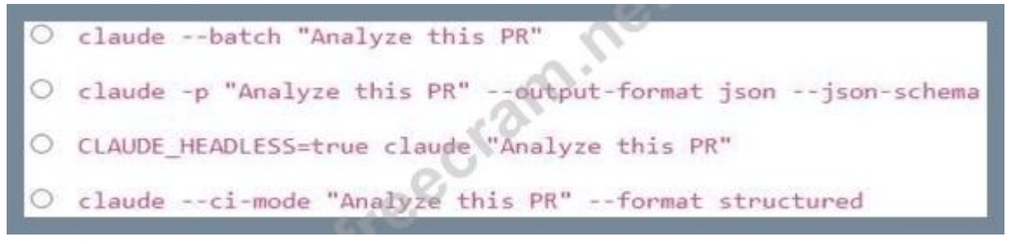
NEW QUESTION: 117
To extract metadata from a highly variable set of invoices, you use an enum field for document_type. How should you handle unexpected document types to maintain extraction quality?
A. Include an other enum value paired with a detail string field for extensible categorization.
B. Omit the enum constraint entirely and use an open-ended string field.
C. Implement a PreToolUse hook to drop unrecognized document types.
D. Provide an enum containing only known types and rely on the model to find the closest match.
Answer: A (LEAVE A REPLY)
To handle ambiguous or unexpected data during structured extraction, schemas should include an 'other' (or 'unclear') enum value paired with a detail string field. This prevents the model from forcing unexpected data into incorrect predefined categories.

NEW QUESTION: 118
How does utilizing path-scoped rules in Claude Code impact the session's context window?
A. It permanently reserves a block of 1024 tokens for rule processing, reducing the available context for the developer.
B. It loads the rules conditionally only when matching files are being edited, which reduces irrelevant context and saves token usage.
C. It automatically compresses the conversation history using the /compact command whenever a new rule is loaded.
D. It bypasses the context limit entirely by executing the rules in a background subprocess.
Answer: (SHOW ANSWER)
Path-scoped rules defined in the .claude/rules/ directory are loaded conditionally. They only activate and consume context window tokens when matching files are being edited. This targeted activation reduces irrelevant context and optimizes token usage.

NEW QUESTION: 119
During a highly complex, multi-phase refactoring task, the context window risks becoming exhausted by verbose discovery output before any code is modified. What built-in mechanism is designed to mitigate this during the planning phase?
A. The /compact command run automatically inside a PreToolUse hook.
B. The Message Batches API.
C. The Explore subagent, which isolates verbose discovery output and returns summarized findings to the main conversation.
D. The --json-schema CLI flag, which forces discovery output to be heavily compressed.

Answer: (SHOW ANSWER)
The Explore subagent is used for isolating verbose discovery and exploration output during complex planning phases. It performs the heavy lifting in an isolated context and returns a summary, preserving the main conversation context window.

NEW QUESTION: 120
You are integrating Claude Code into a Continuous Integration (CI) pipeline to run automated security analysis. Which flag combination must be used to ensure the automated job runs without hanging and produces machine-parseable output?



A. Option A
B. Option B
C. Option C
D. Option D
Answer: (SHOW ANSWER)
For CI/CD pipelines, the `-p` (or `--print`) flag is strictly required to run non-interactively. Combined with `--output-format json` and `--json-schema`, it provides the structured, predictable shape necessary for automated parsing and posting inline PR comments.

NEW QUESTION: 121
Which of the following are recommended architectural practices for preserving information provenance and handling uncertainty in a multi-agent system at quicktechie.com? Choose 2 correct answers.
A. Requiring subagents to output structured claim-source mappings (including source URLs, document names, and excerpts) that downstream agents must preserve.
B. Silently dropping contested findings to ensure the final synthesis report appears highly confident.
C. Structuring reports with explicit sections that distinguish well-established findings from contested ones, preserving the original methodological context.
D. Omitting publication dates from the metadata to save valuable token space in the coordinator's context window.
Answer: (SHOW ANSWER)
Recommended practices for information provenance include requiring structured claim-source mappings that pass through to the final output, and structuring reports to explicitly distinguish well-established findings from contested ones. Dropping contested findings or omitting temporal metadata are anti-patterns that destroy critical context.

Valid CCA-F Dumps shared by EduDump.com for Helping Passing CCA-F Exam! EduDump.com now offer the **newest CCA-F exam dumps**, the EduDump.com CCA-F exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com CCA-F dumps with Test Engine here:
https://www.edudump.com/exams/Anthropic/CCA-F/premium/ (**112** Q&As Dumps, **35%OFF** **Special Discount Code:** **freecram**)

NEW QUESTION: 122
Why is providing more than 6 examples in a few-shot prompt generally considered an anti-pattern?
A. It causes the model to ignore system prompts.

B. It disables the model's ability to use the tool_use block.
C. It bloats the prompt context window without adding proportional benefit.
D. It forces the Claude API to throw a schema validation error.
Answer: (SHOW ANSWER)
Providing too many examples (more than 4-6) wastes valuable context window tokens and bloats the prompt without offering a proportional increase in output quality or instruction adherence.

NEW QUESTION: 123
Which of the following scenarios are best suited for prompt chaining (fixed sequential pipelines) rather than dynamic adaptive decomposition? Choose 2 correct answers.
A. A predictable, multi-aspect security review where the sequence of code analysis steps is static and predefined.
B. An open-ended investigation into an undocumented legacy API where responses guide the next queries.
C. A data processing pipeline where raw text is summarized, then translated, and finally formatted into JSON.
D. A monolith-to-microservices architectural migration involving dozens of interacting dependencies.
Answer: (SHOW ANSWER)
Prompt chaining is ideal for predictable, linear workflows with static sequences of steps, such as multi-aspect code reviews following a checklist, or deterministic data processing pipelines (summarize -> translate -> format). Open-ended investigations and complex migrations require dynamic adaptive decomposition.

NEW QUESTION: 124
What is the recommended strategy for an agent to trace function usage across wrapper modules in an unfamiliar codebase?
A. Read all project files upfront to build a comprehensive context map in memory.
B. Use the Edit tool to temporarily inject logging statements into every module and observe the output.
C. Start with Grep to identify all exported names and entry points, search for each name across the codebase, and then use Read to trace specific flows.
D. Use Glob to find all . js and .ts files and concatenate them using Bash to preserve token sequence.
Answer: (SHOW ANSWER)
Building codebase understanding should be done incrementally. The recommended strategy is to start with Grep to find entry points or exported names, then search for those names across the codebase, and finally use Read to follow imports and trace flows. Reading all files upfront causes severe context pollution.

NEW QUESTION: 125
What is the only reliable, deterministic signal in the Claude Agent SDK to evaluate whether an agentic loop should continue executing tools or terminate?
A. Parsing the assistant's natural language text output for completion keywords like 'task complete'.
B. Inspecting the stop_reason field of the response for tool_use versus end_turn.
C. Monitoring the total token count and exiting the loop when it reaches the context window limit.
D. Setting a hard maximum iteration cap of 5 loops to prevent infinite execution.
Answer: (SHOW ANSWER)
The stop_reason field is the only reliable signal for controlling the agentic loop. 'tool_use' indicates the loop must continue to execute tools, while 'end_turn' indicates Claude has finished reasoning and the loop should exit. Parsing natural language or setting arbitrary iteration caps are heavily tested anti-patterns.

NEW QUESTION: 126
When streaming responses that include tool calls in the Claude API, why might you observe buffering delays followed by bursts of text?
A. The API automatically waits for the entire tool output to be generated before streaming begins.
B. The API is performing a secondary security check on the generated tokens to prevent prompt injection.

C. The API validates JSON chunks and waits for complete top-level key-value pairs before sending them to the client.
D. The ContentBlockDelta events are rate-limited to 10 events per second.
Answer: (SHOW ANSWER)
The Anthropic API buffers partial JSON chunks to validate complete top-level key-value pairs against your tool's schema before transmitting them. This causes short delays followed by bursts. This validation can be bypassed by enabling fine-grained tool calling.

NEW QUESTION: 127
A quicktechie.com developer needs Claude to implement a distributed caching mechanism in an unfamiliar domain. To ensure edge cases are handled before implementation begins, the developer prompts: "Ask me 3 questions to surface potential design flaws or failure modes before you start." What is this iterative refinement technique called?
A. The Interview Pattern
B. The Generator-Verifier Pattern
C. Sequential Decomposition
D. Concrete Transformation
Answer: (SHOW ANSWER)
The Interview Pattern involves having Claude ask questions to surface design considerations, cache invalidation strategies, or failure modes that the developer may not have anticipated, refining the requirements before writing code.

NEW QUESTION: 128
To ensure PR feedback from Claude Code can be programmatically parsed and posted as inline comments by automation tools, which flags must be used?



A. Option A
B. Option B
C. Option C
D. Option D
Answer: (SHOW ANSWER)
To ensure machine-parseable, structured output in automated CI contexts, Claude Code must be executed with the --output-format json flag combined with the 'son-schema flag to strictly enforce the shape of the returned data.

NEW QUESTION: 129
When implementing a structured error response for an MCP tool, which property explicitly indicates to the agent whether it should attempt the exact same tool call again?
A. isError
B. errorCategory
C. isRetryable
D. tool_choice
Answer: (SHOW ANSWER)

The isRetryable boolean explicitly tells the agent whether retrying the same call might succeed (e.g., true for a transient timeout, false for a strict authorization failure or missing API key).

NEW QUESTION: 130

To keep the main agent's context clean while investigating a complex authentication flow, which architectural pattern is highly recommended?

Freecram.net

A. Spawn a subagent to isolate the verbose exploration output, returning only a focused summary to the main agent.
B. Use the Message Batches API to process the exploration synchronously.
C. Apply progressive summarization to the main conversation history every 2 turns.
D. Use the fork_session command to duplicate the main agent permanently.
Answer: (SHOW ANSWER)
Subagent delegation is the recommended pattern to isolate verbose exploration output (like reading multiple files and tracing functions) from the main coordinator. The subagent does the heavy lifting in its own isolated context and returns only a clean synthesis to the main agent.

NEW QUESTION: 131
When applying few-shot prompting to improve output consistency for a task with ambiguous boundaries, what is the optimal number of examples to provide?
A. 1 example.
B. 2 to 4 examples.
C. 8 to 10 examples.
D. 15 or more examples.
Answer: (SHOW ANSWER)
The golden rule of few-shot prompting is to provide 2-4 examples. Fewer than 2 does not firmly establish a pattern, while providing more than 6 examples generally bloats the prompt without adding proportional benefit.

NEW QUESTION: 132
When implementing a Model Context Protocol (MCP) server, what is the primary architectural purpose of defining an MCP Resource instead of an MCP Tool?
A. To expose executable functions that perform side-effects or mutations on external backend systems.
B. To expose data catalogs or static content that can be fetched by the client and injected directly into prompts without requiring the agent to formulate a tool call.
C. To provide reusable conversation templates consisting of predefined user and assistant messages for the agent.
D. To manage environment variables and authentication secrets safely across the development team.
Answer: (SHOW ANSWER)
MCP Resources are used to expose data (like documents, schemas, or issue summaries) that clients can read and inject directly into prompts, bypassing the need for exploratory tool calls. Tools are for taking actions, and Prompts are for reusable conversation templates.

NEW QUESTION: 133
When implementing a structured data extraction pipeline using tool_use and JSON schemas, what exactly does the tool use mechanism guarantee?
A. Both structural compliance and semantic correctness of the extracted data.
B. Structural compliance only, requiring separate validation for semantic accuracy.
C. Automatic retry and self-correction whenever a required field is missing.
D. Elimination of hallucinations by forcing the model to strictly quote the source document.
Answer: (SHOW ANSWER)

Using tool_use with JSON schemas guarantees structural compliance (all required fields present, correct types, valid enum values), which eliminates JSON syntax errors. However, it does not guarantee semantic correctness (e.g., extracting the wrong date or line items not summing to the total). Semantic validation must be performed separately using business rules.

NEW QUESTION: 134
When implementing multi-pass review architectures for automated CI/CD pipelines, why is configuring the exact same Claude SDK session to review its newly generated code considered a major anti-pattern?
A. It causes the token limit of the context window to exceed its maximum capacity.
B. The Message Batches API does not support synchronous self-review workflows.
C. Tools invoked in the first pass are permanently locked and cannot be reused in the second pass.
D. The reviewer retains the reasoning context from the generation phase, creating confirmation bias and reducing the likelihood of catching subtle flaws.
Answer: (SHOW ANSWER)
Same-session self-review is ineffective because the model retains its original reasoning context and assumptions in the conversation history, creating a blind spot or confirmation bias. Using an independent review instance with fresh context objectively evaluates the code much better.

NEW QUESTION: 135
In a multi-agent system, a subagent times out querying a database. How should this error be propagated back to the coordinator agent?
A. Catch the timeout and return an empty result set marked as successful to keep the workflow moving.
B. Return structured error context including the failure type, what was attempted, and partial results.
C. Propagate the exception directly to a top-level handler to terminate the entire workflow.
D. Return a generic string {"error": "Operation failed"}.
Answer: (SHOW ANSWER)
Structured error propagation is critical. Returning structured error context (failure type, what was attempted, partial results) enables the coordinator to make intelligent recovery decisions. Silently returning empty results or terminating the whole workflow immediately are anti-patterns.

NEW QUESTION: 136
You are deep into a complex codebase analysis session and need to explore an alternative API integration approach without polluting your main session's context window. Which session management mechanism should you use?
A. Use the /clear command to reset the context window before exploring.
B. Use claude fork_session to create a branched session that inherits current context but diverges safely.
C. Use claude--resume to append the alternative exploration onto a previously saved checkpoint.
D. Create a scratchpad file and write the alternative approach directly to the local disk.
Answer: (SHOW ANSWER)
The fork_session feature creates a branched session for exploration that inherits the existing context but keeps subsequent interactions isolated, preventing exploration noise from polluting the main session's context.

Valid CCA-F Dumps shared by EduDump.com for Helping Passing CCA-F Exam! EduDump.com now offer the **newest CCA-F exam dumps**, the EduDump.com CCA-F exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com CCA-F dumps with Test Engine here:
https://www.edudump.com/exams/Anthropic/CCA-F/premium/ (**112** Q&As Dumps, **35%OFF** **Special Discount Code:** **freecram**)

NEW QUESTION: 137
When using tool_use with a strict JSON schema for data extraction, what does the Claude API explicitly guarantee?
A. Semantic correctness of the extracted data.
B. Structural compliance, effectively eliminating JSON syntax errors.
C. 100% factual accuracy of the extracted entities.
D. Elimination of the need for downstream data validation.
Answer: (SHOW ANSWER)
Using tool_use with JSON schemas guarantees structural compliance (all required fields present, correct types, valid enum values), which eliminates JSON syntax errors. However, it does not guarantee semantic correctness, as the model may still extract wrong values or place them in the incorrect fields.

NEW QUESTION: 138
Which of the following strategies are highly recommended for integrating Claude into a robust test generation workflow? Choose 2 correct answers.
A. Document testing standards, valuable test criteria, and available fixtures in the project-level CLAUDE .md.
B. Rely on aggregate test coverage metrics to determine if the generated tests are semantically correct.
C. Share specific test failure outputs with Claude to guide progressive improvement of the implementation.
D. Execute tests automatically within a PreToolUse hook to prevent Claude from saving failing code.
Answer: (SHOW ANSWER)
Documenting testing standards and fixtures in CLAUDE.md ensures baseline consistency across the project. Sharing test failures guides iterative refinement, enabling the model to self-correct. PreToolUse hooks block execution and cannot run tests automatically on unsaved code, while aggregate metrics mask semantic correctness issues.

NEW QUESTION: 139
You need to deterministically block any refund transaction that exceeds $500, escalating it to a human instead. What is the architecturally correct method to enforce this critical business rule using the Agent SDK?
A. Add a strong, capitalized warning in the agent's system prompt forbidding refunds over $500.
B. Provide 5 negative few-shot examples showing the agent successfully refusing high-value refunds.
C. Implement a hook (e.g., PostToolUse or PreToolUse) to programmatically intercept and block the tool call execution.
D. Set the agent's tool_choice parameter to auto so it can assess the financial risk dynamically.
Answer: (SHOW ANSWER)
Critical business rules requiring deterministic enforcement must use programmatic hooks. Relying on prompt instructions or few-shot examples is probabilistic and leaves a non-zero chance that the LLM will ignore the rule. Hooks run as code and provide 100% reliable enforcement.

NEW QUESTION: 140
What is the primary risk associated with applying progressive summarization techniques to a long-running customer support session?
Freecram.net
A. It bypasses the stop_reason logic of the agentic loop.
B. It frequently causes the Token Context Window to exceed maximum limits.
C. It silently condenses and loses critical specific details such as dates, exact amounts, and unique IDs.
D. It forces the agent to use tool_choice: "auto".
Answer: (SHOW ANSWER)
Progressive summarization compresses conversation history, which often silently destroys critical details (e.g., an exact $150.00 charge becomes a generic 'billing issue'). Case facts blocks are required to safely persist this data.

NEW QUESTION: 141
Using the @anthropic-ai/claude-agent-sdk, you want to programmatically narrow the set of tools available to a specific query loop to prevent the agent from making unintended modifications. Which configuration is used to restrict tools in the SDK?
A. Pass the allowedTools: [" Read , '" ' Glob"] array inside the options object of the query() function.
B. Inject a permissions: read-only tag into the top-level AgentDefinition.
C. Set tool choice: "none" to force the model to only use search tools.
D. Edit the . mcp. j son file to globally disable the write tools for the entire project.
Answer: (SHOW ANSWER)
In the @anthropic-ai/claude-agent-sdk, you restrict the toolset for a specific execution loop by passing the 'allowedTools' array (e.g., ["Read", "Glob")) in the options object of the 'query()' function. This maps to the CLI's -allowedTools flag.

NEW QUESTION: 142
Which of the following scenarios represent architecturally appropriate use cases for implementing hooks in a production Claude application? Choose 2 correct answers.
A. Normalizing tool outputs, such as date formats or status codes, before they are appended to the context window.
B. Guaranteeing that an extracted JSON object has no semantic reasoning errors.
C. Ensuring Claude maintains a conversational and friendly tone with customers.
D. Intercepting tool calls to deterministically block policy-violating actions before they execute.
Answer: (SHOW ANSWER)
Hooks are designed for programmatic, deterministic intervention during the tool lifecycle. Valid uses include PostToolUse hooks for normalizing data and PreToolUse hooks for blocking policy-violating actions. Hooks do not guarantee semantic accuracy and are not used for soft preferences like conversational tone, which are better handled via prompts.

NEW QUESTION: 143
You are implementing a validation-retry loop for a structured data extraction prompt. The extraction fails a business logic validation (e.g., line items do not sum to the total). Which techniques represent best practices for the follow-up retry prompt? Choose 2 correct answers.
A. Return a generic error message like There were errors in your extraction. Please try again.
B. Append specific validation errors, detailing which field failed and the nature of the discrepancy.
C. Include the original document and the failed extraction in the retry prompt to guide self-correction.
D. Increase the model temperature to 1.0 so it thinks of a different answer.
Answer: (SHOW ANSWER)
A successful validation-retry loop requires appending specific error details (not generic messages) to the prompt and supplying both the source document and the failed extraction. This provides the exact context the model needs to correct its semantic mistake.

NEW QUESTION: 144
An architect is designing an MCP tool that occasionally fails due to downstream rate limits. Which fields should be included in a well- designed structured error response to help the agent recover? Choose 2 correct answers.
A. isError: true
B. system_prompt_override
C. isRetryable: true
D. temperature _ adjustment
Answer: (SHOW ANSWER)

A structured error response should explicitly signal a failure using the 'isError' flag and indicate if the agent should attempt the call again using the 'isRetryable' boolean. It should also include the error category and context. System prompt overrides and temperature adjustments are not part of standard tool error metadata.

NEW QUESTION: 145
When designing the output structure for a synthesis agent that processes diverse data streams (e.g., financial statistics, news prose, and technical specifications), what is the recommended presentation pattern?
A. Convert all extracted data into a uniform, flat JSON array to simplify downstream parsing.
B. Render different content types appropriately (e.g., financial data as tables, news as prose, technical findings as structured lists) rather than forcing a uniform format.
C. Use the Code Execution tool to automatically generate an Excel workbook containing all data types.
D. Restrict the output strictly to natural language paragraphs to maintain conversational fluency.
Answer: B (LEAVE A REPLY)
To preserve the utility and readability of synthesized multi-source data, different content types should be rendered appropriately (e.g., financial data as tables, news as prose, technical findings as structured lists) rather than destructively converting everything into a single uniform format.

NEW QUESTION: 146
Why is configuring the exact same Claude session to review its newly generated pull request code considered an architectural anti-pattern?
A. It typically exceeds the maximum token limit of the context window during the generation phase.
B. The reviewer retains the reasoning context and assumptions from the generation phase, creating confirmation bias.
C. Tools invoked during the first pass are permanently locked and cannot be reused in the review pass.
D. The Message Batches API does not support synchronous self-review workflows.
Answer: (SHOW ANSWER)
Same-session self-review is an anti-pattern because the model retains the context and reasoning from the generation phase, leading to a blind spot or confirmation bias. Independent review instances with fresh contexts are much more effective at objectively evaluating code.

NEW QUESTION: 147
When implementing a hub-and-spoke multi-agent architecture using the Claude Agent SDK, which of the following statements about subagent spawning and execution are correct? Choose 2 correct answers.
A. The coordinator agent must have Task included in its allowedTools configuration to spawn subagents.
B. Subagents automatically inherit the full conversation history of the coordinator agent by default.
C. A single coordinator response cannot emit multiple Task tool calls; subagents must be spawned sequentially across separate turns.
D. Parallel execution is achieved when the coordinator agent emits multiple Task tool calls in a single response.
Answer: (SHOW ANSWER)
To spawn subagents, the coordinator must be configured with 'Task' in its allowedTools. Furthermore, emitting multiple Task tool calls in a single response allows the subagents to execute their tasks in parallel. Subagents do not automatically inherit conversation history, and forcing sequential spawning is inefficient.

NEW QUESTION: 148
A developer is designing a hub-and-spoke multi-agent system. How should context be passed from the coordinator down to a newly spawned subagent?
A. The subagent automatically inherits the coordinator's full conversation history by default.
B. Provide the coordinator's complete context window inside the Task tool prompt for maximum context visibility.
C. Pass ONLY the explicit context, findings, and parameters specifically relevant to the subagent's assigned subtask.
D. Write all context to a shared scratchpad file that all agents read globally before executing their turns.

Answer: (SHOW ANSWER)
Context isolation is a core principle in multi-agent orchestration. A subagent does not automatically inherit the parent's context and should never be passed the full conversation history, as this causes context pollution. It should only receive explicit, relevant context specific to its task.

NEW QUESTION: 149
To mitigate the 'lost in the middle' effect when passing a large set of aggregated documents to Claude, where should you place the most critical instructions and key findings summaries?
A. Evenly distributed throughout the middle of the document chunks.
B. At the beginning and end of the aggregated input.
C. Inside a hidden HTML comment block <!-- -->.
D. In a separate PostToolUse hook payload.
Answer: (SHOW ANSWER)
Language models reliably process information at the beginning (primacy) and end (recency) of long inputs, but may omit findings from the middle sections. Therefore, position-aware ordering dictates placing the most important information at the beginning and end.

NEW QUESTION: 150
When designing subagents to handle conflicting data from different sources during the final synthesis phase, which of the following are recommended practices? Choose 2 correct answers.
freecram.net
A. Silently average the conflicting numerical values to provide a balanced estimate.
B. Require subagents to output structured claim-source mappings.
C. Preserve information provenance including the source, confidence level, and timestamp.
D. Arbitrarily select the value from the most recently updated database to resolve the conflict.
Answer: B,C (LEAVE A REPLY)
When handling conflicting information, the system must maintain information provenance by requiring structured claim-source mappings (tracking source, confidence, and timestamp). This allows the system or a human reviewer to resolve conflicts based on reliability. Arbitrary selection or silently averaging data are architectural anti-patterns.

NEW QUESTION: 151
Using the FastMCP Python SDK, which decorators are used to expose capabilities to an MCP client? Choose 2 correct answers.
A. @mcp.tool
B. @mcp.agent
C. @mcp.action
D. @mcp.resource
Answer: (SHOW ANSWER)
The FastMCP Python SDK utilizes @mcp.tool to automatically generate JSON schemas for executable functions, and @mcp.resource to expose fetchable data endpoints. @mcp.agent and @mcp.action are not valid decorators in this SDK.

Valid CCA-F Dumps shared by EduDump.com for Helping Passing CCA-F Exam! EduDump.com now offer the **newest CCA-F exam dumps**, the EduDump.com CCA-F exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com CCA-F dumps with Test Engine here:
https://www.edudump.com/exams/Anthropic/CCA-F/premium/ (**112** Q&As Dumps, **35%OFF** **Special Discount Code:** **freecram**)

NEW QUESTION: 152
When implementing an agentic loop using the Claude API, what is the only architecturally reliable signal to determine whether the loop should terminate or continue executing?
A. Parsing the assistant's natural language text output for completion keywords like 'task complete'.
B. Inspecting the stop_reason field of the API response for tool_use versus end_turn.
C. Implementing a hard maximum iteration cap of 5 loops to prevent infinite execution.
D. Monitoring the total token count and exiting when it reaches the context window limit.
Answer: (SHOW ANSWER)
The stop_reason field is the only reliable, deterministic signal for controlling the agentic loop. 'tool_use' indicates the loop must continue to execute tools, while 'end_turn' indicates Claude has finished reasoning and the loop should exit. Parsing natural language or setting arbitrary iteration caps are explicitly called out as anti-patterns.

NEW QUESTION: 153
Which of the following statements represent correct principles of tool distribution and interface design? Choose 2 correct answers.
A. Increasing the total number of tools given to a single coordinator agent inherently improves its decision-making accuracy.
B. Grouping related tools and assigning them to specialized subagents improves tool selection reliability.
C. Setting tool_choice: "auto" forces Claude to always execute at least one tool.
D. Providing detailed input formats, edge cases, and boundaries in tool descriptions prevents the model from misrouting requests between similar tools.
Answer: (SHOW ANSWER)
Proper tool distribution involves keeping tool counts low per agent by delegating to specialized subagents, and using highly detailed tool descriptions to define boundaries and prevent misrouting. Providing too many tools degrades accuracy, and tool_choice: "auto" allows the model to optionally bypass tools entirely.

Valid CCA-F Dumps shared by EduDump.com for Helping Passing CCA-F Exam! EduDump.com now offer the **newest CCA-F exam dumps**, the EduDump.com CCA-F exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com CCA-F dumps with Test Engine here:
https://www.edudump.com/exams/Anthropic/CCA-F/premium/ (**112** Q&As Dumps, **35%OFF** **Special Discount Code:** **freecram**)