COMMUNITY STUDY GUIDE · V1.3 · JULY 2026

Claude Certified Architect: Foundations

A structured learning guide for AI agent architecture, built around the CCA-F body of knowledge. Free to share.

5	18	35 hrs	80+
EXAM DOMAINS	STUDY UNITS	TOTAL CONTENT	CHECKLIST ITEMS

> **The core principle behind everything in this guide:**
> When something *must* happen, enforce it programmatically (hooks, interceptors, code). Never rely on prompt instructions for critical business logic.

About This Guide

This guide was built by Prompt Goblins, an independent community for people who build with AI. We're using the CCA-F exam's five domains as a framework to create a properly scoped, logically ordered study plan that covers everything a production Claude architect needs to know, and then some. The exam itself is currently limited to organizations in the Claude Partner Network — though the network is free for any organization bringing Claude to market to join — and is delivered through Pearson VUE at $125/attempt (raised from the $99 launch price on June 30, 2026; partner-tier discounts apply). The body of knowledge it tests is relevant to anyone building with Claude.

The CCA-F was the first professional certification in AI agent architecture, and it is now one of **four Claude certifications**: Claude Certified Associate – Foundations ($99), Claude Certified Developer – Foundations ($125), Claude Certified Architect – Foundations ($125, this guide), and Claude Certified Architect – Professional ($175). Whether or not you plan to take the exam, the five domains it tests (agentic architecture, tool design, Claude Code configuration, prompt engineering, and context management) represent a comprehensive map of the skills needed to build production Claude systems. That makes it a useful learning scaffold even if the exam itself stays behind Anthropic's partner wall.

How to read the attribution tags

Every resource in this guide is tagged with one of three tiers so you know exactly where it came from:

OFFICIAL	**Named on an Anthropic-curated CCA-F course list.** Anthropic now publishes two: the public Partner Network Learning Path (4 courses) and a partner-gated "CCA-F Prep Courses" page (7 courses). Also includes the Exam Guide PDF. (The official practice exam was retired when exam delivery moved to Pearson VUE.)
ANTHROPIC	**Anthropic-published content, community-mapped to exam domains.** Other Anthropic Academy courses, official docs (platform.claude.com and code.claude.com), the "Building Effective Agents" blog post, and DeepLearning.AI partnership courses taught by Anthropic staff. High-quality and relevant, but not on any Anthropic CCA-F prep page.
SUPPLEMENTAL	**Community and third-party resources.** Study guides, practice exams, and general-purpose courses included to round out coverage.

Tags are deliberately grayscale — in this guide, **color always means an exam domain (D1–D5)**, never a source tier.

Anthropic's exam guide recommends *hands-on building* over reading. It doesn't name specific courses, blog posts, or doc pages. The public Partner Network Learning Path names only four Skilljar courses (Intro to Agent Skills, Building with the Claude API, Intro to MCP, Claude Code in Action); the partner-gated prep page names seven (adding Claude 101, AI Fluency, and the Vertex AI / Bedrock cloud courses, but dropping Agent Skills). The two lists don't fully agree, and everything beyond them (including most of what makes this guide useful) is community-mapped. We've done the mapping carefully, but we want you to know the difference.

Versioning & updates

This is **v1.3 (July 2026)**, a full re-validation of the May 2026 v1.0 against the live web (July 19–20, 2026), reconciled against the official exam guides, with navigation and readability improvements — see the Change Log at the end. It is based on Anthropic's official **exam guide v1.0 (effective July 2026)**, confirmed against the primary-source PDF — same five domains and weights and the same 60-question, 4-of-6-scenario format as the earlier v0.1 draft, at the current $125 price. Things will keep changing. You can always drop this PDF into Claude and ask *"is this still up to date?"* to check for changes. We'll version updates on promptgoblins.ai as things evolve.

Exam Domains at a Glance

D1 · Agentic Architecture & Orchestration — **27%**

Agentic loops, hub-and-spoke multi-agent, Agent SDK, lifecycle hooks, task decomposition

D3 · Claude Code Configuration & Workflows — **20%**

CLAUDE.md hierarchy, Skills, slash commands, `-p` flag, subagents, plan mode

D4 · Prompt Engineering & Structured Output — **20%**

Few-shot, JSON schemas via `tool_use`, validation-retry loops, batch API, multi-pass review

D2 · Tool Design & MCP Integration — **18%**

MCP primitives, tool descriptions, tool boundaries, `.mcp.json`, structured errors

D5 · Context Management & Reliability — **15%**

Escalation patterns, progressive summarization risks, error propagation, provenance tracking

Study Plan Overview

Work through top to bottom. All Skilljar courses are free with certificates. **Total: ~35 hours / 2–4 weeks.**

OFFICIAL = on Anthropic's Partner Network Learning Path. **ANTHROPIC** = Anthropic content, community-mapped to exam. **SUPPLEMENTAL** = third-party / community.

Jump to: Unit details · Anti-Patterns · Cliffs Notes · Exam Logistics · Exam Guide Appendix · Change Log — resource names below jump to their detail card; platform names link straight out.

#	PHASE	RESOURCE	PLATFORM	FORMAT	TIME	DOMAINS	✓
1	Foundations	Claude 101 O	Skilljar	Video+quiz	1h	—	☐
2	Foundations	AI Fluency O: Framework & Foundations	Skilljar	Video+quiz	1h	D4 D5	☐
3	API & Agents	Building with the Claude API	Skilljar	84 lec, 10 quiz	8h	D1 D2 D4 D5	☐
4	API & Agents	"Building Effective Agents"	anthropic.com	Reading	1h	D1	☐
5	API & Agents	Building Toward Computer Use	DLAI	9 vid+code	2h	D4 D2	☐
6	MCP	Intro to Model Context Protocol	Skilljar	Video+code	2h	D2	☐
7	MCP	MCP: Build Rich-Context AI Apps	DLAI	11 vid+code	2h	D2	☐
8	MCP	MCP: Advanced Topics	Skilljar	Video+code	2h	D2 D5	☐
9	Claude Code	Claude Code in Action	Skilljar	15 lec+quiz	1h	D3 D2 D5	☐
10	Claude Code	Claude Code: Agentic Coding	DLAI	10 vid	2h	D3 D1 D5	☐
11	Skills & Agents	Intro to Agent Skills	Skilljar	6 vid	30m	D3 D1	☐
12	Skills & Agents	Agent Skills with Anthropic	DLAI	10 vid	2.5h	D1 D3	☐
13	Skills & Agents	Intro to Subagents	Skilljar	Video+quiz	1h	D1 D3	☐
14	Skills & Agents	Intro to Claude Cowork	Skilljar	Video+code	1h	D3	☐
15	Reading	Documentation deep reads	platform/code.claude.com	Reading	3h	All	☐
16	Reading	Community study guides	GitHub / blogs	Reading	2h	All	☐
17	Exam Prep	Official Exam Guide PDF	GitHub PDF	PDF	1h	All	☐
18	Exam Prep	Practice Exams (target 900+)	Community	60+ MCQs	2h	All	☐

Detailed Study Plan

PHASE: FOUNDATIONS

1 Claude 101 OFFICIAL

Link: anthropic.skilljar.com/claude-101 · **Time:** ~1h · **Format:** Video+quiz · **Cert:** Yes

Core Claude features, conversation patterns, Projects. On Anthropic's partner-gated CCA-F prep-course list. *Experienced users: skim at 2x, grab cert.*

- [ ] Context window mechanics and token limits
- [ ] System prompts vs. user prompts vs. assistant turns
- [ ] Projects: persistent instructions and knowledge bases
- [ ] When to use Claude.ai vs. API vs. Claude Code

2 AI Fluency: Framework & Foundations OFFICIAL D4 D5

Link: anthropic.skilljar.com/ai-fluency-framework-foundations · **Time:** ~1h · **Cert:** Yes

The 4D Framework. D4 questions assume you understand *why* strategies work. Discernment → D5 confidence calibration.

- [ ] 4D Framework: Delegation, Description, Discernment, Diligence
- [ ] Delegation: matching task complexity to AI capability
- [ ] Description: specificity, structure, and constraint design
- [ ] Discernment: evaluating and validating AI outputs
- [ ] Diligence: responsible use, bias awareness, human oversight

PHASE: CORE / API & AGENTS

3 Building with the Claude API OFFICIAL D1 D2 D4 D5

Link: anthropic.skilljar.com/claude-with-the-anthropic-api · **Time:** ~8h · **Format:** 84 lectures, 10 quizzes · **Cert:** Yes
Also on: Coursera (single course, instructor Stephen Grider)

> **This is the single most important course.** Covers the most exam surface area. Budget a full week at 1–2h/day.

API fundamentals	Auth, SDKs, message structure, models, parameters
Prompt engineering	XML tags, few-shot, chain-of-thought, evaluation workflows
Structured output	`tool_use` with JSON schemas, validation-retry loops
Tool use	Custom tools, `tool_choice`, `stop_reason`, batch ops
RAG	Chunking, embeddings, BM25, contextual retrieval
Agents	Agentic loops, parallelization, chaining, routing, orchestrator-workers

- [ ] Agentic loop: send request → check `stop_reason` → if `tool_use` continue, if `end_turn` stop
- [ ] `stop_reason` values and what each means for loop control
- [ ] `tool_choice`: `auto`, `any`, `tool`: when to use each
- [ ] Tool definition: `name`, `description`, `input_schema` with JSON Schema
- [ ] Tool descriptions are the primary mechanism Claude uses to select tools
- [ ] Few-shot prompting: 2–4 examples is the sweet spot
- [ ] Validation-retry: generate → validate → if invalid, feed error → regenerate
- [ ] Message Batches API: 50% savings, 24h window, latency-tolerant only
- [ ] Prompt caching: cache static content to reduce cost/latency
- [ ] Extended thinking: `thinking` blocks, `budget_tokens` parameter
- [ ] Agent patterns: chaining, routing, parallelization, orchestrator-workers, evaluator-optimizer
- [ ] Multi-pass review: independent instances (no prior reasoning context)

4 "Building Effective Agents" (Blog Post) ANTHROPIC D1

Link: anthropic.com/engineering/building-effective-agents · **Time:** ~1h · **Format:** Reading

> **Conceptual backbone of Domain 1.** Multiple prep guides call this the document behind ~65% of the exam's mental model.

- [ ] Workflows vs. agents: workflows = code controls flow; agents = LLM controls flow
- [ ] Prompt chaining: sequential steps, each output feeds the next
- [ ] Routing: classify input → specialized handler
- [ ] Parallelization: simultaneous LLM calls, aggregate results
- [ ] Orchestrator-workers: central LLM decomposes, delegates
- [ ] Evaluator-optimizer: generate → evaluate → loop until quality threshold
- [ ] "Start simple": single call → workflow → agent (escalate only when needed)
- [ ] Failure modes: unbounded loops, context degradation, tool misselection

5 Building Toward Computer Use with Anthropic ANTHROPIC D4 D2

Link: deeplearning.ai/courses/building-toward-computer-use-with-anthropic · **Time:** ~2h (1h 47m)
Instructor: Colt Steele (Anthropic) · **Also on:** Coursera

Prompt engineering, XML structuring, n-shot, caching, tool use schemas. *Computer Use is out of scope for CCA-F.*

- [ ] Multimodal message construction: text + images
- [ ] XML tag structuring for complex prompts
- [ ] N-shot prompting patterns in practice
- [ ] Chain-of-thought elicitation techniques
- [ ] Prompt caching implementation and cost implications
- [ ] Tool use JSON schema enforcement patterns

PHASE: CORE / MCP

6 Introduction to Model Context Protocol OFFICIAL D2

Link: anthropic.skilljar.com/introduction-to-model-context-protocol · **Time:** ~2h · **Cert:** Yes
Also on: Coursera

- [ ] **The three MCP primitives** (heavily tested):
  - [ ] Tools = model-controlled (LLM decides when to call)
  - [ ] Resources = app-controlled (application decides when to load)
  - [ ] Prompts = user-controlled (user selects from templates)
- [ ] MCP server architecture: tool definition shifts to specialized servers
- [ ] Python SDK decorators for server construction
- [ ] Resource URIs, MIME types, and template resources
- [ ] MCP Inspector for debugging and testing

7 MCP: Build Rich-Context AI Apps ANTHROPIC D2

Link: deeplearning.ai/courses/mcp-build-rich-context-ai-apps-with-anthropic · **Time:** ~2h
Instructor: Elie Schoppik (Anthropic)

Hands-on build. Theory from Unit 6 + this = comprehensive D2 coverage.

- [ ] FastMCP for rapid server prototyping
- [ ] Converting existing tool-using apps to MCP architecture
- [ ] Reference servers: filesystem, fetch
- [ ] Claude Desktop MCP configuration
- [ ] Local vs. remote MCP server deployment

8 Model Context Protocol: Advanced Topics ANTHROPIC D2 D5

Link: anthropic.skilljar.com/model-context-protocol-advanced-topics · **Time:** ~2h · **Cert:** Yes
Also on: Coursera

- [ ] **Sampling:** servers request LLM calls through clients (shifts AI costs)
- [ ] Progress and logging notifications via context objects
- [ ] **Roots-based file access:** permission boundaries, security
- [ ] JSON-RPC 2.0 message architecture
- [ ] **Transport selection** (directly tested): stdio = local; StreamableHTTP = remote/production
- [ ] Stateless vs. stateful server scaling

PHASE: CORE / CLAUDE CODE

9 Claude Code in Action OFFICIAL D3 D2 D5

Link: anthropic.skilljar.com/claude-code-in-action · **Time:** ~1h · **Cert:** Yes · **Also on:** Coursera

- [ ] Claude Code's internal multi-tool architecture
- [ ] Context management: `/clear`, `/compact`, file mentions, escape
- [ ] **Plan Mode vs. direct execution**: when each is appropriate
- [ ] Custom slash commands: creation and use
- [ ] MCP server integration within Claude Code
- [ ] GitHub workflow: automated code review, PR creation

10 Claude Code: A Highly Agentic Coding Assistant ANTHROPIC D3 D1 D5

Link: deeplearning.ai/courses/claude-code-a-highly-agentic-coding-assistant · **Time:** ~2h · **Instructor:** Elie Schoppik

- [ ] **CLAUDE.md hierarchy** (heavily tested): Enterprise → Project → User (overrides cascade down)
- [ ] `.claude/rules/` with glob patterns for path-specific rules
- [ ] `-p` **flag** for CI/CD non-interactive mode. Without it, CI jobs hang
- [ ] Git worktrees for parallel Claude Code sessions
- [ ] **Hooks:** PreToolUse, PostToolUse: programmatic enforcement points
- [ ] Subagent brainstorming: isolated context for exploration
- [ ] Testing and debugging workflows

PHASE: CORE / SKILLS & AGENTS

11 Introduction to Agent Skills OFFICIAL D3 D1

Link: anthropic.skilljar.com/introduction-to-agent-skills · **Time:** ~30min · **Cert:** Yes

> The distinction between Skills, commands, CLAUDE.md, and hooks is **frequently tested in D3**.

- [ ] **Skills vs. other customization:** Skills=reusable SKILL.md folders; Commands=slash-triggered; CLAUDE.md=persistent context; Hooks=programmatic pre/post
- [ ] SKILL.md frontmatter: description field drives trigger matching
- [ ] Progressive disclosure for context window efficiency
- [ ] `allowed-tools` for restricting tool access within a Skill
- [ ] Sharing Skills: repository commits, plugins

12 Agent Skills with Anthropic ANTHROPIC D1 D3

Link: deeplearning.ai/courses/agent-skills-with-anthropic · **Time:** ~2.5h (2h 19m) · **Instructor:** Elie Schoppik

> **The only course covering the Claude Agent SDK** with skills + MCP + external integrations.

- [ ] Skills vs. Tools vs. MCP vs. Subagents: clear comparison
- [ ] Skills + Claude API: code execution, files API, filesystem access
- [ ] Skills + Claude Code: code gen, review, testing workflows
- [ ] Subagents with skills for isolated context delegation
- [ ] **Agent SDK:** building a research agent with skills + MCP + web search

13 Introduction to Subagents ANTHROPIC D1 D3

Link: anthropic.skilljar.com/introduction-to-subagents · **Time:** ~1h · **Cert:** Yes

- [ ] Subagent mechanics: separate context window, input in, summary returns
- [ ] Creating custom subagents via `/agents` command
- [ ] Limited tool access per subagent (isolation principle)
- [ ] When to use vs. avoid subagents
- [ ] **Anti-pattern:** sharing full coordinator context with all subagents

14 Introduction to Claude Cowork ANTHROPIC D3

Link: anthropic.skilljar.com/introduction-to-claude-cowork · **Time:** ~1h · **Cert:** Yes

- [ ] Cowork task loop vs. Claude Code's direct interaction
- [ ] Plugin ecosystem and skills within Cowork
- [ ] File and research workflows
- [ ] Responsible steering of multi-step autonomous work

PHASE: READING & REVIEW

15 Documentation Deep Reads

ANTHROPIC D1 D2 D3 D4 D5

Time: ~3h · **Format:** Reading

Read in this order. Each builds on the previous. *Note: Anthropic's docs moved off docs.anthropic.com in 2025–26 — API docs now live at platform.claude.com/docs, Claude Code and Agent SDK docs at code.claude.com/docs. Old links redirect.*

TOPIC	URL	DOMAIN
Tool Use guide	platform.claude.com/docs/en/agents-and-tools/tool-use/overview	D1 D2
Prompt Engineering	platform.claude.com/docs/.../prompt-engineering/overview	D4
Prompting Best Practices	platform.claude.com/docs/.../claude-prompting-best-practices	D4
Structured Outputs	platform.claude.com/docs/.../structured-outputs	D4
Extended Thinking	platform.claude.com/docs/.../extended-thinking	D4 D5
Context Windows	platform.claude.com/docs/.../context-windows	D5
Agent SDK Overview	code.claude.com/docs/en/agent-sdk/overview	D1
Claude Code Skills	code.claude.com/docs/en/skills	D3
Claude Code Workflows	code.claude.com/docs/en/common-workflows	D3
Claude Code CLI	code.claude.com/docs/en/cli-usage	D3
MCP Specification	modelcontextprotocol.io/specification	D2

Hands-on repos:

Anthropic Cookbooks	github.com/anthropics/claude-cookbooks
Prompt Eng. Tutorial	github.com/anthropics/prompt-eng-interactive-tutorial

16 Community Study Guides

SUPPLEMENTAL

Time: ~2h · **Format:** Reading

paullarionov study guide	github.com/paullarionov/claude-certified-architect (~3,900★, actively maintained)
claudecertificationguide.com	claudecertificationguide.com — 30 lessons, 240+ questions, mock exam; tracks exam guide v0.2
Rick Hightower 8-part series	pub.towardsai.net/...cca-foundations-exam
Tutorials Dojo	tutorialsdojo.com/ccar-f-study-guide
timothywarner-org repo	github.com/timothywarner-org/claude-architect — study materials + code examples
CCA-F study plugin	github.com/carolinacherry/claude-certified-architect — Claude Code plugin covering all 5 domains

PHASE: EXAM PREP

17 Official Exam Guide PDF OFFICIAL

Time: ~1h · **Format:** PDF

Public PDF (v0.1 community mirror): github.com/paullarionov/.../guide_en.pdf · The current **v1.0** is downloadable from the partner-academy certification page (partner login).

Contains: 5 domain breakdowns, 6 scenarios, sample Q&As, and the out-of-scope list. The current official guide is **v1.0 (effective July 2026)**; it keeps the same domains, weights, and 4-of-6-scenario format as the earlier v0.1 draft.

> **Study tip:** Wrong-answer explanations are as important as correct answers. They reveal the anti-patterns.

18 Practice Exams SUPPLEMENTAL

Time: ~2h · **Format:** 60+ MCQs with explanations

> **The official Anthropic practice exam was retired** in the move to Pearson VUE (per Anthropic's certification FAQ). Everything below is unofficial community material — quality varies, so treat these as drills, not as predictors of the real exam.

> **Target: 900+/1000 before scheduling.** Below 800 = more prep needed.

Rick Hightower 60-question	medium.com/@richardhightower/...practice-exam
OlivierAlter 77 scenarios	github.com/OlivierAlter/...Certification-Exam
CertSafari (600+ questions)	certsafari.com/.../claude-certified-architect
claudecertificationguide.com mock exam	claudecertificationguide.com

Ten Anti-Patterns the Exam Tests

Roughly half the exam tests whether you can spot what's wrong. Memorize these.

#	ANTI-PATTERN	DO THIS INSTEAD
1	Parsing natural language for loop termination	Use `stop_reason == "end_turn"`
2	Arbitrary iteration caps (`max_iterations=5`)	`stop_reason` -driven termination
3	Prompt-based critical rules ("don't process refunds >$500")	PostToolUse hooks / programmatic checks
4	Sharing full coordinator context with subagents	Pass only relevant context; isolate scope
5	Sentiment-based escalation	Explicit request, policy gap, capability limit
6	Summarizing transactional facts (losing order #s)	Immutable "case facts" blocks
7	Self-review in same session	Independent review instance, no prior context
8	Tool overload (>4–5 per agent)	Scope per role; decompose into subagents
9	Few-shot as first fix for misrouting	Fix the tool description first
10	Batch API for blocking workflows	Batch = 24h window, latency-tolerant only

Concept Cliffs Notes

D1 · Agentic Architecture & Orchestration (27%)

The agentic loop is the core pattern: send request → check `stop_reason` → if `tool_use`, execute tool, return result, loop → if `end_turn`, stop. This is the *only* reliable termination mechanism.

Hub-and-spoke (coordinator-subagent) is the primary multi-agent pattern. Coordinator decomposes tasks, delegates to subagents with isolated context. Each subagent gets only what it needs. Never dump full coordinator context into subagents.

Lifecycle hooks (PreToolUse, PostToolUse) enforce business rules programmatically. When the exam says "must" or "always," the answer is hooks, not prompts.

Agent SDK provides agent definitions, tool registration, session management, orchestration. Hooks register at SDK level.

D2 · Tool Design & MCP Integration (18%)

Tool descriptions are the primary selection mechanism. Poor descriptions → misrouting. Fix descriptions *before* adding few-shot examples.

MCP primitives: Tools = model-controlled. Resources = app-controlled. Prompts = user-controlled. The exam tests which primitive for which scenario.

Tool boundaries: 4–5 tools per agent max. Beyond that, selection degrades. Decompose into subagents.

Structured errors: transient (retry), validation (fix input), business (rule violation), permission (access denied).

Transport: stdio = local/same machine. Streamable HTTP = remote/production. The older HTTP+SSE transport was deprecated in the 2025-03-26 spec revision and is being sunset. `.mcp.json` = project level; `~/.claude.json` = user level.

D3 · Claude Code Configuration & Workflows (20%)

CLAUDE.md hierarchy: Enterprise → Project → User. Overrides cascade downward.

`.claude/rules/`: rule files with glob patterns for path-specific instructions (e.g. `*.test.ts` rules only in test files).

Skills = folders with SKILL.md. The `description` field triggers automatic loading. Progressive disclosure for context efficiency.

`-p` **flag** = non-interactive mode for CI/CD. Without it, jobs hang. **Plan Mode** = plan before executing (complex changes). Direct = skip planning (quick edits).

D4 · Prompt Engineering & Structured Output (20%)

Few-shot: 2–4 examples optimal. Demonstrate exact output format including edge cases.

Structured output via `tool_use`: Define JSON schema as tool input. More reliable than asking for JSON in prompt. *Real-world note: the API now also has a native structured-outputs feature (`output_config.format` with constrained decoding, plus `strict: true` tool schemas). The exam guide teaches the `tool_use` pattern — know both.*

Validation-retry: Generate → validate → if invalid, feed error back → regenerate. Converges in 1–2 retries.

Multi-pass review: Independent instance, no prior context. Explicit criteria with thresholds.

Batch API: 50% savings, up to 24h processing. Latency-tolerant only.

D5 · Context Management & Reliability (15%)

Progressive summarization is dangerous for transactional data. Maintain immutable "case facts" blocks.

Valid escalation: (1) policy gap, (2) explicit customer request, (3) capability limit. **Invalid:** sentiment analysis, self-assessed confidence.

Error propagation: Errors flow upward with full context: what, which agent, what failed.

Scratchpad files: Persist info across long sessions without relying on context window memory.

Exam Logistics

Questions	60 MCQ (1 correct, 3 distractors)
Duration	120 minutes (~135 min total with check-in procedures)
Delivery	Pearson VUE since mid-2026: OnVUE online proctoring or test centers
Proctoring	Closed-book, no AI tools, no external docs
Scoring	100–1,000 scale · **Pass at 720**
Guessing penalty	None. Answer every question
Scenarios	4 drawn at random from a bank of 6 (official exam guide v1.0)
Score report	Score on screen at exam end; Credly badge email usually within minutes (online) — with domain-level breakdowns
Cost	$125/attempt (raised from $99 on June 30, 2026) · 50% off for Select/Preferred/Global Premier partner tiers; free for Global Premier through Aug 31, 2026
Retake	Up to 4 attempts per rolling 12 months; waits of 14/30/90 days after attempts 1/2/3; full fee each
Validity	12 months from passing date (extended from 6 months on June 30, 2026); renewable via free non-proctored assessment
Badge	Digital credential via Credly, shareable on LinkedIn
Eligibility	Claude Partner Network organizations (network is free to join)
Other exams	Associate–F ($99) · Developer–F ($125) · Architect–Professional ($175)
Out of scope	Fine-tuning, auth, vision, streaming, computer use
Registration	anthropic-partners.skilljar.com/.../foundations-certification (partner login required)
Exam Guide PDF	github.com/paullarionov/.../guide_en.pdf (v0.1 community mirror; the current v1.0 is downloadable from the partner-academy cert page)

Appendix: CCA-F Exam Guide Reference

Domain and scenario structure below matches Anthropic's official Claude Certified Architect: Foundations Exam Guide **v1.0 (effective July 2026)**, verified against the primary-source PDF. An earlier community mirror (exam guide v0.1, Feb 10 2025) is at github.com/paullarionov/claude-certified-architect, with expanded notes at guide_en.md. Domains, weights, and the 4-of-6-scenario format are unchanged between v0.1 and v1.0.

TARGET CANDIDATE

The ideal candidate is a **solution architect** who designs and ships production applications with Claude, with at least 6 months of hands-on experience across: Claude Agent SDK (multi-agent orchestration, subagents, tool integration, lifecycle hooks), Claude Code (CLAUDE.md, MCP servers, Agent Skills, planning mode), MCP (tools and resources for backend integration), prompt engineering (JSON schemas, few-shot examples, data extraction), context windows (long documents, multi-agent context passing), CI/CD (automated code review, test generation), and escalation/reliability (error handling, human-in-the-loop).

PRODUCTION SCENARIOS

The exam draws **4 scenarios at random from a bank of 6** per sitting (official exam guide v1.0); all questions are anchored within them. The 6 official scenarios are below. (Some community study guides list extra candidate-reported scenarios such as "Conversational AI Architecture Patterns" and "Agentic AI Tools"; these are not in the official guide.)

Scenario 1: Customer Support Agent

Build an agent using Claude Agent SDK to handle returns, billing disputes, and account issues. Uses MCP tools: `get_customer`, `lookup_order`, `process_refund`, `escalate_to_human`. Target: 80%+ first-contact resolution with appropriate escalation.

Scenario 2: Code Generation with Claude Code

Use Claude Code to accelerate development: code generation, refactoring, debugging, documentation. Integrate with custom slash commands, CLAUDE.md configuration, and understand when to use planning mode vs. direct execution.

Scenario 3: Multi-Agent Research System

A coordinator agent delegates to specialized subagents: web research, document analysis, synthesis, and report generation. The system must produce complete reports with citations and handle conflicting sources.

Scenario 4: Developer Productivity Tools

The agent helps engineers explore unfamiliar codebases, generate boilerplate, and automate routine tasks. Uses built-in tools (Read, Write, Bash, Grep, Glob) combined with MCP servers.

Scenario 5: Claude Code for CI/CD

Integrate Claude Code into CI/CD pipelines for automated code reviews, test generation, and PR feedback. Uses `-p` flag for non-interactive mode and `--output-format json`. Prompts must minimize false positives.

Scenario 6: Structured Data Extraction

Extract information from unstructured documents, validate output with JSON schemas via `tool_use`, and maintain high accuracy. Must handle edge cases, nullable fields, and inconsistent formats using validation-retry loops.

DOMAIN 1: AGENTIC ARCHITECTURE & ORCHESTRATION (27%)

7 task statements. The heaviest domain.

TASK	WHAT IT COVERS
1.1	Agentic loop lifecycle: `stop_reason` handling (`tool_use` → continue, `end_turn` → stop), tool result appending, model-driven vs. hard-coded decision trees
1.2	Hub-and-spoke architecture: coordinator owns all inter-agent communication, error handling, routing. Subagents operate with isolated context.
1.3	Subagent invocation via `Task` tool with explicit context passing. Subagents do NOT inherit parent context. `AgentDefinition` configuration.
1.4	Multi-step workflows with enforcement and handoff. When to use hooks vs. prompt instructions.
1.5	Agent SDK hooks: `PreToolUse` (block/modify before execution), `PostToolUse` (validate/transform results). Deterministic enforcement of business rules.
1.6	Task decomposition strategies: fixed pipelines vs. dynamic adaptive decomposition. Risk of overly narrow decomposition by coordinator.
1.7	Session management: `fork_session` for exploring alternatives, `--resume` for continuing investigations, when to start fresh vs. resume.

DOMAIN 2: TOOL DESIGN & MCP INTEGRATION (18%)

5 task statements.

TASK	WHAT IT COVERS
2.1	Tool interface design: descriptions as primary selection mechanism. Must include what it does/returns, input formats, edge cases, when to use vs. alternatives.
2.2	Structured error responses with `isError` flag. Four categories: transient (retry), validation (fix input), business (rule violation), permission (escalate).
2.3	Tool distribution across agents (4-5 max per agent). `tool_choice` configuration: `auto` , `any` , `tool` (forced).
2.4	MCP server config: `.mcp.json` (project-level, in VCS) vs. `~/.claude.json` (user-level). Environment variable expansion for secrets.
2.5	Built-in tools: Read, Write, Edit, Bash, Grep, Glob. When to prefer built-in vs. MCP tools.

DOMAIN 3: CLAUDE CODE CONFIGURATION & WORKFLOWS (20%)

6 task statements.

TASK	WHAT IT COVERS
3.1	CLAUDE.md hierarchy: user-level (`~/.claude/CLAUDE.md`) → project-level (`.claude/CLAUDE.md`) → directory-level. `@path` syntax for imports (max depth 5).
3.2	Custom slash commands (`.claude/commands/`) and Skills (`.claude/skills/`) with SKILL.md frontmatter: `context: fork` , `allowed-tools` , `argument-hint` .
3.3	Path-specific rules: `.claude/rules/` with YAML frontmatter and glob patterns. Loaded only when editing matching files.
3.4	Plan mode (investigate + plan, no changes) vs. direct execution. When to use each. Combined approach: plan → approve → execute.
3.5	Iterative refinement: few-shot, test-driven, interview pattern (clarifying questions before implementing).
3.6	CI/CD: `-p` flag (non-interactive, print mode), `--output-format json` , `--json-schema` . Session context isolation for review.

DOMAIN 4: PROMPT ENGINEERING & STRUCTURED OUTPUT (20%)

6 task statements.

TASK	WHAT IT COVERS
4.1	Explicit criteria vs. vague instructions. Define severity with examples. Specify what to flag AND what NOT to flag.
4.2	Few-shot prompting: 2-4 examples covering ambiguous scenarios, output formats, acceptable vs. problematic patterns, diverse document formats.
4.3	Structured output via `tool_use` with JSON schemas. Required vs. optional fields. Nullable types `["string", "null"]`. Enums with `"other"` and `"unclear"`.
4.4	Validation-retry loops with Pydantic. Structural validation → semantic validation → retry with error context. When retry helps vs. doesn't.
4.5	Batch API: 50% savings, 24h window, `custom_id` for correlation, no multi-turn tool calling. SLA planning.
4.6	Multi-instance review: independent Claude instance without prior reasoning context. Explicit evaluation criteria. Preventing duplicate comments on re-review.

DOMAIN 5: CONTEXT MANAGEMENT & RELIABILITY (15%)

6 task statements.

TASK	WHAT IT COVERS
5.1	Context management: lost-in-the-middle effect, progressive summarization traps (losing numeric values, dates), immutable "case facts" blocks.
5.2	Escalation: valid triggers (explicit request, policy gap, capability limit) vs. invalid (sentiment, self-confidence). Structured handoff protocols.
5.3	Error propagation: structured subagent errors with `failure_type`, `partial_results`, `alternative_approaches`, `coverage_impact`. Coverage annotations.
5.4	Large codebase exploration: scratchpad files, `/compact` for context compression, Explore subagent for isolating verbose output.
5.5	Confidence calibration: field-level scores, stratified random sampling, routing by confidence threshold. Overall accuracy can hide per-category errors.
5.6	Information provenance: claim → source linking, handling conflicting data with attribution, including dates for temporal disambiguation.

KEY API CONCEPTS REFERENCE

stop_reason values

VALUE	MEANING	ACTION
`end_turn`	Model finished its response	Show result to user. Loop complete
`tool_use`	Model wants to call a tool	Execute tool, return result, continue loop
`max_tokens`	Token limit reached	Response truncated. May need to increase limit
`stop_sequence`	Stop sequence encountered	Handle per application logic
`pause_turn`	Long-running turn (server tools) paused	Resend the response as-is to continue. *Newer value — post-dates exam guide v0.1*
`refusal`	Model declined to respond	Handle gracefully; do not blind-retry. *Newer value — post-dates exam guide v0.1*

tool_choice values

VALUE	BEHAVIOR	WHEN TO USE
`{"type": "auto"}`	Model decides: tool or text	Default for most cases
`{"type": "any"}`	Model must call some tool	When you need guaranteed structured output
`{"type": "tool", "name": "..."}`	Model must call a specific tool	Forced first step / execution ordering
`{"type": "none"}`	Model may not call any tool	Suppress tool use while keeping tools defined. (Note: `any` / `tool` are incompatible with extended thinking)

Hooks vs. Prompt Instructions

ATTRIBUTE	HOOKS	PROMPT INSTRUCTIONS
Guarantee	Deterministic (100%)	Probabilistic (>90%, not 100%)
When to use	Critical business rules, financial ops, compliance	General preferences, recommendations, formatting
Example	Block refunds > $500	"Try to solve before escalating"

CLAUDE.md Hierarchy

LEVEL	LOCATION	SCOPE	IN VCS?
User	`~/.claude/CLAUDE.md`	Personal preferences	No
Project	`.claude/CLAUDE.md` or root `CLAUDE.md`	All project contributors	Yes
Directory	`CLAUDE.md` in subdirectories	Files in that directory	Yes

MCP Primitives

PRIMITIVE	CONTROL	DESCRIPTION
Tools	Model-controlled	Functions the agent calls to perform actions (CRUD, API calls, commands)
Resources	App-controlled	Data loaded for context (docs, schemas, catalogs)

PRIMITIVE	CONTROL	DESCRIPTION
Prompts	User-controlled	Predefined templates for common tasks

Error Categories in Multi-Agent Systems

CATEGORY	EXAMPLES	RETRYABLE?	AGENT ACTION
Transient	Timeout, 503, network failure	Yes	Retry with exponential backoff
Validation	Invalid input, missing field	No (fix input)	Modify request, retry
Business	Policy violation, threshold exceeded	No	Explain; propose alternative
Permission	Access denied	No	Escalate

Escalation Decision Matrix

SITUATION	ACTION
Customer explicitly asks "get me a manager"	Escalate immediately. Do not attempt to solve
Policy does not cover the request	Escalate (e.g., competitor price matching)
Agent cannot make progress after attempts	Escalate after reasonable attempts
Financial operation above threshold	Escalate (enforce via hook, not prompt)
Multiple customer matches found	Ask for additional identifiers. Do not guess

UNRELIABLE TRIGGER	WHY IT FAILS
Sentiment analysis	Mood ≠ case complexity; culturally biased
Model self-rated confidence (1-10)	Model can be confidently wrong; poor calibration
Automatic classifier	Overengineering; may need training data you don't have

REQUIRED & OUT-OF-SCOPE TECHNOLOGIES

Required Technologies

Claude Agent SDK · Model Context Protocol (MCP) · Claude Code (CLAUDE.md, `.claude/rules/`, `.claude/commands/`, `.claude/skills/`, `-p` flag, plan mode) · Claude API (Messages API, `tool_use`, `tool_choice`, `stop_reason`) · Message Batches API · JSON Schema · Pydantic · Built-in tools (Read, Write, Edit, Bash, Grep, Glob)

Out of Scope: Will Not Be Tested

Fine-tuning · API authentication/billing · Server deployment · Claude's internal architecture · Constitutional AI / RLHF · Embeddings / vector databases · Computer Use · Vision · Streaming implementation · Rate limiting · OAuth · Cloud provider configurations (AWS, GCP) · Benchmarking · Prompt caching implementation details · Tokenization specifics

OFFICIAL DOCUMENTATION LINKS

RESOURCE	URL
Claude API: Messages	platform.claude.com/docs/en/api/messages
Claude API: Tool Use	platform.claude.com/docs/.../tool-use/overview
Claude API: Batch Processing	platform.claude.com/docs/.../batch-processing
Claude Agent SDK: Overview	code.claude.com/docs/en/agent-sdk/overview
Claude Code: CLAUDE.md / Memory	code.claude.com/docs/en/memory
Claude Code: Skills	code.claude.com/docs/en/skills
Claude Code: Hooks	code.claude.com/docs/en/hooks
Claude Code: Sub-agents	code.claude.com/docs/en/sub-agents
Claude Code: CLI / Headless	code.claude.com/docs/en/cli-usage
MCP Specification	modelcontextprotocol.io/specification
MCP: Tools	modelcontextprotocol.io/specification/.../server/tools
MCP: Resources	modelcontextprotocol.io/specification/.../server/resources
Prompt Engineering Guide	platform.claude.com/docs/.../prompt-engineering/overview
Claude Cookbooks	github.com/anthropics/claude-cookbooks

Change Log

V1.3 — JULY 28, 2026

Navigation and readability release — no factual changes.

- [ ] **Tier tags restyled to grayscale** (solid = OFFICIAL, outlined = ANTHROPIC, gray = SUPPLEMENTAL). The old blue/green/gold tag colors collided with the domain colors (D2/D4/D5), making page 2 vs. page 3 read as one color system when they were two. Color now always means exam domain.
- [ ] **Study plan is now a working table of contents:** every resource name in the overview links to its detail card (units 1–18); platform names still link straight out to the course. Added a "Jump to" nav line and section anchors. Links work in both the HTML and the PDF.
- [ ] **Screen-friendly HTML:** added screen-only styles so the same file reads comfortably in a browser (for web/SEO publishing) while the print/PDF output is unchanged.

V1.2 — JULY 20, 2026

Reconciled against the primary-source official exam guides (all v1.0, "effective July 2026") and companion guides published.

- [ ] **Exam guide version confirmed:** the current official CCA-F guide is **v1.0 (July 2026)** — verified against Anthropic's own PDF. Replaced the earlier unverified "v0.2 (June 30)" hedge throughout.
- [ ] **Scenarios corrected back to 6:** the official v1.0 guide specifies a bank of 6 scenarios (4 drawn per sitting). The "now 8 scenarios" note (from a community mirror's candidate-reported additions) was over-stated and has been rolled back; the extra scenarios are now flagged as unofficial.
- [ ] **Companion guides:** this program now has four certifications. Published a combined **Builder Track** guide (Developer–Foundations + Architect–Foundations + Architect–Professional, core-first) and a standalone **Associate–Foundations** guide, both built from the official v1.0 blueprints. This CCA-F guide's material is also carried as the Architect–Foundations path inside the Builder Track guide.

V1.1 — JULY 19, 2026

Full re-validation of every link and factual claim against the live web (Anthropic Academy/Skilljar, the partner-academy certification FAQ, Pearson VUE, platform.claude.com and code.claude.com docs, DeepLearning.AI, Coursera, GitHub, and community sites). Changes below are grouped by type.

Exam & certification program (corrections)

- [ ] **Price:** $99 → **$125**/attempt, effective June 30, 2026. Partner-tier discounts added (50% for Select/Preferred/Global Premier; free for Global Premier through Aug 31, 2026). Source: Anthropic certification FAQ.
- [ ] **Removed "free for first 5,000 partner employees":** could not be corroborated on any official Anthropic page; the FAQ documents the tier-discount system instead.
- [ ] **Validity:** 2 years → **12 months** (itself extended from an original 6 months on June 30, 2026), renewable via a free non-proctored assessment.
- [ ] **Retakes:** now specified — up to 4 attempts per rolling 12 months, waiting periods of 14/30/90 days after attempts 1/2/3, full fee each.
- [ ] **Score report:** "within 2 business days" corrected — score appears on screen at exam end; Credly badge email usually arrives within minutes for online-proctored exams. Domain-level breakdown confirmed.
- [ ] **Delivery:** added — exams moved to Pearson VUE (OnVUE online proctoring or test centers) around June 30–July 13, 2026.
- [ ] **Official practice exam:** retired in the Pearson move (per FAQ); Unit 18 retagged SUPPLEMENTAL with a disclaimer that all remaining practice exams are unofficial.
- [ ] **Program expansion:** CCA-F is now one of four certifications — Associate–Foundations ($99), Developer–Foundations ($125), Architect–Foundations ($125), Architect–Professional ($175).
- [ ] **Scenarios:** exam guide v0.1 lists 6 (4 selected per sitting); the community guide, updated from candidate reports, now tracks 8 (4 of 8). Appendix note added, including the new Scenario 7 (Conversational AI Architecture Patterns) and reported Scenario 8 (Agentic AI Tools).
- [ ] **Exam guide version:** noted community reports of a v0.2 exam guide (June 30, 2026) with unchanged domains/weights/format; not yet primary-source-verified.

- [ ] **Registration link:** updated to the partner-academy certification page (the old access-request URL redirects there; partner login required).

Documentation links (all Anthropic docs migrated domains)

- [ ] API docs: docs.anthropic.com → **platform.claude.com/docs**; Claude Code & Agent SDK docs → **code.claude.com/docs**. All 15+ doc links updated to canonical URLs (old links redirect).
- [ ] "Building Effective Agents" moved from anthropic.com/*research*/ to anthropic.com/*engineering*/.
- [ ] **spec.modelcontextprotocol.io is dead** (connection failure, not a redirect) — replaced with modelcontextprotocol.io/specification. MCP Tools/Resources concept pages now redirect into the versioned spec; links updated.
- [ ] "Claude 4 Best Practices" page superseded by the generational "Prompting best practices" page (covers Fable 5 / Mythos 5 / Opus 4.8 / Sonnet 5 / Haiku 4.5, with per-model sub-pages); link and title updated.
- [ ] "Message Batches" renamed "Batch processing" in the docs; link and label updated.
- [ ] anthropic-cookbook repo renamed **claude-cookbooks**; appendix link updated to match Unit 15.

Courses (metadata corrections)

- [ ] DeepLearning.AI restructured URLs from /short-courses/ to /**courses**/; all four DLAI links updated.
- [ ] "Building Towards Computer Use" corrected to "Building **Toward** Computer Use" (actual title); runtime 1.5h → ~2h (1h 47m).
- [ ] "MCP: Build Rich-Context AI Apps" runtime 1.5h → ~2h (1h 58m). "Agent Skills with Anthropic" annotated at 2h 19m.
- [ ] Coursera "Building with the Claude API" is a single course (instructor Stephen Grider), not a specialization; link updated.
- [ ] All 10 Skilljar course URLs re-verified live. Skilljar course at claude-with-the-anthropic-api confirmed displaying as "Building with the Claude API."
- [ ] Retagging: Claude 101 and AI Fluency → OFFICIAL (both on Anthropic's partner-gated CCA-F prep-course list); Intro to Claude Cowork → ANTHROPIC (it is Anthropic-published). Noted that Anthropic's two official course lists (4-course public path, 7-course partner prep page) don't fully agree.

Community resources (added / removed / fixed)

- [ ] **Added:** claudecertificationguide.com (30 lessons, 240+ questions, mock exam, tracks exam guide v0.2), timothywarner-org/claude-architect, and the carolinacherry Claude Code study plugin.
- [ ] **Removed:** claudecertifications.com — site is down (origin DNS error at check time).
- [ ] **Fixed:** Tutorials Dojo URL slug changed (cca-f → ccar-f); guide_en.MD link 404'd due to case (actual file is guide_en.md); CertSafari question count 363 → 600+ (614 at check time).
- [ ] **License note softened:** the paullarionov repo has no detectable license file, so the previous "CC BY 4.0" claim was removed in favor of plain attribution.

Technical reference (updates)

- [ ] `stop_reason` table: added newer values `pause_turn` and `refusal` (marked as post-dating exam guide v0.1).
- [ ] `tool_choice` table: added `{"type": "none"}` and the note that `any` / `tool` are incompatible with extended thinking.
- [ ] D4 cliffs notes: added a real-world note on the native structured-outputs API feature (`output_config.format` , `strict` tool schemas) alongside the exam-taught `tool_use` pattern.
- [ ] D2 cliffs notes: noted the HTTP+SSE MCP transport was deprecated (2025-03-26 spec revision) in favor of Streamable HTTP.
- [ ] Verified as still accurate: domain weights (27/18/20/20/15), 60 questions/120 min/pass at 720, no guessing penalty, closed-book proctoring, CLAUDE.md hierarchy, `.claude/rules/` glob rules, SKILL.md frontmatter fields, PreToolUse/PostToolUse hooks, `-p` / `--output-format json` / `--json-schema` flags, `.mcp.json` vs `~/.claude.json` , `fork_session` / `--resume` , Batch API 50%/24h, MCP primitives, the out-of-scope list, and Claude Cowork (real product and course).

V1.0 — MAY 2026

- [ ] Initial release: 18-unit study plan, anti-patterns, cliffs notes, exam logistics, and exam guide v0.1 appendix.
