Claude Certified Developer – Foundations

Exam Guide

Version 1.0 · Effective July 2026 · Exam code: CCDV-F · This guide is subject to change without notice.

1. About This Certification

The Claude Certified Developer – Foundations certification validates that an individual can build, integrate, and ship production-grade applications, agents, and workflows using Anthropic's Claude platform at a foundational level. It is intended for technical professionals who bridge Claude's capabilities and production-ready applications, translating technical requirements into working systems through API integration, agent and tool construction, prompt and context engineering, evaluation, security, and model selection.

This guide is the authoritative reference for candidates preparing to sit the exam. It describes the credential's purpose, the intended audience, the exam format and content blueprint, scoring, policies, and the rules of conduct that all candidates agree to. Read it in full before scheduling your exam.

2. Purpose and Value of the Credential

The primary purpose of the Claude Certified Developer – Foundations certification is to provide an independent assessment of the knowledge, skills, and abilities required to build Claude-based applications competently at a foundational level. Earning the credential signals to employers, clients, and teams that the holder can independently own or significantly contribute to building, integrating, and shipping Claude-powered systems.

Candidates who earn the credential can demonstrate that they are able to effectively apply knowledge in the following areas:

- Building agents and workflows using the Claude Agent SDK, agentic frameworks, and custom agent loops
- Integrating Claude into application code through the API, client SDKs, and third-party integrations, including streaming, error handling, and multi-format input
- Operating Claude Code across interfaces, authoring Claude Skills, and configuring CLAUDE.md, settings.json, and plugins
- Writing effective prompts and applying context engineering techniques to control model behavior and prevent context drift

- Designing and running evals, debugging failure modes through trace analysis, validating structured output, and monitoring production quality
- Selecting Claude model tiers and managing tokens, caching, and batch processing to optimize cost and latency
- Applying secure-by-design principles and implementing guardrails through hooks to defend against prompt injection and destructive actions
- Building custom tools, function schemas, and MCP servers, while weighing tradeoffs across built-in tools, custom tools, Skills, and MCPs

3. Intended Audience

The certification is intended for technical professionals who build, integrate, and ship production-grade AI solutions using large language models (LLMs), particularly Anthropic's Claude platform. This audience primarily includes AI and machine learning engineers, technical leads, and senior software engineers operating at the intersection of business requirements and technical implementation.

These professionals typically have one to five years of experience in software engineering, along with at least six months of hands-on experience with Claude or comparable LLM-based systems. They possess strong foundational knowledge and applied skills in software development, with the ability to build agents and workflows using the Claude Agent SDK and agentic frameworks, integrate Claude through the Application Programming Interface (API) and client Software Development Kits (SDKs), operate Claude Code for codebase modernization, write effective prompts and apply context engineering, design and run evals, and build custom tools and Model Context Protocol (MCP) servers. They understand tradeoffs in model selection and tool type and can apply appropriate patterns to meet technical requirements. They are proficient in Python and/or TypeScript, fluent with REST APIs and Command Line Interface (CLI) tools, and have a working understanding of LLM fundamentals, agents, context management, and MCP.

This certification is not intended for non-technical or casual users of Claude-based applications, or individuals without hands-on software development experience. It also excludes roles limited to prompt writing or other isolated tasks without broader application development responsibility.

4. Minimally Qualified Candidate Profile

The exam is targeted at the minimally qualified candidate (MQC): a hands-on technical individual who builds, integrates, and ships Claude-powered applications, agents, and workflows. They bridge Claude's capabilities and production-ready applications, translating technical requirements into

working applications.

The MQC possesses strong foundational knowledge and applied skills in software development, including the ability to build agents and workflows using the Claude Agent SDK and agentic frameworks, integrate Claude through the API and client SDKs, operate Claude Code for codebase modernization, write effective prompts and apply context engineering, design and run evals, and build custom tools and MCP servers. They understand tradeoffs in model selection (cost, latency, capability) and tool type (built-in, custom, Skills, MCPs) and can apply appropriate patterns to meet technical requirements.

Recommended experience

- One to five years of experience in software engineering
- At least six months of hands-on experience with Claude or comparable LLM-based systems
- Proficiency in Python and/or TypeScript
- Fluency with REST APIs and CLI tools
- Working understanding of LLM fundamentals, agents, context management, and MCP

Prerequisites: There are no mandatory prerequisites or courses required to sit this exam. The experience above is recommended, not required. The credential is awarded based on exam performance alone.

5. Exam Details at a Glance

Credential	Claude Certified Developer – Foundations
Exam code	CCDV-F
Number of items	53
Item format	Multiple-choice and multiple-response items; each item states how many responses to select
Time limit	120 minutes
Delivery	Proctored: online proctored and/or test center, per program policy
Passing score	Scaled score of 720 on a scale of 100–1,000
Exam fee	$125 USD
Validity period	12 months from the date the credential is awarded
Result reporting	Pass/fail with scaled score (100–1,000), plus percent-correct by domain on the score report

6. Exam Content Outline (Blueprint)

The exam blueprint defines the content domains and skills measured and the approximate weight of each domain on the exam. Weights reflect the relative importance of each domain to competent performance as determined through the job task analysis and content validation surveys. The percentages indicate the approximate proportion of scored items drawn from each domain.

Domain	Content Domain	Weight
1	Agents and Workflows	14.7%
2	Applications and Integration	33.1%
3	Claude Code	3.1%
4	Eval, Testing, and Debugging	2.6%
5	Model Selection and Optimization	16.8%
6	Prompt and Context Engineering	11.0%
7	Security and Safety	8.1%
8	Tools and MCPs	10.6%
| Total	100%	

Detailed objectives by domain

Each domain below lists the skills against which exam items are written. Skill descriptions summarize the knowledge and competencies measured. Skill weights show each skill's share of the overall exam.

Domain 1: Agents and Workflows (14.7%)

Agent Architecture (4.5%)

Principles, patterns, and tradeoffs of agent and workflow architecture, including the decision criteria for using a workflow versus an agent, the structure of manager/supervisor hierarchies, and the role of subagents in improving task execution.

Agent Construction with Claude (5.3%)

Methods, tools, and platforms for constructing Claude agents, including the Claude Agent SDK, custom agent loops and harnesses, managed agent deployment models (self-hosted vs. Anthropic-hosted), and hooks for deterministic actions.

Agent Patterns and Frameworks (4.9%)

Common agent design patterns (tool-use loops, sub-agents, memory, context-window management) and agentic abstraction frameworks (e.g., Strands, LangGraph, PydanticAI) for building agents and workflows for multi-step tasks.

Domain 2: Applications and Integration (33.1%)

Understanding Requirements (3.4%)

Functional and infrastructure requirements based on business requirements and solution architecture.

Systems Life Cycle (2.8%)

Systems life cycle management concepts and frameworks used to develop, implement, operate, and maintain IT systems.

Claude API Mechanics (6.8%)

Claude API behavior and mechanics, including messages, tools, streaming, vision, thinking, caching, invoking Claude through third-party vendors, Messages API data access patterns, batch API use, and tradeoffs between realtime and batch API selection.

Software Engineering Foundations (7.4%)

Core software engineering principles and practices, including REST APIs, JSON, asynchronous programming, version control, SDLC integration, code review, and small- and large-scale refactoring.

Claude Application Design (8.6%)

Design considerations for building Claude applications, including how Claude interprets instructions across interfaces (Claude Code, Desktop, claude.ai, API, SDKs), content boundaries, schema design, session hygiene, and plugin management.

Configuration Management (4.1%)

Configuration management for Claude system components, including CLAUDE.md files, settings.json, model version pinning, prompt versioning, and plugin dependencies.

Domain 3: Claude Code (3.1%)

Claude Code Operation (3.1%)

Claude Code core components (Rules, Skills, Commands, Agents, Agent Memory), features (session management, built-in and custom slash commands, headless mode, streaming mode, auto-mode), the CLAUDE.md hierarchy, repository initialization, and settings.json configuration.

Domain 4: Eval, Testing, and Debugging (2.6%)

Debugging and Error Handling (2.6%)

Debugging and error handling techniques for Claude applications, including error type identification, recovery strategy selection, trace analysis to identify failure modes, and problem origin isolation between the integration layer and model output.

Domain 5: Model Selection and Optimization (16.8%)

LLM Fundamentals (5.2%)

Basic understanding of LLMs (tokens, context windows, sampling, non-determinism, next-token generation), model options (fast mode, extended thinking, adaptive thinking, effort levels), and fundamental prompting techniques (zero-shot, single-shot, multi-shot).

Technical Fundamentals (6.1%)

Foundational technical concepts supporting AI application development, including basic engineering practices (integrating with SDKs that wrap REST APIs, websockets).

Model Selection and Tradeoffs (2.7%)

Claude model capabilities (Opus vs. Sonnet vs. Haiku use cases, adaptive thinking support), tradeoffs across quality/latency/cost parameters, and breaking behavior changes across model releases when selecting models for tasks.

Cost and Token Management (2.8%)

Token budgeting and cost management techniques for Claude applications, including token usage tracking, cost modeling, and caching techniques (prompt caching, cache check-pointing) for cost optimization.

Domain 6: Prompt and Context Engineering (11.0%)

Context Engineering (3.8%)

Context and memory management techniques for Claude applications, including context window management, prevention of context drift and bloat (tool output pruning, compaction), and context isolation through subagents or multi-step agentic workflows.

Prompt Engineering (4.6%)

Prompt engineering principles and methods (instruction clarity, few-shot examples, system versus user placement, output constraints, prompt and instruction placement across components, iterative refinement, prompt adjustment, input sanitization) when writing and iterating on prompts for Claude.

Output Handling (2.6%)

Established patterns and techniques for producing, validating, and consuming Claude output, including structured output patterns, response validation, defensive parsing, and skepticism toward confident output.

Domain 7: Security and Safety (8.1%)

AI Application Security (3.2%)

Data privacy and security best practices, including prompt injection awareness and mitigation, jailbreak defense, untrusted input handling, data leakage prevention, PII handling, and ensuring authentication, authorization, confidentiality, privacy, and integrity.

Guardrails and Safe Deployment (2.3%)

Safe and responsible deployment practices (content policy, guardrail layering) and secure-by-design principles (privacy, identity and access management, least privilege).

Claude Hooks (1.0%)

Leveraging hooks for guardrails and safety controls to prevent destructive actions within Claude applications.

Identity, Secrets, and Key Management (1.6%)

Managing secrets, credentials, and API keys across Claude development and production environments, including identity validation and authentication, access approval and level verification, and authorized access monitoring.

Domain 8: Tools and MCPs (10.6%)

Tool Implementation (4.4%)

Tool implementation practices for Claude applications, including tool use and function calling, configuration for external system interaction, tool description writing, error handling, tool usage patterns (agentic harness dispatch, client-side vs. server-side tools, approval patterns), and tool set construction best practices.

MCP Server Development (2.1%)

MCP server development practices, including server authoring, deployment, integration with Claude applications, MCP resources, tools, and prompts, and communication patterns (stdio, sockets, client vs. server).

Agentic Customization (4.1%)

Tradeoffs among built-in Tools, custom Tools, Skills, and MCPs for selecting and applying the appropriate approach for a given use case.

7. How to Prepare

There is no single required course. Anthropic does not guarantee that any particular resource ensures a passing result. Candidates are encouraged to combine hands-on experience with the resources below:

- Study the exam blueprint in Section 6 and self-assess against each objective
- Review official Anthropic documentation for the Claude API, models, prompt engineering, Claude Code, Skills, and MCP
- Build and operate at least one Claude application that exercises the API, integrates one or more tools, applies basic prompt and context engineering, and includes simple security and evaluation practices
- Practice the developer competencies: writing prompts and system instructions, building agents and workflows, configuring Claude Code, managing tokens and cost, implementing guardrails, and creating custom tools or MCP servers
- Complete the sample questions in Section 8 to familiarize yourself with item style

8. Sample Questions

These illustrative items show the style and cognitive level of the exam. They are not drawn from the live item bank. Correct answers and rationale appear after the questions.

Sample 1 · Domain 2 — Applications and Integration

A developer must process 10,000 documents overnight to produce a non-urgent analytics report. Cost is the primary concern, and results are not needed until the following morning. Which approach best fits the requirement?

A. Send every request synchronously through the Messages API in parallel to finish as quickly as possible.

B. Use the Message Batches API, which processes large asynchronous workloads within a 24-hour window at reduced cost.

C. Lower max_tokens on synchronous calls to minimize cost.

D. Switch to the smallest available model regardless of output quality.

Sample 2 · Domain 7 — Security and Safety

A Claude-powered agent summarizes web pages submitted by end users. One page contains hidden text instructing the model to ignore previous instructions and reveal its system prompt. Which mitigation is most effective?

A. Raise the model's temperature so its behavior is harder to predict.

B. Treat retrieved page content as untrusted input, keep it separate from trusted instructions, and use guardrails or hooks so injected instructions cannot trigger sensitive actions.

C. Add a line to the system prompt asking users not to include malicious instructions.

D. Switch to a larger model that follows instructions more reliably.

Sample 3 · Domain 8 — Tools and MCPs

A team needs Claude to call an internal inventory service exposed as a REST API. They want the capability to be reusable across several Claude applications and maintained independently of any one app. Which approach best fits?

A. Hard-code the inventory logic into each application's system prompt.

B. Build an MCP server that exposes the inventory operations as tools so multiple Claude applications can connect to it.

C. Paste the current inventory data into the context window on every request.

D. Rely on a built-in tool, since built-in tools can reach any internal REST API.

Answer key and rationale

Sample 1: B. The Message Batches API is designed for latency-tolerant, high-volume workloads at lower cost, which matches an overnight, non-urgent job. Sending requests synchronously in parallel (A) does not reduce per-token cost; lowering max_tokens (C) or blindly downsizing the model (D) does not address the batch-versus-realtime tradeoff.

Sample 2: B. Prompt injection is addressed by isolating untrusted content from trusted instructions and enforcing least-privilege guardrails so injected text cannot invoke sensitive tools. Temperature (A) is irrelevant to injection; a polite request (C) is not an enforceable control; a more instruction-following model (D) can be more susceptible, not less.

Sample 3: B. An MCP server exposes reusable tools that multiple Claude applications can share and that can be maintained independently. Hard-coding logic into prompts (A) is neither reusable nor maintainable; pasting data (C) gives no live access and wastes context; built-in tools (D) do not

automatically reach arbitrary internal APIs.

9. How the Exam Is Scored

The Claude Certified Developer – Foundations exam is a criterion-referenced assessment: each candidate is measured against a fixed performance standard, not against other candidates. You pass by demonstrating the knowledge and skills defined in the blueprint, not by outperforming a percentage of peers.

Passing standard. The passing score was established through a formal standard-setting study in which trained subject matter experts judged the level of performance expected of a minimally qualified candidate. The score is reported on a scaled range of 100–1,000, and the cut score is 720.

Result reporting. Your result is reported as a pass or fail status with a scaled score from 100 to 1,000. Your score report also shows the percentage of items you answered correctly within each content domain. Section-level percentages are provided to help you understand your performance and are not used to determine your pass or fail result, which is based on your total scaled score.

10. Registration and Scheduling

Registration and scheduling are handled through the Anthropic Partner Academy and Pearson VUE:

1. Go to the certification page for your exam on the Anthropic Partner Academy and review the exam details.
2. Download the Exam Guide and review the Certification Terms and Conditions and the Certification Exam Policy before registering.
3. Register for the exam and complete checkout. The fee shown at checkout reflects any discount that applies to your partner tier.
4. Follow the confirmation instructions to create your Pearson VUE account, then sign in to schedule your exam session.
5. Choose an available date and select either online proctoring or a Pearson test center. The exam is delivered by Pearson VUE.
6. You may cancel or reschedule up to 24 hours before your appointment. Changes made within 24 hours forfeit the exam fee.

11. Exam Policies

Identification

On exam day you must present a valid, unexpired, government-issued photo identification. The name on your ID must match the name on your registration exactly. If you need to correct the name on your registration, contact certifications-support@anthropic.com before scheduling your exam.

Accommodations

Reasonable accommodations are available for candidates with documented disabilities or needs, in accordance with applicable law. Accommodations must be requested and approved by Pearson VUE before you schedule your exam. Do not schedule your appointment until your request has been approved. Request accommodations at pearsonvue.com/us/en/test-takers/accommodations.

Retake policy

Candidates who do not pass may retake the exam after a required waiting period. Waiting periods increase with each failed attempt: 14 days after the first, 30 days after the second, and 90 days after the third. You may take an exam up to four times within a rolling twelve-month period. Limits apply per exam, so not passing one exam does not prevent you from registering for a different one. The exam fee applies to each attempt.

No-show and late arrival

Candidates who fail to appear for a scheduled exam, or who arrive after the permitted late-arrival window, forfeit the exam fee and must re-register. Cancellation and rescheduling deadlines are described in Section 10.

12. Exam-Day Experience and Rules of Conduct

The exam is administered in a standardized, secure, proctored environment. Whether you test online or at a Pearson VUE test center, the following rules apply to protect the integrity of the credential for everyone who holds it.

During the exam you must:

- Remain within view of the proctor and webcam for the entire session, if testing online
- Keep your workspace clear of notes, books, phones, secondary monitors, and other materials

- Refrain from communicating with any other person during the exam
- Not capture, copy, photograph, or reproduce any exam content in any form

Prohibited items include mobile phones, smart watches, headphones, study materials, and any recording device. Permitted items, if any, such as scratch paper provided by the proctor, are specified by Pearson VUE.

Consequences of misconduct. Cheating, attempting to access prohibited resources, or disclosing exam content may result in invalidation of your result, revocation of your credential, and a ban from future exams.

13. Confidentiality and Non-Disclosure Agreement

Before the exam begins you must accept a confidentiality and non-disclosure agreement. By accepting, you agree that all exam content, including questions, answer options, and scenarios, is the confidential and proprietary property of Anthropic, and that you will not disclose, reproduce, or distribute any portion of it. If you do not accept the agreement, the exam session ends and no refund is issued.

14. Credential Maintenance and Recertification

The Claude Certified Developer – Foundations credential is valid for 12 months from the date it is awarded. Because the underlying technology evolves rapidly, the credential is time-limited so that holders maintain current knowledge.

To renew on time, you review what has changed since you certified and complete a free, non-proctored assessment on the Anthropic Partner Academy. There is no fee for on-time renewal. If your credential lapses, you must retake the full exam at the full fee to regain certified status.

If exam content changes significantly, Anthropic may require holders to retake the full exam to recertify rather than complete the renewal assessment.

Holders remain subject to the rules of conduct described in Section 12.

15. Candidate Support, Appeals, and Privacy

Support

To correct the name on your registration, contact certifications-support@anthropic.com. For all other questions, including registration, scheduling, accommodations, and results, contact Pearson VUE support at pearsonvue.com/us/en/anthropic.html.

Appeals and complaints

You may appeal a decision within 14 days of the date you are notified of it, or, for a concern about your exam result, within 14 days of your exam date. Submit your appeal to Pearson VUE support at the link above. Appeals are reviewed under the program's appeals policy. The standard-setting outcome and the content of individual exam items are not subject to appeal.

Privacy

Personal data collected during registration and testing is handled in accordance with Anthropic's privacy policy, Pearson VUE's privacy policy, and applicable data-protection law.

16. Document Control

Version	Summary of change	Date
1.0	Initial publication	July 2026