



To discuss this course and customizations:  
Call: +1 434-509-6890 or Email: [sales@cloudcontraptions.com](mailto:sales@cloudcontraptions.com)

# Claude Code in Practice

---

## Class Duration

14 hours of live training delivered over 2-3 days to accommodate your scheduling needs.

## Student Prerequisites

- Professional software development experience
- Comfort with the command line and terminal workflows
- Anthropic Claude API access or Claude Code subscription

## Target Audience

Senior software engineers, tech leads, and platform engineers who want to master terminal-driven agentic development using Claude Code. Ideal for teams building custom agentic pipelines, automating development workflows, or using Claude Code as the backbone of a CI-integrated AI workflow. Teams focused on delegating issue-to-PR work to Claude across the terminal, IDE, and github.com can continue with [Agentic Coding with Claude Code](#).

## Description

Claude Code is Anthropic's terminal-based agentic coding tool, designed for developers who want a powerful, scriptable AI coding partner that lives in the command line rather than an IDE plugin. This course covers the full Claude Code feature set from setup through advanced agentic workflows. You will learn CLAUDE.md project configuration, slash commands, Skills (the unified skill/command model), and Plugins for distributable capability bundles. From there we cover custom tools, hooks for lifecycle automation, subagent orchestration (including background and worktree-isolated subagents), and integration with CI/CD pipelines. Participants build increasingly autonomous workflows throughout, and by the end can assemble a complete agentic pipeline for a realistic development task.

## Learning Outcomes

- Install, authenticate, and configure Claude Code for a professional development environment.

To discuss this course and customizations:  
Call: +1 434-509-6890 or Email: [sales@cloudcontraptions.com](mailto:sales@cloudcontraptions.com)

- Author effective CLAUDE.md files for persistent project context and behavioral guidelines, and drive Claude Code with slash commands and interactive session management.
- Build and package Skills (the unified `.claude/skills/` model) that expose reusable capabilities through a slash-command interface, and bundle Plugins to share Skills, hooks, subagents, and MCP server configurations across teams.
- Define and invoke custom tools to extend Claude Code's capabilities with internal systems.
- Configure hooks for pre/post-task automation and safety guardrails.
- Orchestrate subagents for parallelizable and delegated coding tasks, including background and worktree-isolated execution.
- Integrate Claude Code into CI pipelines for automated review, test generation, and refactoring.
- Apply cost management and observability practices for Claude Code usage.

## Training Materials

Comprehensive courseware is distributed online at the start of class. All students receive a downloadable MP4 recording of the training.

## Software Requirements

Claude Code CLI installed, Anthropic API key, Git, and the student's preferred language runtime.

## Training Topics

### Setup and Configuration

- Installation and authentication
- Model selection and API key management
- Shell integration and editor passthrough
- Claude Code IDE extensions for VS Code, JetBrains (IntelliJ, PyCharm, WebStorm, Rider), and Zed (no official Visual Studio extension; third-party options only)
- Cost controls and usage monitoring

### CLAUDE.md: Project Context Files

- Purpose and anatomy of CLAUDE.md

To discuss this course and customizations:  
Call: +1 434-509-6890 or Email: [sales@cloudcontraptions.com](mailto:sales@cloudcontraptions.com)

- Project conventions, constraints, and workflow rules
- Hierarchical CLAUDE.md: repo, directory, and user levels
- Version-controlling CLAUDE.md with the team
- Coordinating CLAUDE.md with a GitHub Spec Kit constitution for spec-driven workflows

### **Slash Commands and Interactive Sessions**

- Built-in slash commands reference
- Custom slash commands in `.claude/commands/`
- Session management: resume, export, and clear
- Multi-turn task continuation
- Interactive vs. non-interactive (headless) modes

### **Skills**

- Skills as the unified `.claude/skills/` model
- Skill anatomy: frontmatter, prompts, allowed tools, model overrides
- Building reusable Skills for common engineering tasks
- Surfacing Skills as slash commands automatically

### **Custom Tools**

- Tool definition syntax and schema
- Reading files, running commands, and calling APIs
- Tool safety boundaries and confirmation gates
- Testing and debugging custom tools

### **Hooks and Lifecycle Automation**

- Pre-task and post-task hook types
- Input/output transformation hooks
- Safety hooks: blocking dangerous commands
- Integrating hooks with existing scripts

### **Subagents and Orchestration**

- When to spawn a subagent
- Subagent scope, isolation, and output collection
- Background subagents and worktree isolation
- Patterns: fan-out parallelism, pipeline sequencing
- Error handling and retry strategies



To discuss this course and customizations:  
Call: +1 434-509-6890 or Email: [sales@cloudcontraptions.com](mailto:sales@cloudcontraptions.com)

### **Plugins: Packaging and Distribution**

- Plugin anatomy: bundling Skills, hooks, subagents, and MCP servers
- Authoring an internal team plugin
- Versioning, distribution, and approval workflows
- Discovering and installing community plugins from the Claude Skills Marketplace (April 2026)
- Skill and plugin governance for team adoption

### **CI/CD Integration**

- Running Claude Code in GitHub Actions / GitLab CI
- Automated code review on pull requests
- Test generation as a CI step
- Secrets management in CI context

### **Observability and Cost Management**

- Token usage and cost attribution
- Logging and tracing agentic sessions
- Rate limit management for CI workloads
- Responsible usage and policy controls