Master Class: Agentic Software Engineering using Claude Code (ASECC)

ID ASECC **Price** CHF 4,890.—(excl. VAT) **Duration** 4 days

Who should attend

- Software engineers who want to move from prompting an AI to designing agentic engineering systems
- Tech leads and architects introducing Claude Code across a team and setting the harness conventions
- DevOps and platform engineers automating CI, deployments and infrastructure with headless agents
- Developers preparing for the Claude Certified Architect: Foundations certification

Prerequisites

- Comfortable with at least one programming language (labs use Python, C#, Angular and React)
- Confident on the command line and with Git, including branches and worktrees
- Basic understanding of REST APIs and JSON
- Nice to have, not required: familiarity with containers and SSH for the DevOps and deployment modules
- Nice to have, not required: a Claude Code installation with an active Claude account (a GitHub Codespace / DevContainer is provided)

Course Content

Module 1: Claude Code Fundamentals

- Setup: terminal, desktop, VS Code
- Model selection strategy
- Prompt caching, cost and tokens
- Permission model: tools, Bash, network
- Checkpoints and session safety
- Git worktrees: isolated checkouts
- Agents view: background sessions
- Artifacts: results as shareable pages

Module 2: Claude Code Configure Harness

- Settings hierarchy: global, project
- CLAUDE.md, scoped rules and @-mentions
- Slash commands
- MCP servers and tools
- Skill folders and sources
- Sub-agents under .claude/agents/
- Hooks: PreToolUse, PostToolUse, Stop
- Mods: extend and customize
- Auto-memory: MEMORY.md, four file types

Module 3: Creating & Managing Skills

- Skills vs. slash commands
- Invocation model, anatomy, body structure
- Authoring manually or from conversation
- Front matter and tool restrictions
- Composite skills and cross-skill delegation
- Testing: triggers and measurement
- Distribution: portable folder, skills API
- Repo facts: rules file contract

Module 4: Consuming & Designing MCP Tools

- MCP architecture, sessions, capabilities
- Remote auth: tokens, keys, OAuth
- Building a server in C#
- MCP Apps: interfaces inside hosts
- Tool design, errors, deferred loading

Module 5: Automate Claude Code from the Command Line

- Agent harness: tools, knowledge, actions
- Headless CLI: JSON, exit codes
- Least-privilege tool scoping
- CI gates and GitHub Actions
- Worktree lanes and policy gates
- Always-on agents with turn limits

Module 6: Scale Task Execution with Sub-Agents and Agent Teams

- Frontmatter: model, tools, permissions
- Context isolation for parallel agents
- Domain agents: backend, frontend, QA
- Evaluating agents with test cases
- Orchestration: delegation, loops, workflows
- Agent teams vs. sub-agents
- Enabling and steering agent teams
- Lead, teammates, task list, messaging

- Competing hypotheses in a team
- Plan approval and lifecycle hooks

Module 7: Optimize, Secure and Observe the Agentic Harness

- /doctor prompt-audit: dated instructions
- Caveman Compression: shorter CLAUDE.md
- Plugin Eval: graded skill experiments
- Prompt injection: quarantine, MCP responses
- Hook guardrails: PreToolUse hard stops
- OpenTelemetry: tracing, cost, failure diagnosis

Module 8: Bundle and Distribute Your Config as a Plugin

- Plugins vs. standalone configuration: namespacing
- Bundling: manifest, skills, hooks, MCP
- LSP servers and background monitors
- Default agent via plugin settings
- Local testing and reload
- Converting .claude/ configs into plugins
- Release tags and version identity
- Official and team marketplaces
- Plugins by Anthropic

Module 9: Agentic DevOps & Automation

- Deployment sub-agent: allowlists, prove-it contract
- SSH bootstrap and Ubuntu hardening
- Compose, Caddy, DNS before ACME
- Core Web Vitals, SEO loops

Module 10: Build and Host Agents with the Agent SDK and Managed Agents

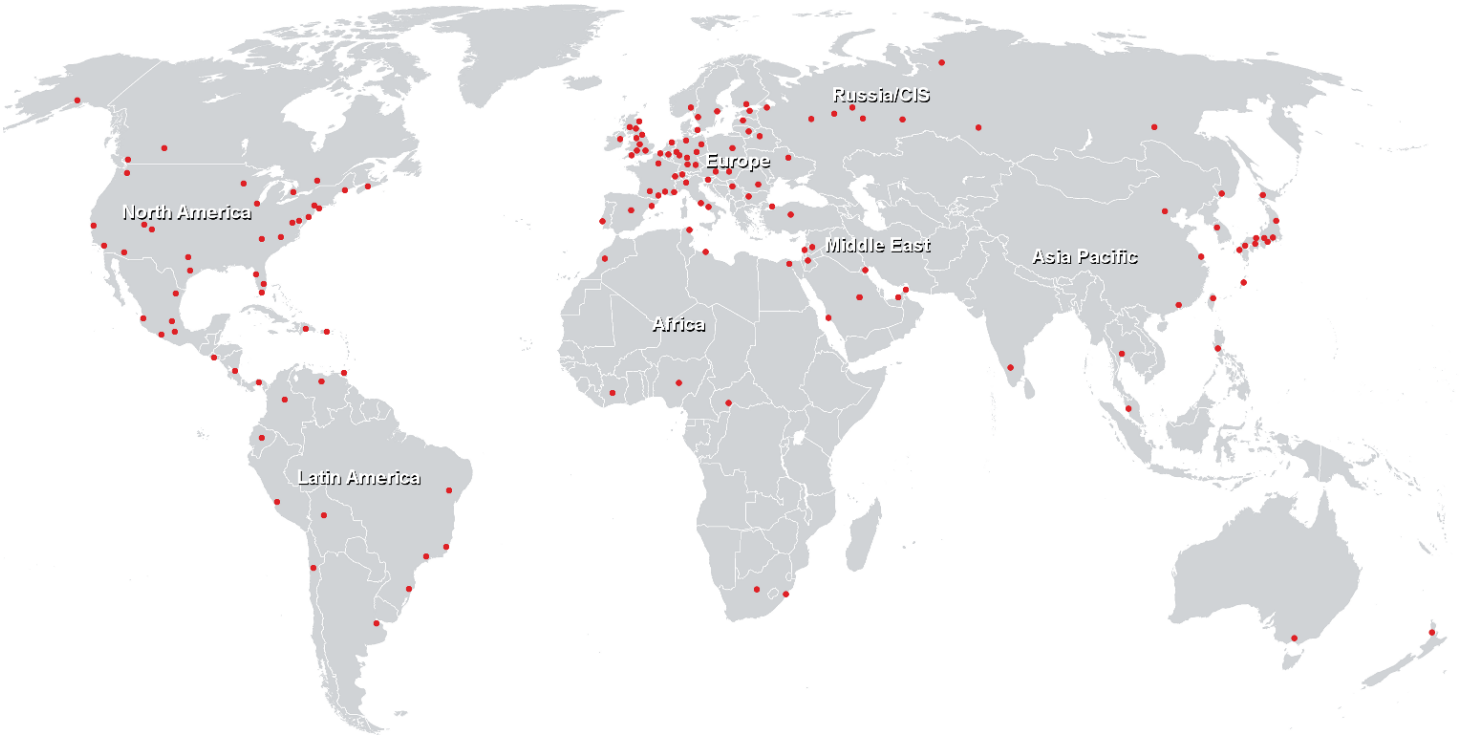
- Agent SDK: query() and sessions
- Agent SDK: parallel agents, orchestrator
- Managed Agents: Agent, Environment, Session
- Managed Agents: files, SSE stream
- Managed Agents: vaults, versions, policies
- Managed Agents: MCP servers, skills
- Human-in-the-loop: approval gates
- Multiagent: outcomes, memory stores

Module 11: Planning & Spec-Driven Development (Optional)

- Brownfield analysis: the current state
- Plan Mode and stepwise context
- Authoring and iterating specs
- Work packages with done criteria
- Why spec-driven development
- GitHub Spec Kit pipeline
- Spec-driven feature, end to end

Master Class: Agentic Software Engineering using Claude Code (ASECC)

Training Centres worldwide





Fast Lane Institute for Knowledge Transfer (Switzerland) AG

Husacherstrasse 3
CH-8304 Wallisellen
Tel. +41 44 832 50 80

info@flane.ch, https://www.flane.ch