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

ID ASECC **Preis** CHF 4'890.– (exkl. MwSt.) **Dauer** 4 Tage

Zielgruppe

- Software Engineers, die vom Prompten einer KI zum Design von Agentic Engineering Systems wechseln wollen
- Tech Leads und Architects, die Claude Code im Team einführen und die Harness Conventions festlegen
- DevOps und Platform Engineers, die CI, Deployments und Infrastructure mit Headless Agents automatisieren
- Entwickler, die sich auf die Zertifizierung Claude Certified Architect: Foundations vorbereiten

Voraussetzungen

- Sicher in mindestens einer Programmiersprache (die Labs nutzen Python, C#, Angular und React)
- Sicher auf der Command Line und mit Git, inklusive Branches und Worktrees
- Grundverständnis von REST APIs und JSON
- Nice to have, nicht erforderlich: Vertrautheit mit Containern und SSH für die DevOps- und Deployment-Module
- Nice to have, nicht erforderlich: eine Claude Code Installation mit aktivem Claude Account (ein GitHub Codespace / DevContainer wird bereitgestellt)

Kursinhalt

Modul 1: Claude Code Fundamentals

- Setup: Terminal, Desktop, VS Code
- Model Selection Strategy
- Prompt Caching, Kosten und Tokens
- Permission Model: Tools, Bash, Network
- Checkpoints und Session Safety
- Git Worktrees: isolierte Checkouts
- Agents View: Background Sessions
- Artifacts: Ergebnisse als teilbare Seiten

Modul 2: Claude Code Configure Harness

- Settings Hierarchy: Global, Project
- CLAUDE.md, Scoped Rules und @-Mentions
- Slash Commands
- MCP Servers und Tools
- Skill-Ordner und Quellen
- Sub-Agents unter .claude/agents/
- Hooks: PreToolUse, PostToolUse, Stop
- Mods: erweitern und anpassen
- Auto-Memory: MEMORY.md, vier Dateitypen

Modul 3: Creating & Managing Skills

- Skills vs. Slash Commands
- Invocation Model, Anatomie, Body-Struktur
- Authoring: manuell oder aus Conversation
- Front-Matter und Tool-Restrictions
- Composite Skills und Cross-Skill Delegation
- Testing: Trigger und Messung
- Distribution: portabler Ordner, Skills API
- Repo-Fakten: Rules-File-Contract

Modul 4: Consuming & Designing MCP Tools

- MCP Architecture, Sessions, Capabilities
- Remote Auth: Tokens, Keys, OAuth
- Einen Server in C# bauen
- MCP Apps: Interfaces in Hosts
- Tool Design, Errors, Deferred Loading

Modul 5: Claude Code von der Command Line automatisieren

- Agent Harness: Tools, Knowledge, Actions
- Headless CLI: JSON, Exit Codes
- Least-Privilege Tool Scoping
- CI Gates und GitHub Actions
- Worktree Lanes und Policy Gates
- Always-on Agents mit Turn Limits

Modul 6: Task Execution mit Sub-Agents und Agent Teams skalieren

- Frontmatter: Model, Tools, Permissions
- Context Isolation paralleler Agents
- Domain Agents: Backend, Frontend, QA
- Agents mit Test Cases evaluieren
- Orchestration: Delegation, Loops, Workflows
- Agent Teams vs. Sub-Agents
- Agent Teams aktivieren und steuern
- Lead, Teammates, Task List, Messaging

- Competing Hypotheses im Team
- Plan Approval und Lifecycle Hooks

Modul 7: Die Agentic Harness optimieren, absichern und beobachten

- /doctor Prompt Audit: veraltete Instructions
- Caveman Compression: kürzere CLAUDE.md
- Plugin Eval: bewertete Skill-Experimente
- Prompt Injection: Quarantäne, MCP Responses
- Hook Guardrails: PreToolUse Hard Stops
- OpenTelemetry: Tracing, Kosten, Fehlerdiagnose

Modul 8: Die eigene Config als Plugin bündeln und verteilen

- Plugins vs. Standalone Configuration: Namespacing
- Bundling: Manifest, Skills, Hooks, MCP
- LSP Servers und Background Monitors
- Default Agent über Plugin Settings
- Lokales Testing und Reload
- .claude/ Configs in Plugins überführen
- Release Tags und Version Identity
- Offizielle und Team Marketplaces
- Plugins von Anthropic

Modul 9: Agentic DevOps & Automation

- Deployment Sub-Agent: Allowlists, Prove-it-Contract
- SSH Bootstrap und Ubuntu Hardening
- Compose, Caddy, DNS vor ACME
- Core Web Vitals, SEO Loops

Modul 10: Agents mit dem Agent SDK und Managed Agents bauen und hosten

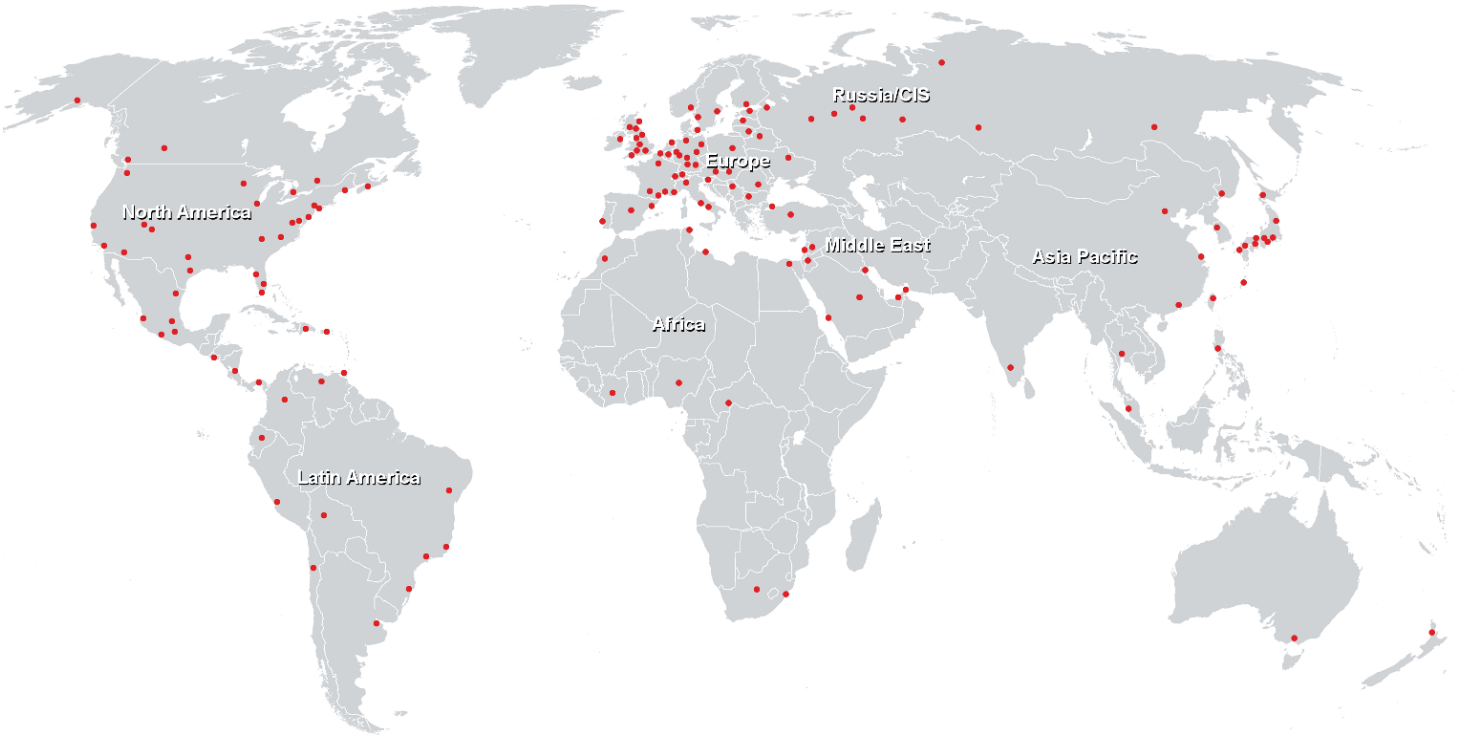
- Agent SDK: query() und Sessions
- Agent SDK: parallele Agents, Orchestrator
- Managed Agents: Agent, Environment, Session
- Managed Agents: Files, SSE Stream
- Managed Agents: Vaults, Versions, Policies
- Managed Agents: MCP Servers, Skills
- Human-in-the-Loop: Approval Gates
- Multiagent: Outcomes, Memory Stores

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

- Brownfield-Analyse: den aktuellen Stand erfassen
- Plan Mode und Stepwise Context
- Specs schreiben und iterieren
- Work Packages mit Done-Kriterien
- Warum Spec-Driven Development
- GitHub Spec Kit Pipeline
- Spec-Driven Feature, End-to-End

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

Weltweite Trainingscenter





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