BLUEMARK ACADEMY

Agentic Software Development with Claude Code Training

Duration: 2 Days · **Training format:** Virtual Classroom / Online

Description

Advanced 2-Day Program with Subagent and MCP Integration

The objective of this training is to enable software teams to use Claude Code not merely as a terminal-based AI tool, but as an extensible agentic development platform powered by subagent architectures and Model Context Protocol (MCP) integration.

By the end of the program, participants will be able to apply a reasoning-first development approach, design modular subagent architectures, integrate external systems via MCP, and build secure, enterprise-grade AI agent architectures.

Training Outcomes

- Subagent-based development approach
- External system integration capability with MCP
- Enterprise AI agent architecture design skills
- Token and cost optimization strategies
- Secure and auditable AI usage model

Outline

Day 1 – Claude Code Fundamentals and Subagent Architecture

1. Reasoning-First Development Approach

- Plan-first development
- Step-by-step reasoning
- Advantages of long context
- Deterministic generation techniques
- Planning before code generation approach

2. Subagent Concept

- Main agent (orchestrator) logic
- Task-based subagent structure
- Isolated context execution model
- Role-based task delegation

3. Types of Subagents

- Planner Agent
- Coder Agent
- Test Agent
- Reviewer Agent
- Refactor Agent

4. Subagent Orchestration Best Practices

- Task decomposition approach
- Defining clear input/output formats
- Enforcing structured output
- Plan → Execute → Validate loop
- Managing reasoning loop risks

Day 2 – MCP Integration and Enterprise Agent Architecture

5. What is Model Context Protocol (MCP)?

- MCP architecture and principles
- Tool exposure concept
- Context injection
- Claude Code + MCP integration structure

6. External System Integration with MCP

- Git repository access
- File system integration
- API calls
- Database querying
- Log analysis

7. Agent + MCP Security Model

- Permission boundaries
- Tool scoping
- Prompt injection risks
- Output validation
- Audit and logging approach

8. Enterprise Agent Architecture Design

- Monolithic vs modular agent approach
- Central orchestrator design
- Subagent lifecycle management

- Token and cost control
- On-prem vs cloud strategy

9. Final End-to-End Scenario

- Requirement analysis
- Planner subagent design
- Code and test generation
- Data integration via MCP
- Security validation
- Reporting and documentation generation