Agentic Software Development with Claude Code Training

Duration: **2 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Overview

This training is designed to help software development teams move beyond using Claude Code as a terminal-based AI assistant and adopt it as an extensible agentic development platform supported by subagent architectures and Model Context Protocol (MCP) integrations.

During this advanced two-day program, participants will explore planning-first development methodologies, the design of task-specific subagents, agent orchestration techniques, and structured output mechanisms. By working with agents assigned to different responsibilities within Claude Code, participants will learn how to organize software development activities into modular, controlled, and manageable workflows.

The second day focuses on connecting AI agents to external resources such as Git repositories, file systems, APIs, and databases through MCP. The program also examines access permissions, security boundaries, token and cost optimization, audit logging, and architectural approaches for enterprise AI systems.

Upon completing the training, participants will have developed the knowledge and practical skills needed to coordinate multiple agents in Claude Code, implement MCP-enabled integrations, and design secure, scalable, and auditable AI agent architectures for enterprise software development.

What You Will Learn

By the end of this training, participants will be able to:

- Apply plan-first software development practices using Claude Code.
- Understand Main Agent, Orchestrator, and Subagent architectures.
- Develop specialized subagents for planning, coding, testing, and code review.
- Manage task distribution and workflows across multiple AI agents.
- Integrate Git, file systems, APIs, and databases through MCP.
- Manage agent security, access permissions, and Prompt Injection risks.
- Apply token usage and cost optimization strategies.
- Design secure, scalable, and auditable enterprise AI agent architectures.

Outline

Day 1 – Claude Code Fundamentals and Subagent Architecture

1. Planning-First Approach to Software Development

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Organizing development workflows using the plan-first methodology
- Applying step-by-step reasoning techniques
- Exploring the benefits of long-context capabilities
- Techniques for achieving deterministic code and output generation
- Performing analysis and planning before generating code

2. Understanding Subagents and Their Architecture

- The role and operating principles of the Main Agent (Orchestrator)
- Designing task-oriented subagent structures
- Executing tasks within isolated context environments
- Delegating responsibilities through role-based task assignment

3. Subagent Types and Responsibilities

- Planner Agent: Planning and task organization
- Coder Agent: Code development and implementation
- Test Agent: Test creation and execution
- Reviewer Agent: Code review and evaluation
- Refactor Agent: Code improvement and restructuring

4. Best Practices for Subagent Orchestration

- Breaking complex tasks into manageable components through task decomposition
- Establishing clearly defined input and output formats
- Enforcing structured outputs across agent workflows
- Implementing the Plan → Execute → Validate cycle
- Identifying and managing risks associated with repetitive reasoning loops

Day 2 – MCP Integration and Enterprise Agent Architecture

5. Introduction to Model Context Protocol (MCP)

- Core principles and architectural structure of MCP
- Understanding the concept of tool exposure

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- How context injection works
- Architecture for integrating MCP with Claude Code

6. Connecting External Systems Through MCP

- Accessing Git repositories
- Integrating file systems
- Executing API requests
- Performing database queries
- Analyzing application and system logs

7. Security Model for Agents and MCP

- Establishing access permissions and permission boundaries
- Restricting tool capabilities through tool scoping
- Evaluating prompt injection threats
- Implementing validation for tool outputs
- Monitoring agent activities through audit and logging practices

8. Designing Enterprise AI Agent Architectures

- Comparing monolithic and modular agent architectures
- Designing a centralized orchestrator
- Managing the lifecycle of subagents
- Controlling token consumption and operational costs
- Evaluating on-premises and cloud deployment strategies

9. Final Project: End-to-End Agentic Software Development

- Analyzing software requirements
- Designing a Planner Subagent architecture
- Generating code and test scenarios
- Implementing data integrations through MCP
- Performing security checks and validation
- Producing reports and technical documentation

Agentic Software Development with Claude Code Training

Duration: **2 Day**

BİLGİNÇ
IT Academy
info@bilginc.com
bilginc.com