

ARTIFICIAL INTELLIGENCE

INTERMEDIATE

3 DAYS

Agentic Development with GitHub Copilot

Copilot is no longer autocomplete. Learn to work with agent mode inside the IDE, delegate whole issues to the cloud coding agent, drive the Copilot CLI, and extend all three with persistent instructions and MCP servers connected to your own systems.

Why Agentic Development with Copilot?

Delegate whole tasks

Move from line-level suggestions to issues that come back as reviewable pull requests.

Persistent project context

Encode your standards once so every agent run follows them.

Know when not to delegate

Judgement on where the agent loop pays off and where it costs you time.

Extend to your systems

Connect internal tools and data through MCP rather than copy-pasting context.

What you'll be able to do

- ✓ Distinguish agent mode, the cloud coding agent and the Copilot CLI, and choose between them
- ✓ Write delegations with clear intent, scope and stop conditions
- ✓ Operate the agent loop across multi-file edits, terminal commands and self-correction
- ✓ Assign issues to the coding agent and review the resulting pull requests critically
- ✓ Use the Copilot CLI plan mode, autopilot and parallel subagents
- ✓ Encode persistent context using custom instructions, AGENTS.md and Copilot Spaces
- ✓ Extend Copilot with MCP servers connected to internal tools and data
- ✓ Select models per task and manage premium request consumption

Who this is for

- Software developers and engineers
- Platform and DevOps engineers
- Tech leads and engineering managers
- Architects evaluating agentic development workflows

Curriculum

01 The Copilot Surface in 2026

- Completions, chat, agent mode, the cloud coding agent and the CLI compared
- Choosing the right surface for a given piece of work
- Multi-model selection and how model choice changes results
- Premium requests: what consumes quota and how to monitor it
- Where Copilot sits against other agentic coding tools

02 Agent Mode in the IDE

- How the agent plans a task, edits across files and runs terminal commands
- Self-correction loops, iteration limits and when it stalls
- Approving, interrupting and steering a run in progress
- Agent mode across VS Code and JetBrains
- **Hands-on:** deliver a multi-file feature end to end in agent mode

03 Writing Delegations That Work

- Intent, scope and explicit stop conditions in a delegation
- Defining success criteria the agent can verify for itself
- Decomposing large tasks into agent-sized units
- Diagnosing why a run produced unusable output, and correcting it
- **Hands-on:** rewrite a failed delegation until it succeeds

04 Persistent Context: Instructions, AGENTS.md and Spaces

- copilot-instructions.md for always-on project context
- AGENTS.md for agent-specific workflow rules and definitions of done
- Nesting instruction files across a monorepo
- Copilot Spaces for curated grounding across files, issues, PRs and docs
- **Hands-on:** author instruction files and measure the change in output

05 The Cloud Coding Agent

- Assigning a GitHub issue and receiving a pull request back
- How the agent executes in the background and what it has access to
- Writing issues the coding agent can actually complete unattended
- Reviewing agent-authored pull requests without rubber-stamping them
- **Hands-on:** take an issue from assignment through to a merged PR

06 Copilot CLI

- Installing and authenticating the CLI alongside your IDE setup
- Plan mode for scoping work before execution
- Autopilot mode and the guardrails worth setting first
- Parallel specialised subagents for explore, task, review and plan work
- Repository memory, hooks and plugins
- **Hands-on:** drive a complete task from the terminal

07 Extending Copilot with MCP

- Model Context Protocol fundamentals and the built-in GitHub MCP server
- Connecting internal APIs, databases and documentation sources
- Scoping access, handling credentials and applying least privilege
- Copilot Extensions and how they differ from MCP connectors
- **Hands-on:** connect an MCP server and complete a task through it

08 Working at Scale and Knowing When to Stop

- Running multiple agents in parallel without collisions
- Review load as the real constraint on agentic throughput
- Recognising when delegation costs more than doing the work yourself
- Team conventions for labelling and reviewing agent-generated code
- Measuring whether the workflow is genuinely faster, not just busier