ARTIFICIAL INTELLIGENCE | BEGINNER | 2 DAYS

GitHub Copilot for Developers

A practical grounding in Copilot for day-to-day development. Covers setup across IDEs, effective prompting, refactoring, testing and debugging with AI assistance, and the review habits that keep code quality intact. Language-agnostic, with labs run in your own stack.

Why GitHub Copilot for Developers?

Speed without decay

Ship faster while keeping the review discipline that protects the codebase.

Prompting that works on code

Practical technique for intent, scope and constraints, not generic prompt theory.

Language-agnostic

Labs run in your team's actual stack rather than a fixed demo language.

Honest about limits

Clear guidance on what to always verify and where Copilot reliably misleads.

What you'll be able to do

- Set up and configure Copilot across VS Code, JetBrains and Visual Studio
- Use inline completions and next edit suggestions effectively
- Work fluently in Copilot Chat across ask and edit modes
- Prompt for code with clear intent, scope and constraints
- Refactor, document and debug existing code with AI assistance
- Generate and strengthen test coverage
- Review AI-generated output critically and know what always needs verification
- Apply privacy, licensing and responsible-use practice in daily work

Who this is for

- Software developers new to AI coding assistants
- Engineers standardising Copilot use across a team
- Technical leads evaluating Copilot for adoption
- Graduate and early-career engineers

Curriculum

01 Getting Started with Copilot

- What Copilot is and how suggestions are actually generated
- Setup across VS Code, JetBrains and Visual Studio
- Plans and which features are available at each tier
- Settings, keybindings and turning it off where it does not belong

02 Completions and Inline Assistance

- Accepting, rejecting and cycling through suggestions
- Next edit suggestions and following a change through a file
- Comment-driven development and where it breaks down
- Working across languages and unfamiliar frameworks
- **Hands-on:** implement a feature using completions alone

03 Copilot Chat in Practice

- Ask mode for explanation, exploration and unfamiliar code
- Edit mode for scoped multi-file changes
- Referencing files, selections and workspace context
- Slash commands and built-in shortcuts worth memorising
- **Hands-on:** build a small feature using chat

04 Prompting for Code

- Intent, scope and constraints in a code-specific context
- Supplying the right context without overloading the request
- Asking for alternatives and trade-off explanations
- Recovering when suggestions head in the wrong direction

05 Refactoring, Testing and Debugging

- Understanding unfamiliar or legacy code quickly
- Refactoring safely by establishing test coverage first
- Generating and extending unit tests
- Debugging, error explanation and documentation generation
- **Hands-on:** refactor and test an existing module in your own stack

06 Reviewing AI Output and Responsible Use

- Common failure modes: code that is plausible and wrong
- What always needs verification before it reaches a commit
- Security, secrets and dependency risks in generated code
- Licensing, content exclusion and organisational policy
- Building a personal review checklist you will actually follow