CREDO SYSTEMZ
Simplifying IT

Copilot Training for Developers

Section 1: Introduction to AI in Development

- Evolution of AI in coding
- **GitHub Copilot** – code generation, debugging, testing
- GitHub Copilot vs Microsoft Copilot
- How AI understands prompts and context in code
- Limitations and best practices
- Future of AI in Development

Section 2: Getting Started with GitHub Copilot

- Creating a GitHub account
- Enabling GitHub Copilot in your account
- Installing the GitHub Copilot extension in VS Code
- Installing in JetBrains IDEs
- Configuring workspace settings for optimal suggestions
- Managing permissions and data privacy settings
- Understanding inline suggestions (ghost text)
- Using block suggestions for larger code snippets
- Shortcuts and hotkeys for faster acceptance

Session 3: Prompt Engineering Fundamentals

- Crafting effective prompts for generating new code
- Converting comments to code and code refactoring techniques
- Developer-focused scenarios and use cases
- Best practices for building complex features with prompts
- Prompt debugging to improve code generation outcomes

Section 4: AI-Powered Code Generation

- Writing natural language prompts for code
- Generating small functions and reusable components
- Complete classes and modules from scratch
- Using Copilot Chat to explain existing code
- Refactoring old code
- Adding inline comments automatically
- Creating API integration scripts
- Automating repetitive code structures
- Building prototypes and MVPs quickly
- Combining AI-generated and manual code effectively

Section 5: Testing & Quality with AI

- Auto-generating unit test cases from code
- Test-driven development workflows
- Generating API test scripts
- Creating test data automatically
- Writing integration tests with minimal input
- Detecting missing test scenarios
- Improving test coverage
- Reviewing AI-generated tests
- Reducing human error in test writing
- Using AI for regression test updates

Section 6: Debugging & Optimization

- Identifying syntax errors automatically
- Suggesting fixes for logical errors
- Explaining the cause of code bugs
- Optimizing database queries
- Improving algorithm efficiency

- Refactoring long functions into smaller
- Detecting security vulnerabilities
- Automating repetitive debugging steps
- Learning debugging patterns from suggestions

Section 7: Multi-language Development

- Switching between programming languages
- AI assisted coding in Python
- AI assisted coding in JavaScript/TypeScript
- AI assisted coding in Java
- Handling different syntax and structure automatically
- Using Copilot for front-end, back-end, and full-stack projects
- Maintaining consistent code style across languages

Real Time Projects:

- Employee Management System
- E-Commerce Product Catalog & Ordering System
- Bug Tracker with AI-assisted Testing
- Real-Time Chat Application
- API Integration and Automation