AI Driven Development using GitHub Copilot

Course Duration: 2 Days

Learn how to accelerate app development by using GitHub Copilot and GitHub Copilot Chat in a Visual Studio Code environment.

Prerequisites

- An active subscription for GitHub Copilot is required for either a personal GitHub account or a GitHub account managed by an organization or enterprise.
- One or more years of code develop experience is recommended.
- Some experience developing C# applications in Visual Studio Code with the C# Dev Kit extension is recommended.

Module 1: GitHub Copilot Essentials

In this module, we'll cover the most common questions about GitHub Copilot, and we will see how we can use GitHub Copilot to accelerate the pace of software development and deliver more value to our customers.

- What GitHub Copilot can do for your organization
- Establishing trust in using GitHub Copilot
- Understanding the GitHub Copilot data pipeline
- Measuring the impact of GitHub Copilot
- Empower developers with AI Policy and Governance
- Tips for a successful rollout of GitHub Copilot
- GitHub Copilot seat management and provisioning
- Understanding billing for GitHub Copilot

Module 2: Get started with GitHub Copilot

This module introduces developers to the GitHub Copilot products, the benefits that GitHub Copilot provides to developers, the GitHub Copilot and GitHub Copilot Chat product features, and the GitHub Copilot extensions for Visual Studio Code.

- Introduction
- Examine AI tools from GitHub, OpenAI, and Microsoft
- Examine GitHub Copilot tools, benefits, and limitations
- Exercise - Install GitHub Copilot extensions for Visual Studio Code
- Examine the autocomplete features of the GitHub Copilot extension
- Examine the AI assistance features of the GitHub Copilot Chat extension
- Exercise - Configure GitHub Copilot extensions for Visual Studio Code

Module 3: Generate documentation using GitHub Copilot tools

This module explores the generation of code explanations, project documentation, and inline code comment documentation using the GitHub Copilot Chat extension for Visual Studio Code.

- Introduction
- Examine the "document" and "explain" features of GitHub Copilot Chat
- Exercise - Generate code explanations using GitHub Copilot Chat
- Exercise - Generate project documentation by using GitHub Copilot Chat
- Exercise - Generate inline code documentation by using GitHub Copilot Chat
- Exercise – Complete the "code documentation" challenge
- Review the "code documentation" solution

Module 4: Develop code features using GitHub Copilot tools

This module explores using GitHub Copilot and GitHub Copilot Chat suggestions to create new code. Autocompletion and code update suggestions are generated, managed, and implemented using the GitHub Copilot extensions for Visual Studio Code.

- Introduction
- Examine the code development features of GitHub Copilot
- Examine GitHub Copilot best practices
- Exercise - Create code by using code line completions
- Exercise - Create code by using GitHub Copilot Inline Chat
- Exercise - Complete the create new code challenge
- Review the create new code solution
- Exercise - Complete the code logic challenge
- Review the code logic solution
- Exercise - Convert code from one programming language to another

Module 5: Develop unit tests using GitHub Copilot tools

This module explores using GitHub Copilot and GitHub Copilot Chat to create unit tests.
Exercises provide practical experience creating unit test projects and running unit tests in Visual Studio Code.

- Introduction
- Examine the unit testing tools and environment
- Exercise - Create unit tests by using GitHub Copilot Chat
- Exercise - Create unit tests for specific conditions by using GitHub Copilot
- Exercise - Complete the "create unit tests" challenge

- Review the "create unit tests" solution

Module 6: Implement code improvements using GitHub Copilot tools

This module explores using GitHub Copilot Chat to develop improvements for an existing codebase. Exercises provide practical experience implementing GitHub Copilot suggestions that improve code quality, reliability, performance, and security.

- Introduction
- Examine GitHub Copilot support for code improvements
- Exercise - Improve code quality by using GitHub Copilot Chat
- Exercise - Improve code reliability and performance by using GitHub Copilot Chat
- Exercise - Improve code security by using GitHub Copilot Chat
- Exercise - Complete the "app improvement" challenge
- Review the "app improvement" solution

Module 7: Guided project - Accelerate app development using GitHub Copilot tools

This Module explores using GitHub Copilot to accelerate development of an end-to-end project. GitHub Copilot is used to explain unfamiliar code, generate project documentation, develop a new app feature, develop unit tests, and implement improvements.

- Introduction
- Prepare the development environment
- Exercise - Analyse and document code using GitHub Copilot tools
- Exercise - Develop code features using GitHub Copilot tools
- Exercise - Develop unit tests using GitHub Copilot tools
- Exercise - Refactor and improve code sections using GitHub Copilot tools