✓ 100 XP

Introduction

3 minutes

This module explores using the GitHub Copilot Chat extension for Visual Studio Code to implement code improvements. You learn about making improvements to code quality, code reliability, code performance, and code security. Prompts are developed for each improvement area and submitted to GitHub Copilot for suggestions on how to improve selected code. Exercises teach you how to craft prompts that generate suggestions for specific improvement areas. Exercises also demonstrate how to implement code updates suggestions, accelerate your development process, and improve the quality of your code.

Imagine you're a developer working on a project that requires secure and efficient code. Delivery deadlines are approaching and the codebase that you inherited is more proof-of-concept than production code. You're concerned that you won't have enough time to thoroughly review the codebase and implement the required updates. You need a tool that can quickly identify areas for improvement in the existing code. The tool should also accelerate the development process for any updates you decide to implement. You want to investigate updates that address code quality, reliability, performance, and security. You hear that GitHub Copilot can help developers evaluate code, identify areas for improvements, and efficiently update their codebase. You're looking forward to accelerating your code improvement process using GitHub Copilot.

The topics covered in this module include:

- Exploring GitHub Copilot's code improvement tools.
- Using GitHub Copilot's quick fix options.
- Code refactoring with GitHub Copilot best practices.
- Refactor code with GitHub Copilot inline chat.
- Refactoring code with GitHub Copilot ask, edit, and agent modes.
- Exploring GitHub Copilot code review features.
- Refactor C# Code with GitHub Copilot.

After completing this module, you’ll be able to:

- Use GitHub Copilot's Chat view modes to analyze, edit, and execute tasks effectively.
- Apply quick fix options to resolve coding issues, test failures, and terminal errors using tools like Fix smart actions and Inline Chat.
- Refactor code to improve readability, reduce complexity, and enhance maintainability by leveraging Copilot's contextual tools and best practices.

- Conduct code reviews using GitHub Copilot's review selection and review changes features to ensure quality, reliability, and security.
- Complete a hands-on exercise to refactor and optimize a C# application using GitHub Copilot.

ⓘ **Important**

To complete this GitHub Copilot training, you must have an active subscription for GitHub Copilot in your personal GitHub account (includes the GitHub Copilot Free plan), or you must be assigned to a subscription managed by an organization or enterprise. Module activities may include GitHub Copilot suggestions that match public code. If you're a member of an organization on GitHub Enterprise Cloud who has been assigned a GitHub Copilot subscription through your organization, the setting for suggestions matching public code may be inherited from your organization or enterprise. If your account blocks suggestions that match public code, module activities may not work as expected.

Next unit: Review GitHub Copilot's code improvement features

Next >