

AI-151: Using Codeium for code and test generation as well as code refactoring, explanation and documentation

Course Length: 8 training hours

Course Description:

Codeium is one of the most popular alternatives to Github Copilot among coding assistant applications which help programmers at generating, refactoring, explaining and documenting programming codes in dozens of languages as well as at test generation for the same codes. Besides its free tier license option for individual programmers with a broad feature set it also has Teams and Enterprise licensing options with additional features, and the options of fine tuning the application to the codebase of the team or enterprise as well as running Codeium locally at the customer's site.

This 8-hours training teaches students how to install and configure Codeium as well as the main features and their effective use of the Teams and Enterprise licensing option.

Main topics:

- Introduction and main features of Codeium
- Codeium autocompletion
- Codeium Chat
- Advanced features (Explain, Document, Refactor, Unit-test generation)
- Adding context
- Writing machine friendly code, comments and commits (Context is everything)

Structure: roughly 50% lecture, 50% hands on lab exercises

Target audience: Software developers and testers as well as their technical managers who want to use Codeium in their daily work.

Prerequisites: Basic understanding of AI concepts, experience in using the programming language where Codeium would be used as a coding assistant, user experience with ChatGPT or similar chatbots.

Detailed Course Outline

Module 1. Introduction and main features of Codeium

- Main features and usage types of Codeium
- Installation and registration
- Setting custom options
- Using different language models (GPT and Codeium's proprietary ones)
- **Lab**

Module 2. Codeium autocomplition

- Auto completion
- Accept suggestion, accept next word, next suggestion
 - With cycle menu
 - With shortcuts
- **Lab**

Module 3. Codeium Chat

- Chat window on the side panel, for proper conversation
 - Linked sources
- Inline chat window for code generation instructions
 - Diff view of the change
 - Refine the output with additional prompts
- Limitations of Codeium Chat
- **Lab**

Module 4. Prompt engineering for code generation

- Prompt Engineering Introduction
- Key Elements of Prompts
- Prompting Techniques:
 - "Just ask" or iterative technique
 - Zero-Shot prompts
 - One- and few-shots technique
 - Chain of Thoughts prompting
- **Lab**

Module 5. Advanced features

- Explain
 - Generate explanation for a code-block: With Code Lens, with Chat and linking the code block with @ in the Chat window as well as with code block selection and global menu
- Documentation
 - Generate language specific documentation in the right format
- Refactoring
 - Language specific refactoring options in the refactor menu
- Unit-test generation
- **Lab**

Module 6. Adding context

- Adding context to get better suggestions from Codeium
- Adding additional context to the model, not present in the codebase
- Handling library updates as a specific additional context
- **Lab**

Module 7. Writing machine friendly code, comment, commit
(Context is everything)

- Link relevant functions in your chat conversation
- Add a top-level comment to each program file
- Meaningful names
- Well scoped, helpful comments
- Keep to a standard (formatter, linter)
- **Lab**

Module 8. (optional) Security aspects of using Codeium

- How secure is the code generated by Codeium?
- Prompt engineering for security
- Practical attacks against code generation tools
- **Lab**