

## AI-151: Using Codeium/Windsurf as coding assistant and generator (code and test generation, code refactoring, explanation and documentation)

**Course Length:** 1 day or 2 half days

### **Course Description:**

Codeium/Windsurf is one of the most popular alternatives to Github Copilot among coding assistant applications which help programmers to generate, refactor, explain and document program codes in dozens of languages as well to generate tests for the same codes.

An important feature of Codeium/Windsurf is the ability to use not only the current file but also local program libraries as well as public and private repos as context for code understanding and generation. This way Codeium/Windsurf can learn the programming standards and styles of the team or the whole organization and can produce code that fits well into the existing code base and can use public or company program libraries.

**Training objectives:** at the end of the training participants

- understand the basic LLM and Codeium/Windsurf concepts
- gain practical skills in prompt engineering
- gain practical skills in using Codeium/Windsurf for code and test generation, code refactoring, explanation and documentation.

### **Main topics:**

- LLMs in a nutshell
- Introduction and main features of Codeium/Windsurf
- Test Driven Development with Codeium/Windsurf
- Codeium/Windsurf autocompletion
- Codeium/Windsurf Chat
- Prompt engineering for code generation
- Advanced features (Explain, Document, Refactor, Unit-test generation)
- Adding context

**Structure:** roughly 50% lecture, 50% hands on lab exercises. The lab exercises are available either in a VS Code + Python, VSCode + C, C++ or IntelliJ + Java environment.

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

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

This training is part of the AI portfolio of Component Soft which explores essential AI topics, such as:

- AI-110: Intro to GenAI with Large Language Model (LLMs) and LLM-based apps.
- AI-151: Using Codeium/Windsurf as coding assistant
- AI-434: GenAI Application Development with LLMs

## Detailed Course Outline

### **Module 1. LLMs in a nutshell**

- Main parts and working of LLMs in a nutshell
- Most important base LLM vendors and models
- LLMs used in Codeium/Windsurf

### **Module 2. Introduction and main features of Codeium/Windsurf**

- Main features of Codeium/Windsurf
- VS Code or IntelliJ basics
- Core features of Codeium/Windsurf
- Different Types of Interaction with Codeium/Windsurf
- Using different LLMs (GPT, Claude Sonnet and Codeium/Windsurf's proprietary ones)
- Test Driven Development with Codeium/Windsurf
- **Lab**

### **Module 3. Codeium/Windsurf autocompletion**

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

### **Module 4. Codeium/Windsurf Chat**

- Codeium/Windsurf Chat: Inline commands
- Codeium/Windsurf Chat: Chat panel
- Codeium/Windsurf Chat: Tips & Tricks
- **Lab**

### **Module 5. Prompt engineering for code generation**

- The 2 golden rules of prompt engineering
- 10 Prompting rules of thumb
  - Be concise and give clear instructions
  - Be specific and include relevant details
  - Add positive and negative prompts
  - Define roles for the LLM
  - Define roles for the LLM's audience
  - Provide examples for the solution or response style (one-shot or few-shot prompting)
  - Add relevant context
  - Divide difficult tasks into subtasks (Prompt Chaining)
  - "Let's think step by step" (Chain of Thought)
  - Let the LLM ask questions
- **Lab**

## **Module 6. 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 7. Adding context**

- Context handling in Codeium/Windsurf
- Custom chat instructions
- Pinned contexts
- Local indexes
- Setting files and directories which won't be part of the context
- Setting up team-wide repos for context
- Using team-wide repos
- Lab