AI-161: Using Amazon Q as coding assistant and generator

(code and test generation, code refactoring, explanation and documentation)

Course Length: 1 day or 2 half days

Course Description:

Amazon Q 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.

Training objectives: at the end of the training participants

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

Main topics:

- LLMs in a nutshell
- Introduction and main features of Amazon Q
- Test Driven Development with Amazon Q
- Amazon Q autocompletion
- Amazon Q 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 VSCode + Java environment.

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

Prerequisites: Basic understanding of AI concepts, experience in using the programming language where Amazon Q 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-161: Using Amazon Q 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 Amazon Q

Module 2. Introduction and main features of Amazon Q

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

Module 3. Amazon Q autocompletion

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

Module 4. Amazon Q Chat

- Amazon Q Chat: Inline commands
- Amazon Q Chat: Chat panel
- Amazon Q 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 Amazon Q
- 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