Accelerating Development with AI: Practical Workflows for Java, .NET, and Python

Duration: 8 hours (1 Day)

Custom Course

Course Overview:

This course empowers developers to harness AI-assisted coding tools—such as GitHub Copilot and Copilot Chat—to streamline development workflows, improve code quality, and accelerate delivery across Java and .NET based projects along with Python-based data science projects. Participants will gain hands-on experience applying AI to real-world scenarios.

Prerequisites:

• **Basic to Intermediate Programming Experience** Familiarity with at least one of the following languages: Java, C#, or Python.

• **Understanding of Software Development Workflows** Experience with IDEs, version control (e.g., Git), and common development patterns.

• **Familiarity with Unit Testing Concepts** Exposure to testing frameworks like JUnit (Java), xUnit/NUnit (C#), or pytest (Python).

• **Basic Knowledge of Web Development** Understanding of REST APIs, HTTP methods, and client-server architecture.

• **For Data Science Module:** Comfort with Python syntax, data structures, and libraries like Pandas or NumPy

Course Syllabus:

1. Introduction

- Overview of the session goals
- Key expectations and learning outcomes

2. Best Practices

- General best practices across languages
- Efficiency and maintainability considerations
- Leveraging AI-powered development

3. Java Use Cases

- **Boilerplate Code Generation**
  - Automating creation of getters/setters, constructors, `toString()`, `equals()`, and `hashCode()` methods
- **Unit Testing Assistance**

    - Writing JUnit test cases for services, controllers, and utility classes
- **Spring Boot Development**
    - Generating REST controllers, service layers, repository interfaces, and configuration classes
- **API Integration**
    - Writing code to consume REST APIs using `RestTemplate` or `WebClient`

4. .NET Use Cases

- **Code Refactoring and Cleanup**
    - Improving legacy code, suggesting better C# idioms, or simplifying logic
- **ASP.NET Core Web Development**
    - Generating controllers, services, middleware, and routing logic
- **Visual Interpretation UI**
    - Leveraging AI for UI design and enhancements

5. Data Science Use Cases

- **Write Structured, Modular & Object-Oriented Code with Documentation**
    - Designing maintainable object-oriented Python code using reusable functions, modules, and classes
- **Create Test Cases**
    - Generating unit tests to validate data processing functions
- **Code Interpretation – Q&A**
    - Understanding and explaining unfamiliar code snippets using AI-powered assistance (e.g., GitHub Copilot Chat)
- **Data Analysis & Data Science Code**
    - Writing code for EDA, feature engineering, model training, and evaluation

6. Summary & Wrap-up

- Key takeaways from the session
- Q&A and discussion on applying the concepts