# GenAI for Developers

This workshop provides a practical introduction to Generative AI and its application in modern software development. Participants will learn how to work with Large Language Models (LLMs) such as GPT using OpenAI and Azure OpenAI services. The training covers key AI model types, tokenization, and the integration of language models into real-world scenarios.

A major focus is on building custom Copilots and AI Agents. Attendees will get an introduction to Copilot Studio and Azure Foundry and will dive deep into Microsoft Semantic Kernel, an open-source framework for building custom copilots and agents in C# (Python and Java are also supported).

By the end of the session, participants will have a solid understanding of how to build modern AI-enhanced applications in the new Software Era using SK and integrate GenAI into their existing systems.

**1. Introduction**

* Welcome & Training Objectives
* Overview

**2. Generative AI Fundamentals**

* What is Generative AI?
* Overview of GPT models
* Model types: Completion, Chat, Embeddings
* Overview Use cases: Summarization, Classification, Semantic Search
* Tokens and tokenization

**3. Exploring OpenAI & Azure OpenAI**

* Accessing the platforms
* Using the OpenAI Playground
* Using Azure Foundry
* Consuming AI as a REST endpoint

**4. Building AI Agents**

* Agents and Copilots
* Building Agents with Copilot Studio (no-code)
* Building Agents with Azure Foundry (no-code, low-code)

**5. Introduction to Semantic Kernel (SK)**

* What is Semantic Kernel?
* Why use SK?
* Supported services and programming languages
* Architecture of SK
* AI Plugins and their role
* Native Functions
* Semantic Functions

**6. Developing AI Agents and Custom Copilots with SK**

* Initializing SK with different Models
* Creating Native Functions
* Creating Semantic Functions
* Managing conversation history
* Prompt templates and variable injection
* Mixing the natural language with programming language

**10. Q&A and Wrap-Up**

* Discussion
* Recap of key concepts
* Feedback