

Responsible AI in software development (RAIISD)

ID RAIISD Prix sur demande Durée 1 jour

A qui s'adresse cette formation

All people involved in using GenAl or developing machine learning

Pré-requis

General development

Objectifs

- Understand various aspects of responsible AI
- How to use generative AI responsibly in software development
- Prompt engineering for optimal outcomes
- · How to apply generative AI throughout the SDLC

Contenu

A brief history of Artificial Intelligence

- · The origins of AI
- Neural networks and "probability engines"
- Early ML coding tools
- The AI coding revolution of the 2020s
- Threats against ML systems

Responsible Al

- What is responsible AI?
- · Accountability and transparency
- Mitigation of harmful bias
- Validity and reliability
- Validity and reliability non-determinism of the code
- Demonstration Experimenting with validity and reliability in Copilot
- · Explainability and interpretability
- · Safety, security, privacy and resilience
- Security and responsible AI in software development

Using GenAl responsibly in software development

- · LLM code generation basics
- Basic building blocks and concepts

- Prompt templating
- System prompts in Al-driven coding
- · GenAl tools in coding: Copilot, Codeium and others
- Can Al... boost your productivity?
- Can Al... take care of the 'boring parts'?
- · Can Al... be more thorough?
- Reviewing generated code the black box blues
- The danger of hallucinations
- Can Al... teach you how to code (better)?
- Demonstration Experimenting with an unfamiliar API in Copilot
- The effect of GenAl on programming skills
- Some further long-term effects of using GenAl
- Where AI code generation doesn't do well
- Prompt engineering
 - Why is a good prompt so important?
 - · Establishing the context for generative AI
 - Zero-shot, one-shot, and few-shot prompting
 - Reasoning-based prompt engineering, chain-ofthought
 - Demonstration Experimenting with prompts in Copilot
 - · Enforcing and following token limits
 - Prompt patterns
 - Prompt patterns and prompt priming
 - The 6 categories of prompt patterns
 - Prompt pattern: Meta Language Creation
 - Prompt pattern: Persona
 - Prompt pattern: Visualization Generator
 - Prompt pattern: Fact Check List
 - Prompt pattern: Alternative ApproachesPrompt pattern: Refusal Breaker
 - Prompt pattern: Flipped Interaction
 - Prompt pattern: Pripped Interaction
 Prompt pattern: Context Manager
 - Some further prompting approaches
 - Least-to-Most and Self-Planning: decomposition of complex tasks
 - Demonstration Task decomposition with Copilot
 - Prompt engineering techniques for refinement and iteration
 - Unit tests, TDD and GenAl
 - Demonstration Test-based code generation with Copilot

Integrating generative AI into the SDLC

Responsible AI in software development (RAIISD)

- Using GenAl beyond code generation
- Using AI during requirements specification
- Prompt patterns for requirements capturing
- Software design and AI
- Prompt patterns for software design
- Demonstration Requirements capturing and API design with Copilot
- Using Al during implementation
- Prompt patterns for implementation
- Demonstration Finding hidden assumptions with Copilot
- · Using AI during testing and QA
- Using AI during maintenance
- · Prompt patterns for refactoring
- Demonstration Experimenting with code refactoring in Copilot
- Prompt patterns for change request simulation

Security of Al-generated code

- Security of AI generated code
- Practical attacks against code generation tools
- Dependency hallucination via generative AI
- Case study A history of GitHub Copilot weaknesses (up to mid 2024)
- · A sample vulnerability
 - Path traversal
 - Demonstration Path traversal
 - Path traversal-related examples
 - Path traversal best practices
 - Demonstration Path canonicalization
 - Demonstration Experimenting with path traversal in Copilot

Summary and takeaways

- Responsible Al principles in software development
- Generative AI Resources and additional guidance

Responsible AI in software development (RAIISD)





Fast Lane Institute for Knowledge Transfer (Switzerland) AG

Husacherstrasse 3 CH-8304 Wallisellen Tel. +41 44 832 50 80

info@flane.ch, https://www.flane.ch