



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Prompting and Context Engineering for Software Engineers

Class Duration

7 hours of live training delivered over 1-2 days to accommodate your scheduling needs.

Student Prerequisites

- Working experience with at least one programming language (Python, TypeScript, Java, C#, Rust, or equivalent)
- Familiarity with a modern IDE or code editor

Target Audience

Software engineers, tech leads, and platform engineers who want to move beyond simple one-line prompts and learn how to structure context deliberately to get consistent, high-quality outputs from LLMs. Equally relevant for developers building LLM-powered features, teams standardizing prompting practices, and learning and development leaders designing AI upskilling programs.

Description

This hands-on course treats prompt engineering as a first-class software engineering discipline. Participants learn how modern large language models actually process context and why that understanding changes everything about how you write prompts. We work through the mechanics of context windows, token budgets, and attention, then build up a practical framework covering system prompts, user turns, few-shot examples, chain-of-thought patterns, structured output schemas, and tool/function definitions. Labs are conducted against real frontier models and use realistic engineering scenarios: code generation, refactoring, test writing, and documentation.

Learning Outcomes

- Explain how context windows and token budgets shape model behavior and cost.
- Write effective system prompts that reliably steer model behavior across use cases.



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

- Apply few-shot, chain-of-thought, and step-back prompting patterns to complex engineering tasks.
- Define structured output schemas (JSON, TypeScript types) and validate model outputs programmatically.
- Describe tool/function calling and construct tool definitions for coding assistant integrations.
- Identify prompt fragility, prompt injection risk, and mitigation strategies.
- Build a reusable personal prompt library for common engineering tasks.

Training Materials

Comprehensive courseware is distributed online at the start of class. All students receive a downloadable MP4 recording of the training.

Software Requirements

Students need access to at least one frontier model API (Anthropic Claude, OpenAI GPT, or Google Gemini) and a code editor. A free or trial API key is sufficient for the labs.

Training Topics

How LLMs Process Context

- Context windows and token budgets
- Attention, position, and recency effects
- Model tiers and when they matter
- Cost, latency, and throughput tradeoffs

System Prompts and Role Definition

- Anatomy of an effective system prompt
- Persona, format, and constraint layers
- Persistent vs. per-request context
- Team-wide system prompt conventions

Core Prompting Patterns

- Zero-shot, one-shot, and few-shot prompting
- Chain-of-thought and step-back techniques
- Self-consistency and verification loops



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

- Prompt chaining and multi-turn strategies

Structured Outputs and Tool Definitions

- JSON Schema and typed output formats
- Validating and parsing model outputs in code
- Function/tool calling syntax and semantics
- Composing tool pipelines for coding agents

Prompt Engineering for Engineering Tasks

- Code generation with rich context
- Refactoring and code transformation prompts
- Test case and test data generation
- Documentation and comment generation
- Commit message and PR description prompts

Security and Reliability

- Prompt injection and jailbreak patterns
- Defense-in-depth for user-facing prompts
- Evaluating prompt robustness
- Building a reusable prompt library

Workshop

- Hands-on labs: structured output pipeline
- Hands-on labs: tool definition exercise
- Q&A session