Cloud Contraptions
TRAINING & CONSULTING

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

Working with Frontier Coding Models: Claude, GPT, and Gemini for Developers

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
- Basic familiarity with REST APIs is helpful but not required

Target Audience

Software engineers and tech leads who regularly use or plan to use frontier LLMs directly in their workflows or applications. Ideal for developers who want a principled basis for choosing between Claude, GPT, and Gemini rather than defaulting to whichever model they tried first, as well as teams standardizing on a model strategy.

Description

This course takes a practical, model-agnostic approach to the frontier coding models that matter most for software development in 2026. We compare Anthropic Claude (Opus 4.7, Sonnet 4.6, Haiku 4.5 and the 1M-context tiers), OpenAI GPT (GPT-5 and GPT-5.5 family, including the o-series reasoning models), and Google Gemini (3.x family) across capability dimensions relevant to developers: code generation quality, instruction following, context length, reasoning, multimodal support, function/tool calling, structured output reliability, pricing tiers, and latency characteristics. Labs benchmark the same representative engineering tasks across all three families so participants develop their own informed intuitions. We also cover how to access these models via API, SDK, and IDE integrations, and when to reach for a smaller or specialized model instead.

Learning Outcomes

- Compare Claude, GPT, and Gemini across capability dimensions relevant to professional software development.

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

- Evaluate cost, latency, and context-length tradeoffs for different use case shapes.
- Access frontier models via REST API, official SDKs, and common IDE integrations.
- Select the appropriate model tier (nano/mini, standard, extended-context) for a given task.
- Explain when to use specialized or fine-tuned models vs. frontier generalists.
- Apply consistent evaluation criteria to new model releases.

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 free or trial API access to at least two of the three covered model families. Labs use lightweight scripts runnable in Python or TypeScript.

Training Topics

The Frontier Model Landscape

- How frontier models are trained and evaluated
- Benchmark literacy: what MMLU, HumanEval, SWE-bench, and code-agent benchmarks measure
- Model families and versioning: Claude 4.x (Opus 4.7, Sonnet 4.6, Haiku 4.5), GPT-5 / GPT-5.5 / o-series, Gemini 3.x
- Extended-context tiers (1M+ context) and when they matter
- The rise of reasoning/thinking modes and “extended thinking”

Claude for Developers

- Strengths: extended context (1M tier), instruction following, code analysis, agentic tool use
- Claude API: authentication, Messages API, streaming, prompt caching
- Tool use, computer use, and the Claude Agent SDK
- Cost and rate limit profile across Opus, Sonnet, and Haiku tiers

Cloud Contraptions
TRAINING & CONSULTING

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

GPT and OpenAI for Developers

- Strengths: ecosystem maturity, function calling, fine-tuning, the o-series reasoning models
- OpenAI API: chat completions, Responses API, Assistants, streaming
- JSON mode, native structured outputs
- The Codex model family: GPT-5.5 (flagship), the GPT-5.4 family, and the GPT-5.2-Codex specialized variant; reasoning-level selection
- Codex as a multi-surface platform (CLI, IDE extensions, cloud agents) — covered in depth in the dedicated Codex in Practice course
- Cost and rate limit profile

Gemini for Developers

- Strengths: multimodal input, very long context, Workspace integration
- Gemini API via Google AI Studio and Vertex AI
- Code Execution tool and grounding with Search
- Cost and rate limit profile

Model Selection Framework

- Task shape vs. model capability matrix
- Context-length budget planning
- Latency-sensitive vs. batch-workload decisions
- When to use smaller (local or fast) models
- Mixing models in a pipeline

API, SDK, and IDE Integration

- Direct REST API vs. official SDKs
- IDE integrations: VS Code, Visual Studio, JetBrains IDEs, Zed, Cursor, and Windsurf
- Managing API keys and credentials safely
- Observability: logging requests and costs

Workshop

- Side-by-side benchmarking lab
- Model selection scenario exercises
- Q&A session