

Class 13: Efficient Code Generation

- **Review:** System Prompt Engineering
- **Discussion:** AI Token Usage
- **Lecture:** Efficient Token Usage
 - AGENTS.md
 - Model Context Protocol (MCP)
- **Studio:** Homework 4

Copyright 2026, Kyle J. Harms

Review: Model Selection and Energy Consumption

- Larger models generally consume more energy than smaller models ($< 8\text{B}$ parameters).
- Models with higher precision (e.g. 32-bit floating point) generally consume more energy than models with reduced precision (e.g. 16-bit floating point, 8-bit integer, etc.).
- Quantization can also help reduce energy consumption by reducing the number of bits required to represent the model's parameters. (e.g. 4-bit quantization, 8-bit quantization, etc.)

Discussion: Homework 4

Goal: Practice your prompt engineering skills:

- Custom system prompts to generate a response from a local LLM
- Use GitHub Copilot to modify the Vue.js code

Implementation:

- Persona selector (at least 3 personas)
- Output format selector (at least 3 formats)
- ...

Discussion: System Prompt Characteristics

- **Persona:** Defines the model's identity and role (e.g., helpful assistant, expert programmer).
- **Safety & Ethics Focus:** Includes directives to avoid harmful, biased, or inappropriate content.
- **Behavioral Guidance:** Instructs on tone (e.g., polite, neutral), helpfulness, and interaction style.
- **Capability Awareness:** Likely outlines general abilities while acknowledging limitations (e.g., knowledge cutoff).
- **Response Formatting:** May include instructions on how to format responses (e.g., concise, detailed, with examples).

AI Tokens

“Tokenomics”

Discussion: AI Tokens

“[Tokenomics: Why making AI pay is tricky](#)” (BBC, 11 Aug 2026)

- Whenever we communicate with an LLM, we are charged for the number of **tokens** used in the prompt and response.
- Tokens are a unit of text that the model processes and generates.
- Currently, LLM providers operate at a substantial loss, subsidizing the cost of tokens to encourage adoption.
- LLM providers/investors will need to **significantly** increase token costs at some point to make a profit.

Discuss: How should token costs shape how you prototype with AI?

Reducing AI Token Usage

- Use a local LLM to avoid token costs.
- Use a cheaper LLM (e.g., older model, smaller model, lower precision, quantized model, etc.)
- Engineer **efficient prompts** that reduce token consumption.
- Don't use the LLM for tasks where the benefits don't outweigh the costs

Efficient Prompt Engineering

- **Specific prompts:** Clearly define the task and desired output.
- **Descriptive prompts:** Provide context and details to guide the model's response.
- **Contextual prompts:** Include relevant information or background to help the model understand the task. (Avoid overloading context; don't dump the entire codebase into the prompt.)
- **Example-based prompts:** Provide examples of the desired output to guide the model's response.
- **Constraint-based prompts:** Specify any constraints or limitations on the output (e.g., length, format, style, etc.)

LLM Code Generation

Discussion: How do you generate code now?

Are your prompts efficient?

- Are your prompts specific?
- Are your prompts descriptive?
- Do you provide context?
- Do you provide examples?
- Do you provide constraints?

Strategies for Efficient Code Generation

- Better prompt engineering – Always start with a clear, specific, and descriptive prompt.
- Use the right AI tool for the job – use ‘ask’ for quick support (fewer tokens), ‘agent’ for complex tasks (lots of tokens)
- Provide dedicated instructions for LLMs – AGENTS.md
- Provide context and tooling for LLMs – Model Context Protocol (MCP)

AGENTS.md

AGENTS.md

AGENTS.md provides a dedicated set of instructions for the LLM to follow when generating code. It can include:

- Sets the project context for AI agents
- Provides instructions for how to generate code
- Provides instructions for how to format code
- Build/test commands
- etc

Best Practices: AGENTS.md

- Put commands at the top. (e.g. `npm run build`)
- Provide code examples over explanations
- Provide clear boundaries (e.g. “What files can the AI touch and not touch?”)
- Provide specifics about your tech stack. (e.g. “What version of Vue.js?”)

Activity: AGENTS.md

1. Working with a peer, study

<https://github.com/microsoft/mcp/blob/main/AGENTS.md>

2. Evaluate the AGENTS.md file and identify areas for improvement.

3. On your handout, list what you should include in your Homework 4 AGENTS.md file.

Class Activity: AGENTS.md

1. **Ask** GitHub Copilot to generate a component.
2. Provide an `AGENTS.md` file for your Homework 4
3. **Ask** GitHub Copilot to generate a component again in a **new chat**.
4. Compare the two components and evaluate the differences in quality.