

# Computer Science 12: Prompt Engineering Guide

## Effective AI Communication for CS Students

Mr. Gullo

Department of Computer Science

February, 2025

# Overview

# Table of Contents

# Section 1

## Core Principles

# Core Principles

This guide combines principles from Google's 10 Hour Prompting Essentials and Anthropic's prompting framework:

## Five-Step Framework

- 1 **Task:** Clearly define what you want the AI to do
- 2 **Context:** Provide relevant background information
- 3 **References:** Incorporate examples to clarify your needs
- 4 **Evaluate:** Assess the AI's output
- 5 **Iterate:** Refine your prompt based on evaluation

## Key Techniques from Emergent Capabilities

- **Role Setting:** Encourage the AI to take on the characteristics of an expert in the chosen task
- **Chain of Thought Reasoning:** Prompt the LLM to explicitly state its reasoning process step by step before answering to allow for more thorough and well-reasoned responses to complex queries

# Table of Contents

## Section 2

### CS-Specific Examples



# Using References in Your Prompts: Part 1

## Example: Floating Point Precision

### Prompt with Reference:

- "You are a CS tutor helping me understand floating point precision issues. Here's my code that's giving wrong results:

```
float result = 0.1 + 0.2;
```

```
if (result == 0.3)
```

```
    cout << "Equal";
```

```
else
```

```
    cout << "Not equal";
```

Please explain why it prints 'Not equal' and how I should fix it."

### Why This Works Better

- Provides concrete code example showing the problem
- Demonstrates specific behavior that needs explanation
- Gives the AI context about your current understanding

# Using References in Your Prompts: Part 2

## Example: Number System Conversion

### Prompt with Reference:

- "You are a CS tutor helping with number systems. I'm trying to convert between number systems but getting confused. Here are my attempts:

Decimal 42 to binary: 00110010

Hexadecimal 0x2A to decimal: 52

Please explain where I went wrong and show the correct conversions with the steps clearly laid out."

### Why This Works Better

- Shows specific examples of your work
- Highlights misconceptions that need correction
- Asks for step-by-step explanation alongside the answer
- Focuses the AI on educational value, not just giving answers

# Floating Point Example

## Weak vs. Optimized Prompts

### Weak Prompt:

- "Explain floating point numbers."

### Optimized Prompt:

- "You are an expert in teaching computer number systems. Explain the IEEE 754 floating point standard, including the differences between single and double precision. Show the bit layout for each format and walk through an example of how a decimal number is converted to its floating point representation. Include common precision issues that arise in calculations. Act as a CS professor."

## Breakdown

- Task: Explain floating point representation
- Context: IEEE 754 standard with examples
- Role Setting: Act as a CS professor

# Memory Usage Example

## Weak vs. Optimized Prompts

### Weak Prompt:

- "How does computer memory work?"

### Optimized Prompt:

- "You are an expert in teaching computer architecture. Explain memory usage in C++ programs, including the stack vs. heap allocation, memory leaks, and efficient memory management techniques. Include code examples that demonstrate proper and improper memory usage. Discuss how different data structures impact memory efficiency. Think step-by-step."

## Breakdown

- Task: Explain memory usage with practical focus
- Context: C++ programming environment
- Role Setting: Expert in teaching computer architecture

# Number Systems Example

## Weak vs. Optimized Prompts

### Weak Prompt:

- "Explain binary numbers."

### Optimized Prompt:

- "You are an expert in teaching number systems. Explain binary, octal, and hexadecimal number systems, and show the conversion process between them. Include bitwise operations (AND, OR, XOR, shifts) with examples of when they're useful in programming. Demonstrate how these concepts are applied in C++ code examples. Think step-by-step."

# Table of Contents

## Section 3

# Use Cases for CS Students

## Example Prompt

"You are an expert in C++ programming. Explain how to use the C++ math library for solving complex calculations. Show how to implement common mathematical functions like logarithms, trigonometric functions, and exponents. Provide code examples that demonstrate error handling and precision considerations. Think step-by-step about the potential pitfalls students might encounter."



# Input Validation with cin

## Example Prompt

"You are an expert in C++ input/output operations. Demonstrate robust input validation techniques using cin for user input. Show how to handle different types of input (integers, floating-point, strings), deal with buffer issues, and recover from input errors. Include examples of common pitfalls and their solutions. Write code that handles edge cases gracefully. Think step-by-step."

# If/Else and Truth Tables

## Example Prompt

"You are an expert in teaching boolean logic and control flow. Explain the relationship between truth tables and if/else statements in programming. Demonstrate how to implement complex logical conditions using AND, OR, and NOT operators. Show how to simplify boolean expressions and convert them to efficient code. Include examples of common logical errors and debugging techniques. Think step-by-step."

# Table of Contents

## Section 4

# Safety and Ethics

# Academic Integrity and Safety Guidelines

## Academic Integrity

- Do not use AI to complete assignments or exams without understanding the underlying concepts
- Use it as a tool to deepen understanding and check work

## Common Pitfalls

- Avoid oversimplification: AI can provide oversimplified answers that lack nuance
- Always critically evaluate the AI's output
- Consult multiple sources to ensure accuracy

# Table of Contents


## Section 5

# LLM Selection for Computer Science

# Choosing the Right LLM for CS Tasks

## Best Practices

- Match LLM strengths to specific tasks
- Consider using multiple models for different use cases
- Verify outputs against known CS principles
- Use benchmarks as guidelines, not absolute measures



Model	Overall	Overall w/ Style Control	Hard Prompts	Hard Prompts w/ Style Control	Coding	Math	Creative Writing	Instruction Following	Longer Query	Multi-Turn
gemini-2.0-flash-thinking-exp-01-21	1	3	1	1	1	1	1	1	1	1
gemini-2.0-pro-exp-02-05	1	2	1	1	1	2	2	1	1	1
chatgpt-4o-latest-20250129	1	1	2	2	1	6	1	1	1	1
deepseek-r1	4	2	1	1	1	1	3	1	1	1
gemini-2.0-flash-001	4	7	3	8	2	2	3	4	1	2



# Table of Contents

## Section 6

# OpenStax-Gemini Partnership (Deprecated?)

# OpenStax Integration with Google Gemini

## Accessing Trustworthy Educational Content

- OpenStax has partnered with Google to integrate their library with Gemini (August 2024)
- 70+ openly licensed, peer-reviewed textbooks now accessible via Gemini
- Available to Gemini users 18+ in the United States

## How to Access

@OpenStax explain the concept of object-oriented programming?

## Benefits for CS Students

- Accurate, attribution-based responses for CS concepts
- Integration with other Gemini AI capabilities
- Ensures academic integrity while using AI tools

# OpenStax and AI in Education

## Core Principles

- Accuracy in educational content
- Proper attribution to sources
- Accessibility for all learners
- Preservation of academic integrity

## Quote

“OpenStax and Google share a unified, responsible vision for the use of AI in education. We believe content provided through AI learning tools should be accurate and inclusive.”

— Professor Richard G. Baraniuk,  
founder and director of OpenStax

## Impact for CS Education

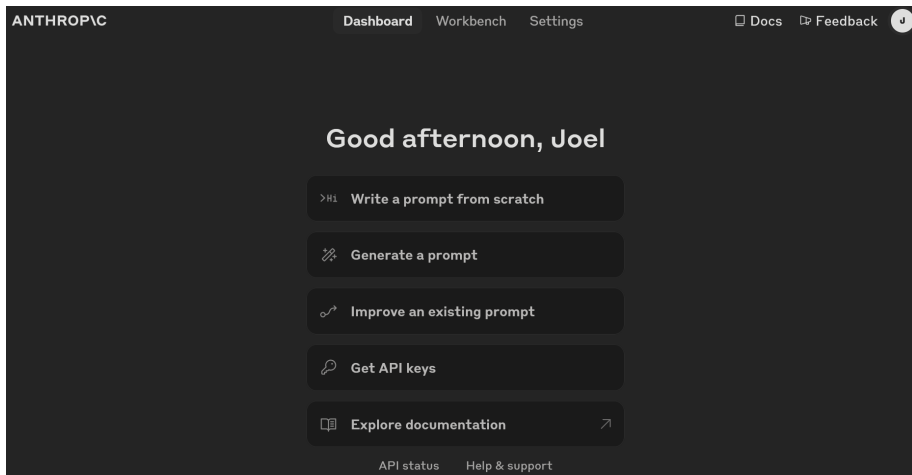
- Access to high-quality CS textbooks through a conversational AI
- Support for challenging programming concepts with trusted content
- Democratizing access to educational resources

# Table of Contents

## Section 7

# Using Anthropic's Console

# Using Anthropic's Console



## ① Prompt Optimizer

- Utilize the prompt optimizer to refine prompts
- Improve performance through suggested modifications

## ② Prompt Generator

- Create production-ready prompt templates
- Describe desired task and output format

## ③ Test Cases

- Generate automatic test cases
- Compare outputs side by side
- Facilitate rapid iteration and refinement



# Table of Contents

## Section 8

# XML Tags in Prompting

# Using XML Tags in Prompts (!Advanced!)

## Purpose of XML Tags

- Structure content in a clear, machine-readable format
- Separate different types of information
- Make prompts more organized and specific

## Common XML Tag Examples

- `<task>Define specific instructions</task>`
- `<context>Provide background information</context>`
- `<example>Show sample content</example>`
- `<output>Specify desired format</output>`

# Table of Contents

## Section 9

# Additional LLM Tools

## Alternative AI Research Assistants

### Perplexity

- Real-time search integration
- Academic paper analysis
- Direct citation capabilities
- Built-in fact-checking

### Google or OpenAI: Deep Research

- Specialized for academic research
- Literature review assistance
- Paper summarization
- Research methodology guidance

## AI-Enhanced Development Environments

- 1 **Windsurf:** Known for its deep integration with project management tools and CI/CD pipelines, offering features like agent mode for hands-free coding sessions and seamless API integration.
- 2 **Cursor:** Offers advanced AI assistance for coding, providing real-time suggestions and code completion.
- 3 **GitHub Copilot:** A widely used AI-powered tool that assists developers by suggesting code and completing tasks based on context.
- 4 **Trae AI:** An IDE that utilizes automated assistance and code generation to enhance software development.
- 5 **Replit:** Evolved into a comprehensive AI-powered development environment with features like real-time collaboration and built-in hosting.

# Summary

- Apply the five-step framework for effective prompts
- Use role setting and chain of thought reasoning
- Create detailed, context-rich prompts for CS problems
- Maintain academic integrity while using AI tools
- Leverage Anthropic's Console for prompt optimization
- Always verify and cross-reference AI-generated content



# Table of Contents

## Section 10

### Sources

## Core Resources

- Google's 10 Hour Prompting Essentials Course  
<https://grow.google/prompting-essentials/>
- OpenStax and Google Gemini Partnership  
<https://openstax.org/blog/press-release-openstax-partners->
- LM Arena Chatbot Leaderboard  
<https://lmarena.ai/?leaderboard>

## Additional Tools & Guides

- Google Gemini for Education  
<https://gemini.google.com/education>
- OpenStax Free Textbooks  
<https://openstax.org>
- MIT Sloan Effective Prompts Guide  
<https://mitsloanedtech.mit.edu/ai/basics/effective-prompts>

# Good afternoon, Joel

>Hi Write a prompt from scratch

Generate a prompt

Improve an existing prompt

Get API keys

Explore documentation



API status

Help & support