



ISDM (INDEPENDENT SKILL DEVELOPMENT MISSION)

◊ CHAIN-OF-THOUGHT PROMPTING FOR STEP-BY-STEP AI RESPONSES

📌 INTRODUCTION

Chain-of-Thought (CoT) prompting is a technique used to **guide AI through a logical reasoning process** by breaking down complex questions into smaller, step-by-step responses. This method enhances the AI's ability to **handle multi-step reasoning, problem-solving, and analytical thinking** more effectively.

Unlike basic prompts that often produce **short and direct answers**, CoT prompting allows AI to **explain intermediate steps**, making responses more structured, detailed, and transparent.

📌 CHAPTER 1: UNDERSTANDING CHAIN-OF-THOUGHT PROMPTING

1.1 What is Chain-of-Thought Prompting?

Chain-of-Thought (CoT) prompting is a structured way of interacting with AI where the model **thinks step by step** before arriving at a final answer. This technique helps AI in **logical reasoning, mathematical problem-solving, and structured decision-making**.

1.2 Why Use Chain-of-Thought Prompting?

- ✓ **Enhances Logical Reasoning** – AI can handle multi-step problems instead of providing a one-step answer.
- ✓ **Breaks Down Complex Tasks** – Helps AI process and solve complicated queries step by step.
- ✓ **Improves Answer Accuracy** – Reduces errors by forcing AI to explain its reasoning.
- ✓ **Ideal for Problem-Solving Tasks** – Useful in math, programming, science, and decision-making scenarios.

📌 CHAPTER 2: HOW CHAIN-OF-THOUGHT PROMPTING WORKS

2.1 Direct Prompt vs. Chain-of-Thought Prompt

A direct prompt simply asks for an answer, whereas a CoT prompt encourages step-by-step reasoning.

📌 Example – Direct Prompt

Prompt: "What is 24×17 ?"

AI Response: "The answer is 408."

📌 Example – Chain-of-Thought Prompt

Prompt: "Solve 24×17 step by step."

AI Response:

"Step 1: Break down the multiplication.

$$24 \times 17 = (24 \times 10) + (24 \times 7)$$

Step 2: Solve each part separately.

$$24 \times 10 = 240$$

$$24 \times 7 = 168$$

Step 3: Add both results.

$$240 + 168 = 408$$

Final Answer: 408"

2.2 Key Features of CoT Prompting

- ✓ Uses **step-by-step explanation** before presenting the final result.
- ✓ Helps AI understand relationships between sub-problems.
- ✓ Ensures logical flow and structured output.

📌 CHAPTER 3: APPLICATIONS OF CHAIN-OF-THOUGHT PROMPTING

3.1 Mathematical Problem Solving

CoT is highly effective for **multi-step calculations** and **word problems**.

📌 Example – Word Problem Using CoT

Prompt: "John has 8 apples. He buys 12 more and gives away 5. How many does he have left?"

AI Response:

"Step 1: Start with 8 apples.

Step 2: Add 12 apples ($8 + 12 = 20$).

Step 3: Subtract 5 apples ($20 - 5 = 15$).

Final Answer: John has 15 apples left."

3.2 Logical Reasoning & Decision Making

CoT prompting is useful for **analyzing different possibilities** before making a decision.

📌 Example – Choosing the Best Investment Option

Prompt: "I have two investment options: A offers 5% annual return, B offers 7% but has more risk. Which should I choose?"

AI Response:

*"Step 1: Identify the potential returns.

Option A: 5% per year (low risk).

Option B: 7% per year (high risk).

Step 2: Consider risk tolerance.

- If I prefer stability, Option A is safer.
- If I can handle risk for higher returns, Option B is better.

Step 3: Evaluate the long-term impact.

- Option A ensures steady returns.
- Option B may bring higher returns but could also lose value.

Final Decision: Choose based on my risk appetite and financial goals."*

3.3 Programming & Debugging

CoT can be used to **explain and debug code errors step by step**.

📌 Example – Debugging a Python Code Error

Prompt: "My Python code has an error: *list index out of range*. What should I do?"

AI Response:

*"Step 1: Understand the error.

- The 'list index out of range' error means that my code is trying to access an index that does not exist in the list.

Step 2: Identify where the issue occurs.

- Check the list length and the index being accessed.

Step 3: Fix the problem.

- Ensure the index is within range by adding a condition:

```
if index < len(my_list):
```

```
    print(my_list[index])
```

```
else:
```

```
print("Index out of range")
```

Final Answer: Check the index before accessing the list to avoid this error.*

❖ CHAPTER 4: BEST PRACTICES FOR USING CHAIN-OF-THOUGHT PROMPTING

4.1 How to Write Effective CoT Prompts

- ✓ **Be specific – Ask AI to show step-by-step reasoning.**
- ✓ **Use role-based instructions – Assign AI a role (e.g., “Act as a math tutor...”).**
- ✓ **Encourage intermediate steps – Use phrases like “Break it down,” “Explain each step,” or “Analyze before answering.”**

❖ Example – Role-Based CoT Prompt

Prompt: *“Act as a teacher and explain how an electric car works step by step.”*

AI Response:

“Step 1: Energy Source – The electric car uses a rechargeable battery as its power source.

Step 2: Power Conversion – The battery supplies electricity to an electric motor.

Step 3: Motion Generation – The motor converts electrical energy into mechanical movement, spinning the wheels.

Step 4: Regenerative Braking – The car recaptures energy during braking, recharging the battery.

Final Summary: An electric car operates by converting electricity into motion using a motor, without requiring gasoline.”

 **CHAPTER 5: EXERCISES & ASSIGNMENTS****5.1 Multiple Choice Questions (MCQs)**

1. What is the main benefit of Chain-of-Thought prompting?

- (a) It speeds up AI responses
- (b) It forces AI to generate long responses
- (c) It improves step-by-step reasoning
- (d) It simplifies AI instructions

2. Which of the following is an example of a CoT prompt?

- (a) "What is $2 + 2$?"
- (b) "Solve $2 + 2$ and explain step by step."
- (c) "Tell me a fun fact about math."
- (d) "List all prime numbers."

3. In which field is CoT prompting most useful?

- (a) Cooking recipes
- (b) Math problem-solving
- (c) Social media posts
- (d) Fast responses

5.2 Practical Assignment **Task 1:** Use CoT prompting to solve a **real-world decision-making problem** (e.g., choosing a career path, financial planning).

📌 **Task 2:** Write a CoT prompt for **explaining how machine learning works step by step** and compare the AI's response with a direct answer.

📌 CHAPTER 6: SUMMARY

- ✓ Chain-of-Thought (CoT) prompting **improves AI's ability to reason through complex problems**.
- ✓ It is **ideal for multi-step calculations, decision-making, and logical reasoning**.
- ✓ Effective CoT prompts **use step-by-step breakdowns, role-based instructions, and structured reasoning**.
- ✓ Best used in **math, programming, problem-solving, and structured analysis**.

ISDM

◊ FEW-SHOT & ZERO-SHOT LEARNING TECHNIQUES IN PROMPTS

📌 INTRODUCTION

In **AI-driven text generation**, particularly with **ChatGPT and other language models**, the way we craft prompts greatly affects the quality of responses. Two powerful techniques used in **prompt engineering** are:

- ☒ **Few-Shot Learning** – Providing examples to guide AI responses.
- ☒ **Zero-Shot Learning** – Asking AI to generate responses without any examples.

These techniques help AI understand context, structure, and **desired output formats**, making it more effective in applications like **content creation, question-answering, and task automation**.

📌 CHAPTER 1: UNDERSTANDING FEW-SHOT & ZERO-SHOT LEARNING

1.1 What is Few-Shot Learning?

Few-Shot Learning is a **prompting technique where the AI is given a few examples before being asked to generate a response**. This approach helps AI understand the expected format, tone, and structure.

📌 Example – Few-Shot Prompt for Writing an Email:

✍ **Prompt:**

"Write a formal email based on these examples:

Example 1:

Subject: Request for Leave

Dear Manager,

I hope this email finds you well. I would like to request leave from [start date] to [end date] due to personal reasons. Please let me know if you need further details. Thank you.

Example 2:

Subject: Request for Meeting

Dear [Recipient's Name],

I would like to schedule a meeting regarding [topic]. Please let me know your availability.

Now, write an email requesting a deadline extension for a project."*

Why It Works?

- ✓ AI follows the **pattern of examples** and generates a response in the same format.
- ✓ Ensures **consistency, correctness, and relevance**.

1.2 What is Zero-Shot Learning?

Zero-Shot Learning is when the AI is **not provided with any examples** but is still expected to generate a response **based on its general understanding**.

Example – Zero-Shot Prompt for Writing an Email:

Prompt:

"Write a formal email requesting a deadline extension for a project."

Why It Works?

- ✓ AI uses **pre-trained knowledge** to infer what a formal email

should look like.

- ✓ May vary in tone and format compared to Few-Shot Learning.
-

📌 CHAPTER 2: COMPARING FEW-SHOT VS. ZERO-SHOT LEARNING

2.1 Key Differences

Feature	Few-Shot Learning	Zero-Shot Learning
Examples Provided?	Yes, a few examples	No examples
AI Understanding?	Learns from provided samples	Uses pre-trained knowledge
Consistency?	More structured and predictable	Can vary in format and tone
Best for?	Generating structured outputs	Exploring new topics creatively

📌 Example:

Task: Write a product description for a smartphone.

- ✓ **Few-Shot Prompt:** Includes 2-3 examples of product descriptions.
 - ✓ **Zero-Shot Prompt:** Just asks AI to write the description directly.
-

📌 CHAPTER 3: WHEN TO USE FEW-SHOT & ZERO-SHOT LEARNING?

3.1 When to Use Few-Shot Learning?

✓ Best when you need **structured, specific, or consistent responses.**

✓ Ideal for:

- Generating **emails, reports, summaries.**
- Maintaining **brand voice** in marketing content.
- **Code generation** in a specific format.

 **Example – Few-Shot for Product Descriptions:**

 **Prompt:**

"Here are two examples of product descriptions. Now write one for a smart TV."

 **AI Response:** Generates a **detailed product description** in the same format.

3.2 When to Use Zero-Shot Learning?

✓ Best when you need **quick, general answers without predefined formats.**

✓ Ideal for:

- **Brainstorming ideas** (e.g., blog topics, story outlines).
- **Quick fact-based responses** (e.g., "Who invented the telephone?").
- **Creative tasks** (e.g., "Write a poem about AI").

 **Example – Zero-Shot for Writing a Poem:**

 **Prompt:**

"Write a short poem about artificial intelligence."

 **AI Response:** AI creates an original poem based on its understanding of poetry.

 **CHAPTER 4: FEW-SHOT & ZERO-SHOT IN AI APPLICATIONS**

4.1 Few-Shot Learning in Business & Marketing

- ✓ Writing consistent brand messaging.
- ✓ Creating customer support responses.
- ✓ Generating email templates.

 **Example:**

 "Here are three examples of professional email responses. Now generate one for a refund request."

 AI provides a structured, professional email response.

4.2 Zero-Shot Learning for AI Assistants

- ✓ Answering general knowledge questions.
- ✓ Generating creative and unique responses.
- ✓ Writing short, to-the-point summaries.

 **Example:**

 "What are the key benefits of AI in education?"

 AI pulls from its training data and creates a factual response.

 **CHAPTER 5: EXERCISES & ASSIGNMENTS**

5.1 Multiple Choice Questions (MCQs)

1. What is Few-Shot Learning?

- (a) Training an AI with large datasets
- (b) Providing a few examples before asking a question 
- (c) Asking AI to generate responses without any context
- (d) Providing only one example

2. Which of these is an example of Zero-Shot Learning?

- (a) Asking AI to write a blog post with two example blogs
- (b) Giving AI three examples of product descriptions before generating one
- (c) Asking AI to write a product description without examples 
- (d) Providing AI with a script format before generating dialogue

3. When should you use Few-Shot Learning?

- (a) When you need structured and consistent responses 
- (b) When you want AI to be creative
- (c) When asking AI for general knowledge
- (d) When you don't have time to write a prompt

5.2 Practical Assignment

❖ **Task 1:** Write two different prompts for the same task:

- One using **Few-Shot Learning**.
- One using **Zero-Shot Learning**.

📌 **Task 2:** Use **ChatGPT** to generate responses for both prompts.

- Compare the **structure, accuracy, and consistency** of the outputs.

📌 CHAPTER 6: SUMMARY

- ✓ **Few-Shot Learning** uses **examples** to guide AI for more structured and **consistent** responses.
- ✓ **Zero-Shot Learning** asks AI to **generate responses without examples**, relying on **pre-trained knowledge**.
- ✓ **Few-Shot is best for** structured writing, code generation, and maintaining consistency.
- ✓ **Zero-Shot is best for** brainstorming, answering questions, and creative writing.
- ✓ **Prompt Engineering techniques** like these **improve AI efficiency** in various applications.

◊ ROLE OF CONTEXT WINDOWS & TOKEN LIMITS IN AI OUTPUTS

📌 INTRODUCTION

When interacting with AI models like ChatGPT, the **context window and token limit** play a crucial role in shaping the **accuracy, coherence, and relevance** of responses. AI models **process text in chunks** (tokens) within a **limited memory window**, meaning they can only "remember" a certain amount of previous input.

Understanding **how AI handles context and token limits** helps users craft **effective prompts, manage long conversations, and optimize AI-generated content**.

📌 CHAPTER 1: UNDERSTANDING CONTEXT WINDOWS IN AI

1.1 What is a Context Window?

A **context window** refers to the **amount of text an AI model can process at once** before losing track of previous information. It determines how much of the past conversation or document the AI can "remember" when generating a response.

📌 Key Facts About Context Windows:

- ✓ Larger context windows improve long-form responses.
- ✓ Limited context means AI forgets older parts of a conversation.
- ✓ Once the limit is reached, AI can no longer recall earlier inputs.

1.2 How AI Uses Context Windows

AI models break down input into smaller text segments, called **tokens**, and process them within the given context window. When new tokens are added, **older information may get pushed out of memory**.

📌 Example:

Imagine an AI with a **context window of 1,000 tokens**:

1. If the conversation has **800 tokens**, the AI remembers everything.
2. If the user adds **300 more tokens**, the **earliest 100 tokens get removed** to stay within the limit.

- ◆ **Key Takeaway:** AI models work best when users provide **concise and relevant prompts** within the available context window.

📌 CHAPTER 2: UNDERSTANDING TOKEN LIMITS IN AI MODELS

2.1 What are Tokens in AI Models?

A **token** is a unit of text that AI processes, which can be:

- ✓ A **word** (e.g., "hello")
- ✓ A **partial word** (e.g., "unhappi" in "unhappiness")
- ✓ A **punctuation mark** (e.g., "?" or ".")

AI models **count text in tokens rather than words**, meaning a 100-word sentence may use **150+ tokens**, depending on complexity.

📌 Token Estimations:

- ✓ "AI is amazing." → **4 tokens**
- ✓ "The quick brown fox jumps over the lazy dog." → **9 tokens**

2.2 How Token Limits Affect AI Responses

AI models have a **maximum token limit per request**. This includes **both input tokens (user prompt) and output tokens (AI response)**.

📌 **Example Token Limits in AI Models:**

AI Model	Context Window (Tokens)
GPT-3.5	~4,096 tokens
GPT-4	~8,192 - 32,000 tokens

If the total **input + output** tokens exceed the model's limit, the AI **truncates older information**, leading to **loss of memory from the conversation**.

📌 **Example:**

1. User Prompt: "*Explain quantum computing in detail.*" (50 tokens)
2. AI Response: *Detailed explanation (300 tokens).*
3. User Adds: "*Now explain how it applies to AI.*" (20 tokens)
4. AI response considers the **last 370 tokens** while processing the next answer.

◆ **Key Takeaway:** To avoid **losing key details**, users should **structure prompts carefully** and provide context concisely.

📌 **CHAPTER 3: MANAGING CONTEXT WINDOWS & TOKEN LIMITS FOR BETTER AI RESPONSES**

3.1 Best Practices to Optimize Context Usage

- ✓ **Use Concise Prompts** – Avoid unnecessary words to maximize available memory.
- ✓ **Summarize Key Information** – Recap previous discussions instead of repeating entire prompts.
- ✓ **Use Systematic Numbering** – Label sections clearly for better AI recall.

✓ **Break Long Tasks into Smaller Parts** – Instead of asking for a 3,000-word essay, request **separate sections**.

📌 **Example:**

- 🚫 "Tell me everything about climate change from causes to effects and solutions in detail." (Inefficient, might exceed token limit)
- ✓ "Explain the causes of climate change in 300 words. Then, provide effects in a separate response." (Optimized for better AI recall)

3.2 Handling Long Conversations

If a conversation **exceeds the context window**, AI forgets earlier messages. To manage this:

- ✓ Reintroduce essential context periodically.
- ✓ Summarize past discussions before continuing.
- ✓ Use numbered sections to organize conversations.

📌 **Example:**

Instead of:

🚫 "What was the topic we discussed earlier?" (AI may not remember)

Use:

✓ "Previously, we discussed climate change causes. Can you now list five solutions?" (Refreshes context)

3.3 Token Economy: Structuring Prompts for Efficiency

To avoid reaching token limits too quickly, follow these strategies:

- ✓ Avoid redundant words.
- ✓ Use short, precise sentences.
- ✓ Break complex questions into smaller steps.

📌 **Example:**

🚫 "Can you please explain, in as much detail as possible, how

machine learning works and provide an example?" (Too long, redundant)

"Explain machine learning briefly with an example." (Concise, token-efficient)

- ◆ **Key Takeaway:** Efficient prompt structuring helps AI retain relevant context longer and improves response quality.

📌 CHAPTER 4: REAL-WORLD APPLICATIONS OF CONTEXT
WINDOWS & TOKEN LIMITS

4.1 AI Chatbots & Virtual Assistants

- ✓ AI chatbots like ChatGPT, Google Bard, and Microsoft Copilot use context windows to recall past interactions.
- ✓ If the conversation is too long, the bot forgets older parts of the chat due to token constraints.

📌 **Example:**

A customer support chatbot answering inquiries remembers the last few questions but may forget early parts of a long conversation.

4.2 AI in Content Writing & Summarization

- ✓ AI content generators optimize token usage to create detailed but concise articles, blogs, and summaries.
- ✓ AI summarization tools extract key points within token limits.

📌 **Example:**

A user requests: "Summarize a 5,000-word document in 300 words." AI processes the text efficiently within token constraints, selecting key information only.

4.3 AI in Programming & Code Generation

- ✓ AI-powered **coding assistants** (like GitHub Copilot) handle **limited lines of code at a time** within the **context window**.
- ✓ AI remembers **recent code snippets** but may forget **earlier functions** if the code is too long.

 **Example:**

A developer asks ChatGPT to improve a Python function, but after multiple requests, AI **forgets earlier parts of the code**. To fix this, the developer **reintroduces past function details manually**.

 **CHAPTER 5: EXERCISES & ASSIGNMENTS**

5.1 Multiple Choice Questions (MCQs)

1. What is a context window in AI?

- (a) The total storage of AI
- (b) The amount of text AI can process at once
- (c) A chatbot interface
- (d) The AI training model

2. What happens when a conversation exceeds the AI's token limit?

- (a) AI forgets older parts of the conversation
- (b) AI stops working
- (c) AI automatically saves previous text
- (d) AI ignores new input

3. Which technique helps optimize AI's memory retention?

- (a) Writing longer, complex prompts
 - (b) Using concise and structured prompts
 - (c) Avoiding numbering in responses
 - (d) Repeating entire past conversations
-

5.2 Practical Assignment

- 📌 **Task 1:** Test ChatGPT's **context retention** by starting a conversation and asking it to recall something from an earlier message after **1,000+ tokens**. Note how AI handles **memory limitations**.
 - 📌 **Task 2:** Write a **300-word blog post** using **two separate prompts** and analyze whether AI remembers details from the first request.
-

📌 CHAPTER 6: SUMMARY

- Context windows limit how much information AI remembers at once.**
- Tokens define the length of input + output text, affecting response quality.**
- Breaking complex queries into smaller, structured prompts improves accuracy.**
- Reintroducing past context ensures AI retains essential details.**

◊ USING PROMPTS FOR AI-ASSISTED CODING, WRITING, AND PROBLEM-SOLVING

📌 INTRODUCTION

AI-powered tools like ChatGPT have revolutionized the way we approach **coding, writing, and problem-solving** by providing intelligent assistance through well-crafted prompts. Whether you are a **programmer, writer, or problem-solver**, knowing how to use AI effectively can **boost productivity, creativity, and accuracy**.

This study material will cover how to **use prompts strategically** for AI-assisted coding, content creation, and analytical problem-solving.

📌 CHAPTER 1: AI-ASSISTED CODING WITH PROMPTS

1.1 How AI Helps in Coding

AI-powered tools can assist in coding by:

- ✓ Generating code snippets based on **specific programming requirements**.
- ✓ Debugging and **fixing errors in existing code**.
- ✓ Explaining **complex programming concepts**.
- ✓ Suggesting **alternative solutions** for improved efficiency.

📌 Example:

- ✗ "Write Python code." (Too vague)
- ✓ "Write a Python function to find the factorial of a number using recursion." (Specific and clear)

1.2 Writing Effective Coding Prompts

- ✓ **Be Specific** – Define the language, function, and logic.
- ✓ **Set Constraints** – Mention input type, expected output, and any limitations.
- ✓ **Request Explanations** – Ask AI to **explain code for better understanding**.

📌 **Example of a Well-Structured Prompt:**

Prompt: "Write a Python script to fetch real-time weather data from an API and display the temperature in Celsius."

AI Output: (Generates structured code with API integration.)

1.3 Debugging with AI-Powered Prompts

- ✓ Paste the **error message** into AI and ask for **troubleshooting steps**.
- ✓ Request **code optimization** for efficiency.
- ✓ Use AI to **explain the bug and provide solutions**.

📌 **Example:**

Prompt: "Fix this Python error: `TypeError: 'int' object is not subscriptable.`"

AI Response: Explains the cause of the error and provides a corrected version of the code.

📌 **CHAPTER 2: AI-ASSISTED WRITING WITH PROMPTS**

2.1 How AI Enhances Writing

- ✓ **Generates content quickly** – Blogs, articles, reports, and emails.
- ✓ **Improves grammar and structure** – AI refines tone and correctness.

- ✓ **Provides creative suggestions** – Helps with brainstorming and storytelling.
- ✓ **Summarizes lengthy documents** – Creates concise overviews.

 **Example:**

Prompt: "Write a professional email requesting a meeting with a potential client about a business proposal."

AI Output: (Generates a well-structured email with a polite and professional tone.)

2.2 Crafting Prompts for Better Writing Outputs

- ✓ **Set the Purpose** – Clearly define the objective (blog post, email, story).
- ✓ **Specify Tone and Style** – Formal, conversational, persuasive, creative.
- ✓ **Define Length Constraints** – Short summary, detailed explanation, or full-length content.

 **Example:**

 "Write about artificial intelligence." (Too broad)

 "Write a 500-word article on the benefits of AI in healthcare, using a formal tone." (Precise)

2.3 Improving & Editing Writing with AI

- ✓ Request grammar and tone enhancements.
- ✓ Use AI for rephrasing unclear sentences.
- ✓ Ask AI to simplify technical content for easier reading.

 **Example:**

Prompt: "Rewrite the following paragraph in simpler language while maintaining the meaning."

AI Response: (*Provides a clearer and more concise version of the paragraph.*)

📌 CHAPTER 3: AI-ASSISTED PROBLEM-SOLVING WITH PROMPTS

3.1 How AI Helps in Problem-Solving

- ✓ **Breaks down complex problems** into simpler steps.
- ✓ **Suggests multiple approaches** for finding solutions.
- ✓ **Analyzes data and trends** to assist in decision-making.
- ✓ **Offers creative solutions** for brainstorming sessions.

📌 Example:

Prompt: "What are some innovative solutions for reducing urban traffic congestion?"

AI Response: (*Suggests smart traffic lights, carpooling incentives, AI-powered navigation, etc.*)

3.2 Structuring Effective Problem-Solving Prompts

- ✓ **Define the Problem Clearly** – What needs to be solved?
- ✓ **Request Step-by-Step Solutions** – Ask for breakdowns in logical steps.
- ✓ **Ask for Alternative Approaches** – Compare different problem-solving methods.

📌 Example:

✗ "Solve climate change." (Too broad)

✓ "List five practical solutions to reduce carbon emissions in urban areas." (Specific)

3.3 AI in Decision-Making & Critical Thinking

- ✓ AI can **analyze multiple perspectives** before making recommendations.
- ✓ Helps in **evaluating pros and cons** of different solutions.
- ✓ **Generates predictive insights** based on past trends.

📌 **Example:**

Prompt: "Compare the advantages and disadvantages of solar and wind energy for residential use."

AI Response: (Provides a structured comparison with key pros and cons.)

📌 **CHAPTER 4: BEST PRACTICES FOR AI-ASSISTED PROMPTING**

4.1 Refining Prompts for Better Results

- ✓ Use **context and examples** to guide AI responses.
- ✓ Set **clear expectations** for length, depth, and format.
- ✓ Experiment with **rewording** prompts to improve results.

📌 **Example:**

- ✗ "Explain blockchain." (Basic)
- ✓ "Explain blockchain technology in 200 words with real-world applications." (Well-structured)

4.2 Avoiding AI Limitations

- ✓ Always **verify** AI-generated facts.
- ✓ Use AI as a **tool for assistance**, not a replacement for human judgment.
- ✓ Request **multiple perspectives** to reduce **bias** in responses.

📌 **Example:**

- ✗ "Is remote work better than office work?" (Leads to biased

responses)

- "Compare the benefits and challenges of remote work and office work." (Balanced)
-

CHAPTER 5: EXERCISES & ASSIGNMENTS

5.1 Multiple Choice Questions (MCQs)

- 1. What is a key benefit of using AI for coding?**
 - (a) AI writes completely error-free code
 - (b) AI replaces programmers completely
 - (c) AI assists in debugging and generating code snippets
 - (d) AI can write code without user input

- 2. What is important when crafting AI prompts for writing?**
 - (a) Making prompts vague
 - (b) Defining tone, style, and word limit
 - (c) Asking AI to guess the topic
 - (d) Providing no structure

- 3. How does AI assist in problem-solving?**
 - (a) By solving all problems instantly
 - (b) By analyzing data, suggesting approaches, and breaking down solutions
 - (c) By providing only one fixed answer

- o (d) By making decisions without human input
-

5.2 Practical Assignment

📌 **Task 1:** Use AI to generate a short blog on "**The future of AI in business.**" Experiment with different prompt variations and compare results.

📌 **Task 2:** Write a prompt asking AI to debug a given Python error message and provide an optimized solution.

📌 CHAPTER 6: SUMMARY

- ✓ AI-assisted coding helps in **code generation, debugging, and explanations.**
- ✓ AI-powered writing enhances **content creation, editing, and structuring.**
- ✓ AI supports **problem-solving by analyzing multiple solutions and perspectives.**
- ✓ **Well-crafted prompts** improve AI-generated outputs, making responses more accurate and useful.



ASSIGNMENT:

DEVELOP A STRUCTURED MULTI-STEP PROMPT TO GUIDE AI THROUGH SOLVING A COMPLEX TASK LIKE SUMMARIZING RESEARCH PAPERS OR GENERATING CODE SNIPPETS.

ISDM-Nxt



ASSIGNMENT SOLUTION

DEVELOP A STRUCTURED MULTI-STEP PROMPT TO GUIDE AI THROUGH A COMPLEX TASK

Objective

The goal of this assignment is to learn how to **create structured multi-step prompts** that guide AI to solve complex tasks effectively. By following this step-by-step guide, you will develop a **well-organized prompt** to help AI **summarize research papers** or **generate structured code snippets** with accuracy and depth.

Step 1: Understanding Multi-Step Prompting

1.1 What is a Multi-Step Prompt?

A multi-step prompt is a structured prompt that **guides AI step by step** through a complex task instead of requesting a one-step response. This approach is especially useful for tasks that require **detailed processing, logical breakdown, or structured output**.

1.2 Benefits of Multi-Step Prompting

- ✓ **Ensures accuracy** – AI follows a structured approach to problem-solving.
- ✓ **Breaks down complex tasks** – Helps AI generate a **logical sequence** of responses.
- ✓ **Enhances control over AI output** – Directs AI **towards specific formats and details**.

❖ Step 2: Choose a Complex Task

For this assignment, we will create multi-step prompts for **two real-world tasks**:

1. **Summarizing research papers** – AI must extract key points, findings, and conclusions.
 2. **Generating structured code snippets** – AI must create readable and functional code with explanations.
-

❖ Step 3: Develop a Multi-Step Prompt for Summarizing Research Papers

3.1 Define the Goal

We want AI to **summarize a research paper in a structured manner**, highlighting key points like the **title, author, research objective, methodology, key findings, and conclusion**.

3.2 Create the Multi-Step Prompt

❖ Multi-Step Prompt:

"I have a research paper that needs to be summarized. Follow these steps for a well-structured summary:"

1. Extract the Paper's Title & Authors

- Identify the full title of the research paper.
- Mention the names of the authors.

2. Summarize the Research Objective

- Explain in one or two sentences what the research aims to accomplish.

3. Outline the Methodology

- Describe the approach, experiments, or data collection methods used.

4. List the Key Findings

- Extract 3 to 5 important results or discoveries from the research.

5. Provide the Conclusion

- Summarize the overall conclusion and implications of the research.

6. Generate a Summary in One Paragraph

- Combine all points into a clear, concise summary of the research paper.

3.3 Expected AI Response

📌 Example Output for a Research Paper on AI in Healthcare

Title & Authors: "The Impact of AI in Medical Diagnosis" by Dr. John Smith & Dr. Sarah Lee

Research Objective: This research aims to analyze how AI can improve diagnostic accuracy in medical fields.

Methodology: The study uses machine learning models trained on medical imaging datasets to detect diseases like cancer.

Key Findings:

1. AI models detected diseases with **92% accuracy** compared to **85%** from human doctors.
2. AI-assisted diagnoses **reduced patient wait times by 40%**.
3. Deep learning algorithms showed **higher sensitivity in early-stage cancer detection**.

Conclusion: *The study concludes that AI can enhance diagnostic precision, reduce errors, and improve healthcare accessibility.*

Final Summary: *The research paper titled "The Impact of AI in Medical Diagnosis" by Dr. John Smith & Dr. Sarah Lee examines AI's role in improving medical diagnosis accuracy. The study utilized machine learning models trained on medical imaging data and found that AI-assisted diagnoses performed better than human doctors, with a 92% accuracy rate. The research concludes that AI technology can significantly enhance healthcare systems worldwide.*

3.4 Analysis of the Multi-Step Prompt

- ✓ Structured response with clear sections (Title, Objective, Findings, etc.).
- ✓ Easy to follow and useful for academic purposes.
- ✓ Improves AI's ability to provide in-depth, coherent summaries.

❖ Step 4: Develop a Multi-Step Prompt for Generating Structured Code Snippets

4.1 Define the Goal

We want AI to **generate a complete code snippet** for a given task while providing explanations for each step.

4.2 Create the Multi-Step Prompt

📌 Multi-Step Prompt:

"Write a Python program that calculates the factorial of a number.
Follow these structured steps to ensure clarity:"

1. Explain the Purpose

- Briefly explain what the program does.

2. Write the Function Definition

- Define the function to calculate the factorial.
- Use recursion or iteration.

3. Provide an Example Input & Output

- Show a sample execution with an input number.

4. Explain the Code Step by Step

- Break down how the function works.

5. Suggest Possible Enhancements

- Mention improvements, such as handling invalid inputs.

4.3 Expected AI Response

📌 Example Output for Factorial Calculation

Purpose:

"This program calculates the factorial of a given number using recursion."

Code Snippet:

```
def factorial(n):  
  
    if n == 0 or n == 1:  
        return 1  
    else:  
        return n * factorial(n - 1)
```

```
    return 1  
  
else:  
  
    return n * factorial(n - 1)
```

num = 5

```
print("Factorial of", num, "is", factorial(num))
```

Example Execution:

- 📌 *Input:* num = 5
- 📌 *Output:* Factorial of 5 is 120

Step-by-Step Explanation:

1. The function `factorial(n)` takes an integer `n` as input.
2. If `n` is **0 or 1**, it returns **1** (base case).
3. Otherwise, it calculates the factorial recursively using `n * factorial(n - 1)`.
4. The program calls the function with `num = 5` and prints the result.

Possible Enhancements:

- ✓ Add **input validation** to ensure `n` is a positive integer.
- ✓ Use an **iterative approach** instead of recursion for better performance.
- ✓ Allow **user input** instead of a fixed number.

4.4 Analysis of the Multi-Step Prompt

- ✓ Ensures clarity and structured output.
- ✓ Encourages AI to explain the code rather than just generating

it.

- ✓ Helps learners understand and modify the code easily.
-

❖ Step 5: Compare and Analyze the Prompts

Complex Task	Prompt Type	AI Response Structure
Summarizing Research Papers	Multi-step academic prompt	Organized summary with sections
Generating Code Snippets	Step-by-step structured prompt	Code + Explanation + Enhancements

📌 Key Insights:

- Multi-step prompts lead to well-structured AI responses.
 - AI follows a logical order when given step-by-step instructions.
 - Breaking down complex tasks improves accuracy and readability.
-

📌 Step 6: Final Assignment Submission

What to Submit?

- 📌 Prepare a document (Word/PDF) that includes:
- Your structured multi-step prompts.
 - AI-generated responses.

- Your analysis of how multi-step prompts improve AI-generated outputs.

Optional Presentation (If Required)

❖ Create 3-5 slides explaining:

- The importance of structured prompting.
- Comparison between basic and multi-step prompts.
- Real-world applications of multi-step prompting.

CONCLUSION

- ✓ Multi-step prompting helps AI generate structured, well-organized responses.
- ✓ It is useful for complex tasks like research summarization and code generation.
- ✓ Breaking down prompts into logical steps improves response quality.
- ✓ AI performs better when given clear, sequential instructions.