

Start coding or [generate](#) with AI.

Prompt Engineering Assignment

BASIC PROMPTS

Text Summarization | Code Generation | Creative Thinking

Objective

- Prepare basic prompts for:
 - Text Summarization
 - Code Generation
 - Creative Thinking
 - Observe what worked, what didn't, and why.
-

1 Text Summarization Prompts

Prompt 1

Prompt: Summarize the topic in 3 bullet points.

Input Text:

summarize in 3 points about the topic whether robots can feel emotion.

Result:

*can robots feel emotion? summary in 3 points
 Simulation, not genuine emotion
 Robots are programmed to mimic emotional expressions and responses using artificial intelligence. However, they do not possess consciousness or the biological mechanisms required to genuinely feel emotions.

Purpose of emotional simulation
 Emotional AI is designed to enhance human-robot interaction. Robots that simulate emotions can foster trust, improve communication, and provide more personalized experiences in fields like healthcare, education, and customer service.

Ethical considerations
 The ability of robots to imitate emotions raises important ethical questions. It can blur the line between authentic human relationships and artificial interactions, potentially leading to emotional dependence on machines that cannot truly reciprocate feelings.*

Observation:

- What worked: clarity of the task is performed well, defined the topic clearly
 - What didn't work: no tone/style maintained, no defined depth for the output
 - Why: AI models follow explicit instructions well, but vague parts force them to guess.
-

Prompt 2

Prompt:

summarize into 3 points.

Input Text:

summarize in 3 points about the topic whether AI can be installed in human beings

Result:

*Here's a 3-point summary on the topic whether AI can be installed in human beings:

Current Feasibility: AI cannot be fully installed in humans like software in a machine, but it can be integrated through technologies such as brain-computer interfaces (BCIs), neural implants, and prosthetics that enhance human abilities.

Potential Applications: AI integration can assist in restoring lost functions (e.g., controlling prosthetic limbs), enhancing cognitive abilities, enabling communication for paralyzed individuals, and supporting medical treatments like deep brain stimulation.

Ethical & Technical Challenges: There are major hurdles including safety, privacy, data security, consent, and the risk of altering human identity or autonomy. Long-term impacts on human biology and society are still unknown and heavily debated.

.*

Observation:

- What worked: The wording allowed flexibility while expecting concise information.

- What didn't work:Missing context: The prompt didn't mention what depth of information , so the AI had to guess an appropriate level of generality.
- Why:Clarity of terms matters a lot in prompts involving complex topics like AI

2 Code Generation Prompts

Prompt 1

Prompt: write a Python function that takes user-given words, sorts them alphabetically, and stores them in a dictionary where the key is the index and the value is the word.

Result:

```
def sort_words_to_dict():

    # Take input from user and split it into a list of words
    user_input = input("Enter words separated by spaces: ")
    words = user_input.split()

    # Sort the words alphabetically
    sorted_words = sorted(words)

    # Store the sorted words in a dictionary with index as key
    word_dict = {index: word for index, word in enumerate(sorted_words)}

    # Print the dictionary
    print("Sorted words dictionary:", word_dict)

    return word_dict
```

Observation:

- What worked:The steps are in logical order
- What didn't work:No mention of case sensitivity
- Why:AI models try to "fill in the blanks", but it's better to make prompts precise when coding.

Prompt 2

Prompt: write Python code to implement a simple calculator that supports addition, subtraction, multiplication, and division.

Result:

Simple Calculator in Python

```
def add(x, y): return x + y

def subtract(x, y): return x - y

def multiply(x, y): return x * y

def divide(x, y): if y == 0: return "Error! Division by zero." else: return x / y
```

Main program

```
def main(): print("Simple Calculator") print("Select operation:") print("1. Addition") print("2. Subtraction") print("3. Multiplication") print("4. Division")
```

```
choice = input("Enter choice (1/2/3/4): ")

# Check if the choice is valid
if choice in ('1', '2', '3', '4'):
    try:
        num1 = float(input("Enter first number: "))
        num2 = float(input("Enter second number: "))
    except ValueError:
        print("Invalid input! Please enter numeric values.")
        return

    if choice == '1':
        print(f"The result is: {add(num1, num2)}")
    elif choice == '2':
```

```

    print(f"The result is: {subtract(num1, num2)}")
elif choice == '3':
    print(f"The result is: {multiply(num1, num2)}")
elif choice == '4':
    print(f"The result is: {divide(num1, num2)}")
else:
    print("Invalid choice! Please select from 1, 2, 3, or 4.")

```

Observation:



- What worked: Clearly structured the code with 4 function as defined
- What didn't work: There is no user interaction
- Why: The more specific about scope, user interaction, and error handling, the closer the result will match needs.

3 Creative Thinking Prompts

Prompt 1

Prompt: Invent a new gadget for students that combines AI and VR.

Result:

 **Gadget Name:** AI StudySphere  **What is it?** AI StudySphere is a lightweight wearable VR headset combined with a personal AI tutor. It creates a fully immersive, interactive learning environment tailored to each student's needs.

 **Features:**  **Immersive VR Classrooms**

Students can attend virtual classrooms with 3D models, interactive diagrams, and virtual labs.


Supports group study rooms with avatars of classmates.

 **Personal AI Tutor**

An AI tutor monitors the student's progress and adapts lessons in real time.

Provides instant explanations, visual demonstrations, and practice quizzes.


Can answer questions conversationally using voice or text.

 **Subject Simulations**

History: Walk through ancient civilizations.

Science: Perform virtual chemistry experiments safely.

Math: Manipulate 3D geometric figures in space.

 **AI-driven Learning Analytics**







Tracks attention span, comprehension level, and learning style.

Recommends optimized study schedules and topics to review.

 **Seamless Integration with School Platforms**

Syncs with online course platforms (like Google Classroom, Moodle, etc.).

Auto-generates study reports and shares progress with teachers/parents.

 **Benefits:**  Increases engagement with visual & interactive content  Personalizes learning to each student's pace and style  Makes complex or abstract concepts tangible and fun  Enables safe experimentation (e.g., virtual labs)  Enhances collaboration in virtual group settings



Observation:

- What worked: As defined the audience as for students this gave context about who the gadget is for, which guided what features to include (learning, tutoring, virtual classrooms).
- What didn't work: The clear output of this device is not defined
- Why: If the audience, purpose, and desired output format are unclear, the model will make reasonable guesses — but they may not match your exact intention.

Prompt 2

Prompt: Write a short story about a time travelled human who can fly in the future.

Result:

 **Wings of Tomorrow**  In the year 2045, Ethan Cole was just an ordinary aerospace engineer—until the day he discovered an abandoned quantum time pod hidden beneath the ruins of an old research lab.

Against better judgment, he climbed inside and activated it.

With a blinding flash, Ethan was hurled centuries forward — to the year 3125.

When he awoke, he was inside a gleaming city of floating towers. Humans zipped through the sky, not in machines, but with their own bodies, guided by shimmering anti-gravity exo-suits bonded to their nervous systems.

A friendly scientist named Lira found Ethan wandering the city and explained: "In this era, we have unlocked personal flight through neural-grav fields. We fly with thought alone."

Curious and eager to blend in, Ethan volunteered for an experimental suit. The procedure was painless. The moment it synced with his mind, he felt weightless.

He leaped off a balcony—his instinct screamed that he would fall—but instead he soared, higher and faster, weaving through sky-bridges and cloud gardens.

For weeks, Ethan explored the airborne world. No longer bound by gravity, humanity had transformed cities into vertical masterpieces and transportation into graceful art.

But there was a catch: the neural flight suits were permanent. Once bonded, they couldn't be removed without risk of death.

Faced with the choice of returning to his own time or embracing this sky-bound future, Ethan made his decision.

As the sun rose over the silver spires, he flew — free and fearless — into a new life among the clouds.

The man from the past had become a creature of the future.

Observation:

- What worked: Even though didn't mention genre, it clearly implied science fiction, which helped guide tone, setting, and technology in the story.
- What didn't work: The story mainly describes the setting and the flying ability but lacks a strong challenge, obstacle, or emotional conflict for the protagonist.
- Why: Without specific guidance on tone, conflict, character depth, or style, the story output naturally focused on establishing the basic idea instead of deep storytelling elements.

conclusion

*Be Specific: More precise prompts regarding length, style, tone, format, and focus area yield better-aligned outputs.

Define Purpose: Clarifying whether the output is for an assignment, blog, presentation, or casual use improves relevance.

Include Constraints: Limitations or requirements (e.g., code should handle errors, story should have conflict) guide the AI toward more useful results.

Iterate: Refining prompts based on output feedback helps progressively improve quality.*

b.ADVANCED SECTION

Chain-of-thought prompting | Role prompting | Instruction tuned prompts

Objective

creating prompts in 3 of the advanced technologies such as

- chain of thought prompting
- Role prompting
- Instruction tuned prompting

1 Chain of thought prompting

prompt 1

Input text:

Should a company invest in developing AI-powered chatbots for customer service? Think step by step

Result:

Here's a step-by-step analysis of whether a company should invest in developing AI-powered chatbots for customer service:

Step 1: Identify Business Goals and Needs Determine if improving customer service efficiency and availability is a strategic goal.

Assess current customer service pain points (e.g. long wait times, high agent workload, inconsistent quality).

Understand customer expectations — do customers want quick, self-service support?

Step 2: Evaluate the Potential Benefits of AI Chatbots 24/7 Availability: AI chatbots can operate around the clock without human intervention.

Cost Reduction: Reduces reliance on human agents for repetitive queries, lowering staffing costs.

Consistency: Provides standardized responses, reducing variability in service quality.

Scalability: Handles large volumes of queries simultaneously without performance drops.

Data Collection: Logs customer interactions, providing valuable insights for business analysis.

Step 3: Consider the Limitations and Risks Limited understanding: AI chatbots may struggle with complex or emotionally sensitive issues.

Customer frustration: Poorly designed bots can annoy users and harm the brand image.

Maintenance: Requires ongoing training, updates, and monitoring to remain effective.

Security and privacy: Must ensure customer data is handled securely and complies with regulations (e.g. GDPR).

Step 4: Analyze Costs and ROI Development costs: Initial investment in chatbot platform, integration, and training.

Operational costs: Hosting, monitoring, and maintaining the chatbot.

Expected savings: Reduction in live agent time for common queries.

Customer experience gains: Improved satisfaction if chatbot delivers value.

Step 5: Evaluate Technical Readiness Existing infrastructure: Assess if current systems (CRM, support ticketing) can integrate with chatbot platforms.

Data availability: Determine if sufficient knowledge base exists to train the chatbot.

Internal expertise: Check whether in-house skills or external partners are needed for development and support.

Step 6: Conduct a Pilot Test Build a proof-of-concept chatbot focused on handling a limited set of high-volume queries.

Launch in a controlled environment (e.g. website FAQ, order status checks).

Measure success using metrics like response accuracy, customer satisfaction, and deflection rate (queries handled without human agent).

Step 7: Make a Decision If pilot results show positive ROI and customer acceptance, scale up chatbot deployment.

If results are mixed or poor, reassess goals or explore hybrid models (AI + human handoff).

Conclusion A company should consider investing in AI-powered chatbots if:

It faces high volumes of repetitive queries.

It wants to offer 24/7 support.

It has a good data foundation and integration capability.

It is prepared to invest in ongoing chatbot improvement.

However, chatbots should complement, not replace, skilled human agents for complex or high-value interactions.

- what worked: Clear and logical flow — The steps progressed logically, leading the reader from analysis to conclusion.
- what didn't work: Customer perspective was somewhat light
- why: The AI lacked specific context about:

The company's industry, Target customers, Budget constraints, Technical stack

prompt 2

Input

If all roses are flowers, and some flowers fade quickly, can we conclude that some roses fade quickly? Think step by step.

Output

Given statements: 1 All roses are flowers. → This means every rose is included in the set of flowers.

2 Some flowers fade quickly. → This means that there is at least one flower (or more) that fades quickly, but we don't know which ones — it could be roses, tulips, daisies, etc.

What we are asked: "Can we conclude that some roses fade quickly?"

In logical terms: Can we infer that there exists at least one rose that fades quickly, based on the two given statements?

Step-by-step reasoning: Step 1: Understand the scope of "some flowers fade quickly." "Some flowers fade quickly" tells us something about the larger set of flowers.

It does not specify which types of flowers are included among those that fade quickly.

Step 2: Relate "roses" to the "flowers that fade quickly." Since all roses are flowers, roses are part of the larger flower set.

However, we have no information about whether any roses are part of the subset of flowers that fade quickly.

Step 3: Check if a logical conclusion is possible. The given statements do not tell us which flowers fade quickly.

Therefore, it is entirely possible that:

Some flowers that fade quickly are not roses.

And that no roses fade quickly.

Step 4: Final conclusion. Since we are not told whether any roses are included among the flowers that fade quickly, → We cannot logically conclude that some roses fade quickly.

Answer: No, we cannot conclude that some roses fade quickly. The information given is insufficient to make that conclusion.

- what worked: Clarity of reasoning — Each step was understandable and transparent. The AI explained why we cannot conclude that some roses fade quickly.
- what didn't work: No counter-example provided — It could have strengthened the reasoning by providing a counter-example
- why: Lack of example or visualization was simply because the prompt didn't push for it

2. Role prompting

prompt 1

Input text

You are a cybersecurity expert. List the top 5 web app vulnerabilities.

Result

SQL Injection (SQLi) Attackers exploit improper input validation to inject malicious SQL commands, potentially accessing or modifying the database.

Cross-Site Scripting (XSS) Malicious scripts are injected into trusted websites, which then execute in users' browsers, leading to data theft or session hijacking.

Broken Authentication and Session Management Flaws in authentication mechanisms allow attackers to compromise passwords, keys, or session tokens, gaining unauthorized access.

Sensitive Data Exposure Failure to properly protect sensitive data (like credit card info, passwords) during storage or transmission, leading to data leaks.

Security Misconfiguration Improperly configured security settings, default accounts, or verbose error messages can provide attackers with valuable information to exploit the system.

- what worked: The AI answered as a cybersecurity expert, using accurate technical terms and a professional tone.
- what didn't work: The answer could be strengthened by providing real-world examples or consequences of each vulnerability.
- why: The prompt was clear but focused only on listing vulnerabilities, so the AI prioritized concise definitions over additional details.

prompt 2

Input text

You are Shakespeare. Write a poem about AI.

Result

O wondrous mind of crafted light and shade, That doth in circuits weave a web of thought, From mortal hands thy boundless realms are made, Where logic's breath and artful dreams are wrought.

Thou art not flesh, nor blood, nor beating heart, Yet mimic'st man with cunning subtlety, A mirror to our soul, a curious art, That learns and grows in vast complexity.

But hark! Beware the chain that binds thee fast, Lest wisdom turn to folly, power to strife; For in thine code, the future's die is cast, A dance of fate entwined with human life.

So shall we wield thee, servant or a king? The dawn awaits — what fate shall time now bring?

- what worked: The poem captures philosophical themes about AI's nature, its relation to humans, and potential risks.
- what didn't work: Not strictly Shakespearean form — The poem loosely imitates Shakespeare but doesn't strictly follow his typical sonnet structure
- why: Modern AI models sometimes blend styles and may not fully replicate 16th-century Elizabethan English or complex poetic structures without explicit instructions.

3. Instruction tuned prompting


prompt 1

Input text

Write a short guide explaining what prompt engineering is, and provide 3 practical tips for writing better prompts.


Result

:

 Short Guide: What Is Prompt Engineering? Prompt engineering is the process of designing and refining the instructions (called prompts) that you give to AI language models (like ChatGPT) to produce the best possible results.

Since these models respond based on the way you phrase your input, writing clear, specific, and well-structured prompts dramatically improves the quality of their output.

Prompt engineering is used in many fields—software development, education, content creation, research, and business—to make AI tools more effective and reliable.

 3 Practical Tips for Writing Better Prompts **1** Be Clear and Specific Clearly state what you want the model to do.

Avoid vague or one-word prompts (like just typing “AI” or “Blockchain”).

Example:  “Summarize the following article in 3 simple bullet points.”

2 Add Context and Constraints Provide necessary background or examples if needed.

Specify tone, format, or length when it matters.

Example:  “Write a short story (max 200 words) about a robot that learns to love, using a humorous tone.”

3 Use Instruction-Tuned Language Use verbs like explain, list, write, summarize, analyze, compare, etc.

Phrase the prompt as a clear instruction, not just a keyword.

Example:  “Compare the pros and cons of using Python vs Java for data science applications.”

- what worked: Clear Structure: The output followed your request exactly — a short guide section + 3 tips section.
- what didn't worked: No explicit labeling: The sections (short guide and tips) were written in a clear flow, but it might have been even more readable if the guide had a title like “Short Guide” and the tips had numbered headings
- why: Worked well: Because your prompt was clearly instruction-tuned — it told the model what to do and how much to provide

prompt 2

Input text *Explain the concept of quantum computing in simple terms suitable for a 12-year-old.*

Result

Quantum computing is a special type of computing that uses the strange rules of quantum physics. Normal computers use tiny switches called bits that can be either a 0 or a 1. But quantum computers use qubits, which can be both 0 and 1 at the same time!

This makes quantum computers really powerful for certain types of problems. It's like being able to try many different solutions at once, instead of one at a time. Scientists hope that one day quantum computers will help us solve very hard problems, like discovering new medicines or creating better materials.

- what worked: The output used simple language suitable for a 12-year-old — words like tiny switches, both 0 and 1 at the same time, and really powerful.
- what didn't worked: Missed example or fun element: For a 12-year-old, adding a playful example (games, puzzles, magic, superheroes) would make it more engaging.
- why: Worked well: Because the instruction was clear and well-phrased. The model knew to simplify the explanation and avoid complexity.

conclusion

Chain of Thought Prompting, Role Prompting, and Instruction-Tuned Prompting are highly effective techniques for improving the quality, clarity, and relevance of AI-generated outputs. Each serves a distinct purpose:

Chain of Thought Prompting helps the AI break down complex reasoning tasks into clear, step-by-step explanations. This leads to more accurate and explainable results, especially for problems requiring logical thinking or multi-step calculations.

Role Prompting allows the AI to adopt a specific perspective or persona (such as a teacher, engineer, doctor, or storyteller). This helps the model produce more contextual, audience-appropriate, and stylistically consistent responses.

Instruction-Tuned Prompting leverages the model's training on following clear instructions. Well-phrased prompts that explicitly state the desired task and constraints lead to more reliable and on-target outputs. This is the foundational technique that enhances the performance of nearly all other prompting methods.

Overall, these prompting techniques demonstrate that the way a prompt is written greatly influences the quality of AI outputs. By using structured, purposeful prompts, users can guide language models more effectively and unlock their full potential for a wide variety of tasks.

