AI Tools and Applications Lab

CONTENTS

S.No	Contents
1.	**Introduction to AI and the AI Tools Landscape** What is AI, Machine Learning, and Generative AI — basic definitions and how they relate to one another. Core categories of AI tools: conversational/text, code assistants, data-analysis tools, image-generation tools, and writing/productivity tools. How large language models process and generate responses (conceptual, non-mathematical overview). Criteria for selecting the right AI tool for a given task — accuracy needs, cost, data sensitivity, and output format. Setting up accounts and workspaces; understanding free-tier versus paid-tier capabilities and limitations
2.	**Prompt Engineering and Generative AI Tools** Anatomy of a good prompt — context, clear instruction, desired format, and constraints. Zero-shot, few-shot, and chain-of-thought prompting techniques and when to use each. Role-based and context-setting prompts (e.g., asking the AI to respond as a reviewer, tutor, or domain expert). Iterative prompt refinement — diagnosing a poor AI response and rewriting the prompt to fix it. Using generative AI for brainstorming, summarizing, and drafting technical content.

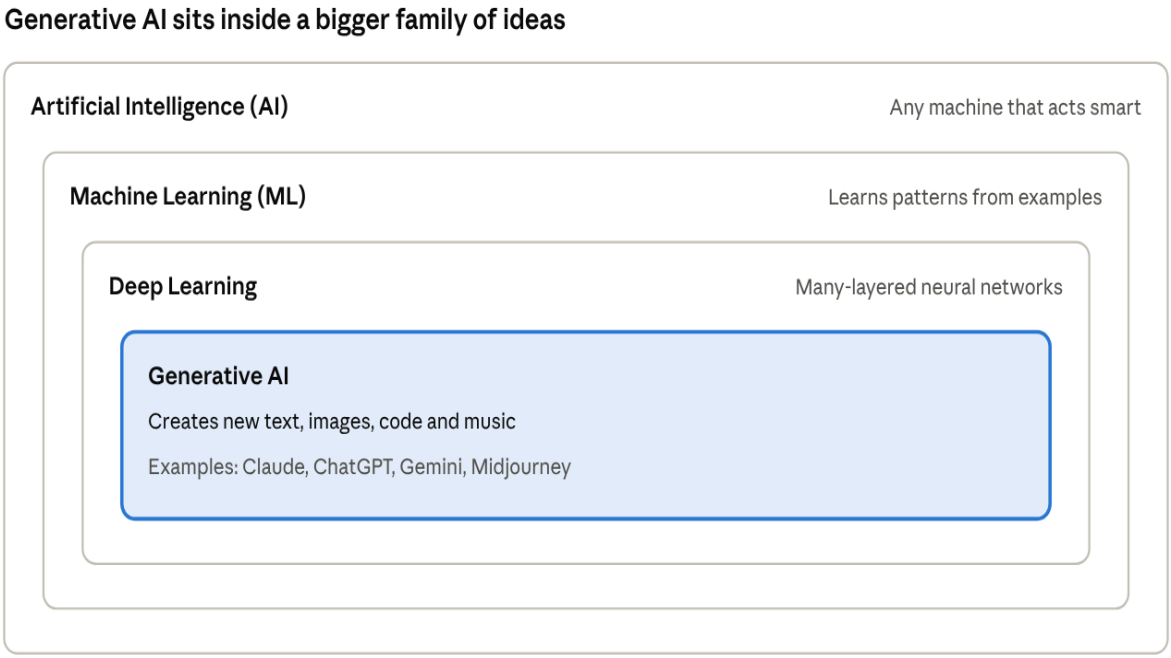
1. Introduction to AI and the AI Tools Landscape

1. What is AI, Machine Learning, and Generative AI?

Artificial Intelligence (AI) is the big idea of making computers do things that normally need human intelligence, such as understanding language, recognizing faces in photos, making decisions, or solving problems. Think of AI as the whole umbrella. Anything that makes a machine act "smart" falls under it, from a simple chess program to a voice assistant on your phone.

Machine Learning (ML) is one way of building AI. Instead of a programmer writing every rule by hand, we show the computer thousands or millions of examples and let it find the patterns itself. For example, if you show a computer many photos labelled "cat" and "dog", it slowly learns what makes a cat look like a cat. It is similar to how a child learns: not by memorizing a rulebook, but by seeing many examples. Machine Learning sits inside AI, and a powerful branch of it called **Deep Learning** (using many-layered "neural networks") drives most of today's breakthroughs.

Generative AI (GenAI) is a newer branch of Machine Learning that does not just recognize or sort things; it **creates** new things. It can write essays, answer questions, produce images, compose music, or generate computer code. ChatGPT, Claude, Gemini, and image tools like Midjourney are all examples. In short: AI is the umbrella, Machine Learning is a method inside it, Deep Learning is a stronger method inside that, and Generative AI is the creative branch built on top of it.



AI, ML, Deep Learning and Generative AI · 4 nested layers

Each inner box is a more specific kind of the box around it.

2. Core Categories of AI Tools

Conversational / text tools. These are chat-style assistants you talk to in plain language. You can ask questions, get explanations, summarize long documents, brainstorm ideas, or translate text. Examples: Claude, ChatGPT, Gemini.

Code assistants. These help programmers write, explain, fix, and test code. You can describe what you want in words and the tool suggests the code, or it can find mistakes (bugs) in code you already wrote. Examples: Claude Code, GitHub Copilot, Cursor.

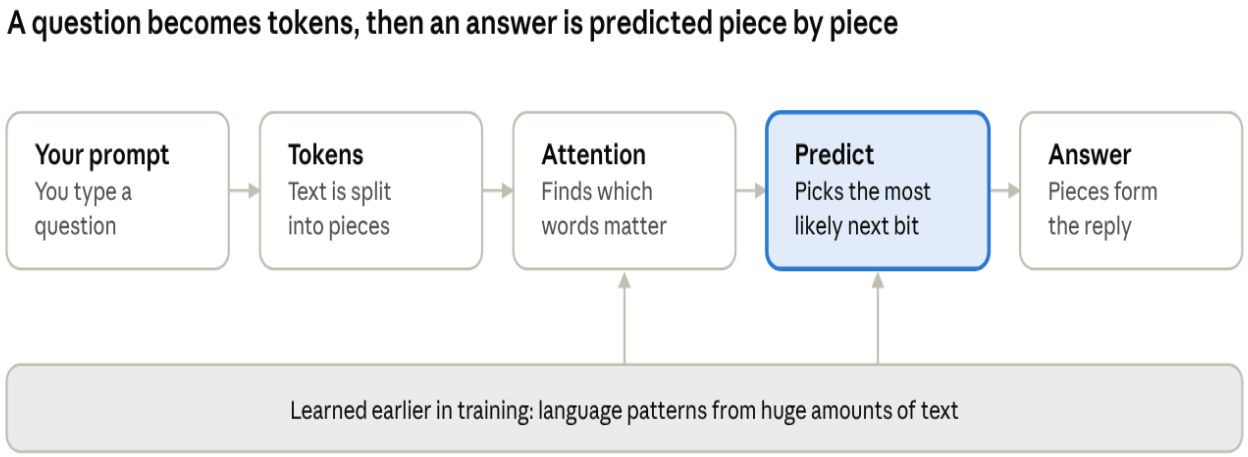
Data-analysis tools. These help you make sense of numbers and spreadsheets. You can upload a file and ask questions like "which month had the highest sales?" and get answers, tables, and charts without writing formulas. Examples: Claude with file analysis, ChatGPT data analysis, Julius AI, Excel with AI features.

Image-generation tools. You type a description (called a prompt), such as "a temple at sunrise in watercolor style", and the tool draws a brand-new picture. They are used for posters, presentations, concept art, and social media. Examples: Midjourney, DALL·E, Stable Diffusion, Adobe Firefly.

Writing and productivity tools. These save time on everyday work: drafting emails, improving grammar and tone, making slides, taking meeting notes, and organizing tasks. Examples: Grammarly, Notion AI, Microsoft Copilot, Otter.ai.

3. How Large Language Models Process and Generate Responses

A **Large Language Model (LLM)** is the engine behind chat tools like Claude and ChatGPT. It is a very large AI system that has "read" an enormous amount of text from books, websites, and articles during a stage called **training**. During training it does not memorize pages like a library. Instead, it learns the patterns of language: which words usually follow which, how sentences are built, how facts and ideas connect, and how different topics are discussed.



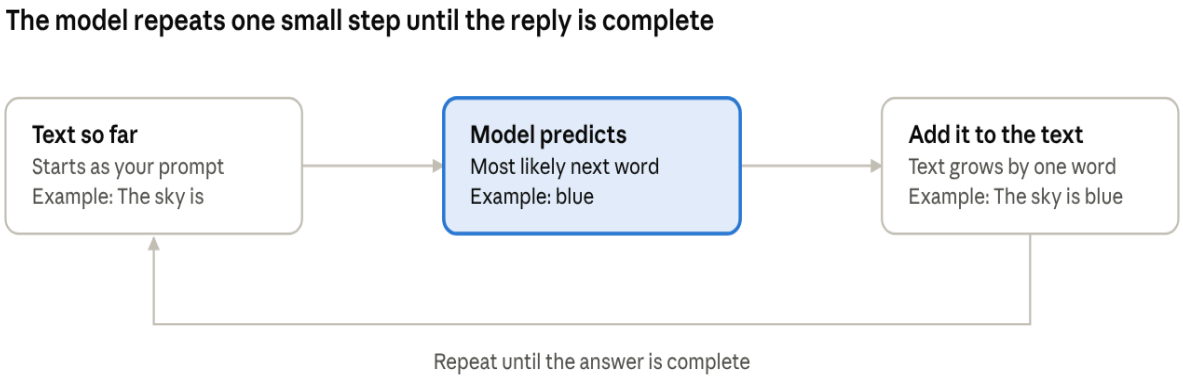
How an LLM turns a question into an answer · 5 steps

The model uses what it learned in training at the attention and prediction steps.

When you type a question, the first step is to break your text into small pieces called **tokens**. A token is roughly a word or part of a word. The model turns these tokens into numbers, because computers work with numbers, and then studies how all the pieces relate to one another. A key trick here is called **attention**: the model figures out which words in your message matter most for understanding the meaning. In the sentence "The bank near the river was muddy", attention helps it understand that "bank" means a riverbank, not a place with money.

Next comes the generation. The model predicts the **most suitable next token**, adds it to the reply, and then repeats the process again and again, one small piece at a time, until the answer is complete. It is like a very advanced version of the "suggested next word" on your phone

keyboard, except it has learned from vastly more text and can keep track of a long conversation. Because it works by predicting likely words, it can sometimes sound confident while being wrong. This is called a **hallucination**, which is why important facts should always be double-checked.



Next-word prediction loop · 3 steps, 1 loop

Every new word is added to the text, and the whole text is read again before the next word is chosen.

The final answer you see is shaped by more than training. Developers fine-tune models with human feedback so replies are helpful, polite, and safe. Your own instructions matter too: the clearer and more detailed your prompt, the better the response. A short formula is: **Input (your prompt) → Tokens → Understanding (attention) → Predict next word, repeat → Final answer.**

4. Criteria for Selecting the Right AI Tool for a Task

No single AI tool is best for everything, so start by asking four simple questions. First, **how accurate must the result be?** For brainstorming a poster idea, a few mistakes are fine. For legal, medical, financial, or academic work, you need a tool that is strong at reasoning, can cite sources, and you must still verify the output yourself. Second, **what is the cost?** Many tools have a free version good enough for light use, while heavy daily use may justify a paid plan. Compare the price against the time it saves you.

Third, **how sensitive is your data?** Never paste passwords, student records, confidential company documents, or personal details into a tool unless you have checked its privacy policy and settings. Some tools may use your chats to improve their models, while business and

enterprise plans usually offer stronger privacy. Fourth, **what output format do you need?** Choose a text assistant for writing, an image tool for pictures, a data tool for charts, and a code assistant for programs. Matching the tool to the format of the final result saves a lot of effort.

5. Setting Up Accounts and Workspaces; Free vs Paid Tiers

Getting started is usually simple. Visit the tool's official website or app store page, sign up with your email address or an existing Google or Microsoft account, and verify your email. Use a strong password and turn on two-step verification for safety. After signing in, take a minute to look at the **settings**: check the privacy options, choose your language, and see what is saved from your chats. Many tools also let you create a **workspace** or **project**, which is a folder-like space where you keep related chats, files, and instructions together, for example one workspace for "Lab Notes" and another for "Personal Writing". Teams can also share a workspace so everyone works with the same files and settings.

Most AI tools offer a **free tier** and one or more **paid tiers**. The free tier is perfect for learning and occasional use, but it normally comes with limits: a cap on how many messages you can send per day or hour, access to a basic or older model, fewer file uploads, slower responses at busy times, and fewer advanced features. Paid tiers usually raise these limits, give access to the most capable models, allow longer documents and more file uploads, and add extras such as deep research, advanced data analysis, image creation, or priority speed.

A smart approach is to **start free**, learn what you actually need, and upgrade only if you keep hitting the limits or need a premium feature for your work. Free and paid plans, prices, and limits change often, so always check the tool's official pricing page for the latest details. Also remember that even a paid plan does not make an AI perfect; you should still rev

Teacher practical examples and student exercises

Experiment 1 (Teacher Demonstration)

Aim

Live comparison of the same general engineering query (e.g., “explain the engineering design process”) answered by two different AI chatbots, highlighting differences in response style and accuracy.

Theory

An AI chatbot is built on a large language model that predicts the next word of a reply from patterns it learned during training. Because every chatbot is trained on different data and tuned in its own way, two chatbots can give different answers to the same question. They may differ in length, tone, structure, and sometimes in correctness, since a model can state a wrong fact in a confident voice. Comparing two answers side by side, and checking facts against a textbook, teaches students to treat AI output as a draft that must be verified.

Procedure

1. Open two different AI chatbots on two browser windows and show them side by side.
2. Type the identical question, “Explain the engineering design process”, into both.
3. Display both answers and let the class read them for two minutes.
4. Compare them using the table below.
5. Verify the design steps against a textbook or trusted website.
6. Ask the same follow-up in both, “Explain it in simple words for a first-year student”, and note how each adapts.

Result

Point	Chatbot A	Chatbot B
Style (short or long, formal or friendly)		
Structure (steps, headings, bullets)		
Accuracy (steps correct and in order)		
Clarity for a beginner		

Conclusion: The two chatbots answered the same query differently in style and detail. Important facts must always be verified before use.

Experiment 2 (Teacher Demonstration)

Aim

Walkthrough of an AI tool’s interface — settings, conversation history, file upload, and model selection — using a common platform.

Theory

Most AI platforms share the same basic layout. **Settings** control privacy, language, and theme. **Conversation history** stores past chats so they can be reopened, renamed, or deleted, and a new chat starts without the memory of old ones. **File upload** lets the user attach a document or image for the AI to read. **Model selection** lets the user choose between faster, lighter models and slower, more capable ones, and free accounts usually offer fewer choices. Button names and positions differ between tools, so the ideas matter more than the exact menus.

Procedure

1. Sign in to a common AI platform and show the main screen on the projector.
2. Open **Settings**; show the privacy options, language, and theme.
3. Show the **conversation history**; start a new chat, reopen an old one, and rename or delete a chat.
4. Click the attach (paperclip) button, upload a small non-confidential file, and ask for a summary.
5. Open the **model selector** and explain the choices available on the account.
6. Ask two students to repeat the four steps on their own devices.

Result

Feature	Where found	What it does	Demonstrated (Yes/No)
Settings			
Conversation history			
File upload			
Model selection			

Conclusion: Students can locate and use the four main parts of an AI tool's interface.

Experiment 3 (Student Exercise)

Aim

Create accounts on two AI platforms and compare their responses to a simple engineering question (e.g., "explain the difference between analysis and design").

Theory

Creating an account gives access to a platform's chat, history, and settings, and sets the limits of its free plan. Asking two platforms the same question is a fair test only if the wording is identical. The answers are then judged on accuracy, clarity, depth, and use of examples. Analysis

means breaking a problem or system into parts to understand it, while design means creating a new solution to meet a need, so a good answer should clearly separate the two.

Procedure

1. Sign up on two AI platforms using your email, and turn on two-step verification if offered.
2. Type the same question into both: “Explain the difference between analysis and design.”
3. Copy each answer into a document or take a screenshot.
4. Compare the answers for accuracy, clarity, and examples, and check one fact in your textbook.
5. Ask one follow-up, “Give me a real-life example”, in both platforms.

Result

Criteria	Platform 1	Platform 2
Correct meaning of analysis		
Correct meaning of design		
Example given		
Ease of understanding		

Conclusion: Write 4 to 6 lines stating which answer was more useful and why, and mention one weakness you noticed.

Experiment 4 (Student Exercise)

Aim

Prepare a one-page comparison chart of five AI tools by category and primary use case.

Theory

AI tools fall into groups: conversational or text assistants, code assistants, data-analysis tools, image generators, and writing or productivity tools. Each group is built for a main purpose, so the best tool depends on the task. Comparing tools by category, primary use, cost, strength, and limitation helps in choosing the right one. Plans and prices change often, so the official website of each tool should be used as the source.

Procedure

1. Select five tools, each from a different category.
2. Visit the official website of each tool and note what it does and what its free plan offers.
3. Fill the chart below with short entries, one line per cell.
4. Check that the chart fits on one page, then submit it.

Result

Tool	Category	Primary use case	Free or paid	Strength	Limitation
1.					
2.					

Tool	Category	Primary use case	Free or paid	Strength	Limitation
3.					
4.					
5.					

Conclusion: The chart shows which tool suits which type of task.

Experiment 5 (Student Exercise)

Aim

Write a short reflection identifying one task in your daily academic life where an AI tool could help.

Theory

AI tools are most useful when matched to a real need, such as making revision notes, understanding a hard topic, planning a project, or checking grammar. Reflection means thinking honestly about where a tool adds value and where it can mislead. Good use of AI keeps the learner in charge: the AI gives a first draft or an explanation, and the student checks the facts and does the final thinking.

Procedure

1. Think about your week and choose one academic task that takes time or feels difficult.
2. Decide which type of AI tool suits it and what you would ask it.
3. Note one risk, such as a wrong fact, and how you will check the answer.
4. Write 150 to 200 words using this start: “One task where AI could help me is ____. I would use a ____ tool by asking it to ____. To make sure the result is correct, I would ____.”

Result

A short reflection of 150 to 200 words stating the task, the tool, the way it will be used, and the method of verification.

Conclusion: AI can support learning when its output is checked and used responsibly.

2. Prompt Engineering and Generative AI Tools

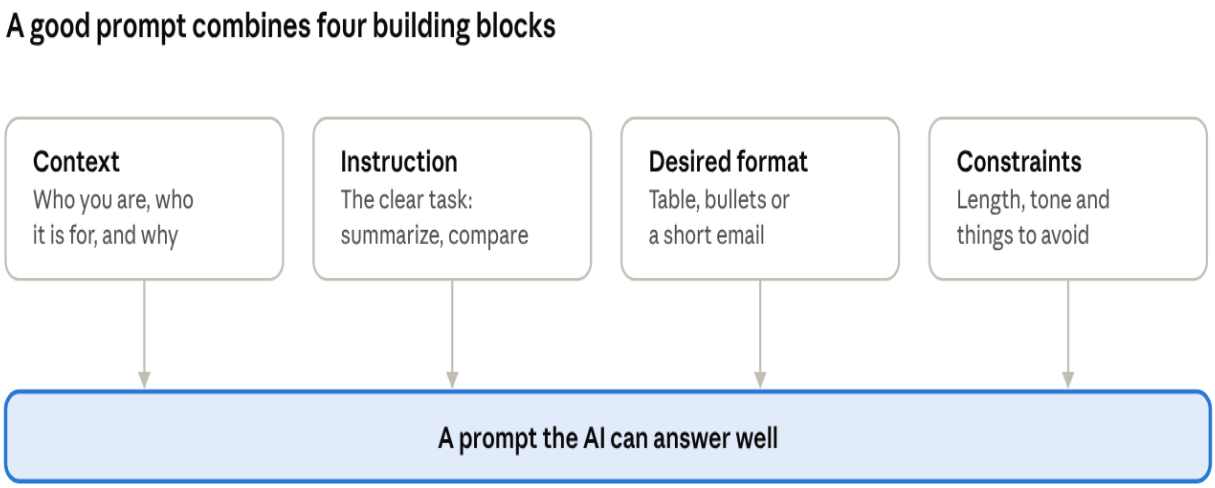
1. Anatomy of a Good Prompt

A **prompt** is simply the message you give an AI tool. Prompt engineering is the skill of writing that message so the AI understands exactly what you want. Think of the AI as a very capable new assistant who knows a lot but knows nothing about your situation. The more clearly you explain the job, the better the result. A vague request like "write about climate" gets a vague answer; a clear request gets a useful one.

A good prompt usually has four parts. **Context** tells the AI the background: who you are, who the audience is, and why you need this. **Instruction** is the clear task, written as an action such as "summarize", "compare", or "explain". **Desired format** says what the answer should look like: a table, five bullet points, a short email, or a 200-word paragraph. **Constraints** are the limits and rules: word count, tone, things to avoid, or the level of difficulty.

Here is a weak prompt turned into a strong one. Weak: "Tell me about blockchain." Strong: "I am a first-year college student with no technical background. Explain blockchain in simple words (context + instruction). Give the answer as five short bullet points with one everyday example

(format). Keep it under 150 words and avoid technical jargon (constraints)." You do not need all four parts every time, but adding even two of them improves the answer a lot.



Four parts of a good prompt · 4 blocks

You can add one block at a time and see how the answer improves.

2. Zero-Shot, Few-Shot, and Chain-of-Thought Prompting

Zero-shot prompting means asking the AI to do a task with no examples, just the instruction. For example: "Translate this sentence into Telugu." It works well for common, simple tasks because the AI has already seen similar tasks during training. **Few-shot prompting** means you show the AI a few examples of what you want before giving the real task. It is like showing a new employee two filled-in forms before asking them to fill in a third. The AI copies the pattern, style, and format of your examples.

Chain-of-thought (CoT) prompting means asking the AI to think step by step before giving the final answer, for example by adding "Explain your reasoning step by step." This helps with math, logic puzzles, planning, and any problem that needs several steps, because writing out the steps makes the AI less likely to jump to a wrong conclusion. It also lets you check the reasoning yourself.

A quick guide to choosing between them:

Technique	What you do	Use it when	Example
Zero-shot	Give only the instruction	The task is simple and common	"Summarize this paragraph in two lines."
Few-shot	Add 2 to 3 examples first	You want a specific style or format	Show two sample product descriptions, then ask for a third
Chain-of-thought	Ask for step-by-step thinking	The task needs reasoning or calculation	"Solve this problem and show each step."

These techniques can be combined. For instance, you can give two examples and also ask for step-by-step reasoning on a hard problem.

3. Role-Based and Context-Setting Prompts

A **role-based prompt** tells the AI who to be before it answers. For example: "Act as a strict exam reviewer", "Act as a patient school tutor", or "Act as a cybersecurity expert." This works because the role changes the vocabulary, depth, and viewpoint of the reply. The same question about a research paper will get very different feedback from a "friendly mentor" than from a "tough journal reviewer".

Context-setting goes one step further by describing the situation: who the audience is, what they already know, and what the goal is. A tutor prompt might say: "Act as a tutor for Class 10 students who find physics difficult. Explain Newton's laws using examples from cricket. Ask me one question at the end to check my understanding." Notice how the role, the audience, the style, and the follow-up are all included.

A few simple tips: pick a role that matches the kind of help you want, say who the answer is for, and be specific about the tone, such as formal, friendly, or encouraging. A role does not make the AI magically more accurate, so you should still check important facts, especially in medical, legal, or financial topics.

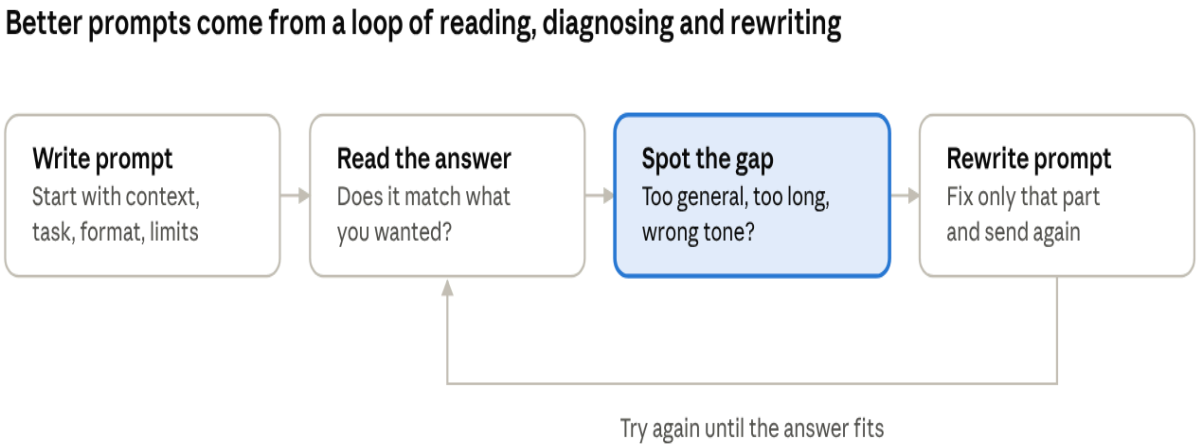
4. Iterative Prompt Refinement: Diagnose and Rewrite

Rarely is the first prompt perfect. **Iterative refinement** means treating the conversation as a loop: write a prompt, read the answer, find what is wrong, improve the prompt, and try again. Think of it like adjusting the focus of a camera until the picture is sharp. Good prompt writers do not give up after one poor answer; they diagnose the problem.

Start by asking what exactly went wrong, because each problem has a different fix:

What went wrong	Likely cause	How to rewrite the prompt
Answer is too general	Missing context	Add who you are, the audience, and the goal
Answer is too long or too short	No length limit	Add a word or point limit
Wrong style or layout	No format given	State the format, or add an example
Mistakes in reasoning	The AI rushed	Ask for step-by-step thinking
Wrong tone	No role or tone given	Add a role and a tone, such as friendly or formal

Example: you ask "Explain machine learning" and get a long, technical answer. You rewrite it as "Explain machine learning to a school student in 100 words using a cooking analogy." The second answer fits your need. You can also simply reply in the same chat with follow-ups like "make it shorter" or "use simpler words", since the AI remembers the conversation.



Prompt refinement loop · 4 steps, 1 loop

Change one thing at a time, so you know which change fixed the problem.

5. Using Generative AI for Brainstorming, Summarizing, and Drafting Technical Content

Brainstorming. AI is excellent at producing many ideas quickly, such as project topics, lab experiment ideas, quiz questions, or titles. Ask for a specific number and a variety, for example: "Give me 10 mini-project ideas on blockchain for beginners, each in one line." Then pick the best ones and ask the AI to expand them. Treat the list as a starting point, not a final decision.

Summarizing. You can paste a long article, report, or meeting notes and ask for a short version. Be clear about the length and the reader, such as "Summarize this in five bullet points for a busy manager" or "Explain the key points in simple words for students." Always compare the summary with the original for important details, because AI can leave out or slightly change facts.

Drafting technical content. AI can create first drafts of lab manuals, study notes, documentation, emails, reports, and code explanations. Give it the topic, the audience, the structure, and any facts it must use. Then review and edit the draft, checking all technical details, numbers, and references yourself. The best way to work is to let AI do the first 70 percent of the typing, and you do the thinking, correcting, and final polishing.

Prompt Engineering: Experiments

Teacher practical examples and student exercises

Experiment 1 (Teacher Demonstration)

Aim

Refine a vague prompt (e.g., “explain project scheduling”) into a precise one step-by-step, showing output improvement at each stage.

Theory

An AI tool can only work with the information it is given, so a vague prompt produces a general answer. A prompt becomes stronger when we add its building blocks one by one: context (who the answer is for), a clear instruction, the desired format, and constraints such as length and tone. Adding one block at a time lets the class see exactly which change improved the output.

Procedure

1. Open an AI chatbot and type the vague prompt (Stage 1). Display the answer.
2. Add the audience (Stage 2) and run it again.
3. Add the format and an example request (Stage 3) and run it again.
4. Add a role and constraints (Stage 4) and run it again.
5. Ask the class to compare all four answers and say what improved each time.

Program (Prompts Used)

> ***Stage 1:*** *Explain project scheduling.*
>
> ***Stage 2:*** *Explain project scheduling to a second-year engineering student who has never managed a project.*

> ***Stage 3:*** *Explain project scheduling to a second-year engineering student who has never managed a project. Give five bullet points and one real-life example.*
>
> ***Stage 4:*** *Act as a project management tutor. Explain project scheduling to a second-year engineering student who has never managed a project. Give five bullet points and one real-life example, in simple words, under 150 words.*

Result

Stage	What was added	Change in the output
1	Nothing (vague prompt)	
2	Audience	
3	Format and example	
4	Role and constraints	

Conclusion: Each added detail made the answer more focused and easier to use, so clear prompts give better results.

Experiment 2 (Teacher Demonstration)

Aim

Demonstrate few-shot prompting by providing examples to guide the AI toward a specific output format (e.g., a formatted comparison table).

Theory

In few-shot prompting, the user shows the AI two or three solved examples before giving the real task. The AI notices the pattern, such as the column names, the length of each cell, and the style, and copies it. It works like showing a new employee two filled-in forms before asking them to fill in a third. It is most useful when a particular format or style is needed.

Procedure

1. Give the task without examples (zero-shot) and show the answer.
2. Give the few-shot prompt below, which contains two solved examples.
3. Compare the two outputs for format, length, and consistency.
4. Change the final item (for example, to "CPM and PERT") and show that the AI keeps the same format.

Program (Prompts Used)

> ***Zero-shot:*** *Compare Gantt chart and PERT chart.*
>
> ***Few-shot:*** *Compare two items in a table with the columns: Feature | Item A | Item B | Best used when.*
>
> *Example 1: Compare Waterfall and Agile. → Feature: Planning | Waterfall: Fixed plan | Agile: Flexible plan | Best used when: Requirements are clear.*

> *Example 2: Compare RAM and ROM. → Feature: Memory type | RAM: Temporary | ROM: Permanent | Best used when: Data must be saved.*
>
> *Now do: Compare Gantt chart and PERT chart.*

Result

Prompt type	Format followed?	Length and style consistent?
Zero-shot		
Few-shot		

Conclusion: Examples guided the AI to the required table format, which a plain instruction did not do reliably.

Experiment 3 (Student Exercise)

Aim

Write three versions of a prompt (basic, detailed, role-based) for the same task and compare outputs.

Theory

The same task can be asked in different ways, and the way of asking changes the answer. A basic prompt states only the task. A detailed prompt adds context, format, and limits. A role-based prompt tells the AI who to be, such as a tutor or a reviewer, which changes its vocabulary, depth, and viewpoint. Comparing the three shows how much control the writer has over the output.

Procedure

1. Choose one task, for example "explain the difference between RAM and ROM".
2. Write the basic version, the detailed version, and the role-based version.
3. Run each prompt in a new chat so earlier answers do not affect the next one.
4. Copy the three outputs and compare them using the table below.

Program (Prompts Used)

> ***Basic:*** *Explain the difference between RAM and ROM.*
>
> ***Detailed:*** *Explain the difference between RAM and ROM to a first-year student in a table with three rows, using simple words, under 120 words.*
>
> ***Role-based:*** *Act as a patient computer-science tutor. Explain the difference between RAM and ROM to a first-year student with an everyday example, and end with one question to check understanding.*

Result

Criteria	Basic	Detailed	Role-based
Clarity			
Matches the format asked			

Criteria	Basic	Detailed	Role-based
Depth and examples			
Tone			

Conclusion: State in 2 to 3 lines which version worked best and why.

Experiment 4 (Student Exercise)

Aim

Use an AI tool to summarize a one-page technical article and verify the summary against the original.

Theory

Summarizing is one of the most useful tasks for AI, but a summary can leave out an important point, change a number, or add something that the article never said. For this reason, a summary should always be checked against the original. A simple check is to list the main points, numbers, and technical terms of the article and see whether each one is present and correct in the summary.

Procedure

1. Select a one-page technical article that is not confidential.
2. Paste the text into an AI tool with the prompt given below.
3. Read the article again and list its main points, numbers, and key terms.
4. Compare the list with the AI summary and fill in the table.
5. Note any missing, changed, or added information.

Program (Prompt Used)

> *Summarize the following technical article in five bullet points for a second-year engineering student. Keep all numbers and technical terms exactly as written. Do not add anything that is not in the article.*

Result

Point or number in the original	Present in summary?	Correct?

Conclusion: Record how many points were correct, missing, or wrong, and whether the summary can be trusted without checking.

Experiment 5 (Student Exercise)

Aim

Draft a chain-of-thought prompt to solve a multi-step numerical/logical problem (e.g., a unit-conversion or estimation problem) and check the reasoning.

Theory

Chain-of-thought prompting asks the AI to think step by step and show its working before giving the final answer. For problems that need several steps, such as unit conversions, this makes mistakes less likely, and it lets the student see each step and check it. The final answer is trusted only after every step has been verified by hand.

Procedure

1. Choose a multi-step problem, for example the water-tank problem below.
2. Solve it yourself first and keep your steps as the reference.
3. Write a chain-of-thought prompt that asks for each step and each unit conversion.
4. Run the prompt and compare each step of the AI with your own working.
5. Mark every step as correct or wrong, and give the final result.

Program (Prompt Used)

> *A water tank supplies 2.5 litres per second. How many cubic metres of water flow out in 3 hours? Solve this step by step. Show every unit conversion, write the calculation for each step, and then give the final answer with its unit. At the end, double-check your answer.*

Result

Step	Reference working	AI step correct?
1. Convert time	3 hours = 3 × 3600 = 10,800 seconds	
2. Find volume in litres	2.5 × 10,800 = 27,000 litres	
3. Convert to cubic metres	27,000 ÷ 1000 = 27 m^3	

Conclusion: The final answer is 27 cubic metres. Record whether the AI reasoning matched the reference at every step.