



CHATGPT FOR STUDY / STUDENTS AND TEACHERS

PROMPTS

**52 ways to make
ChatGPT teach you.**

How this works

Type `/teachme Grade 11 photosynthesis` into ChatGPT and it will teach you photosynthesis, from the ground up, at your pace, and it will not sigh at you for asking twice.

Every prompt on this list works like that. One line, one job. 52 of them means there is one for whatever has you stuck tonight.

Start with one

You don't need the whole list. Pick the prompt that matches what you're stuck on, type it, and put your topic straight after it:

`/teachme Grade 10 algebra, solving for x`

That's the whole setup. Once you're comfortable, run 2 or 3 in a row and it starts to feel like a tutor sitting next to you.

Three habits that double the value

1 Name the subject and the grade.

`"/teachme Grade 11 chemistry, mole calculations"` gets you a far better answer than `"/teachme chemistry"`. The more you tell it, the less it has to guess.

2 Stack 2 or 3.

`/teachme`, then `/quizzme`, then `/mark`. Taught, tested, marked, in 3 prompts and about 20 minutes. That beats an hour of rereading your notes.

3 Push back.

Too easy, type `/challenge`. Too hard, type `/simple`. Still lost, type `/analogy`. You are allowed to keep asking.

Where the line sits

- Use it to understand the work. Do the work yourself.
- Anything you hand in gets typed by you, in your words.
- ChatGPT sounds most confident when it is wrong. Check anything assessable against your textbook or your syllabus, not against a second AI.
- Your school has an AI policy. Read it. It outranks this sheet.
- Close the screen and explain your answer out loud. If the words are there, you learned it. If they're not, the machine learned it.

Learn and capture



Learn it

<code>/teachme [topic]</code>	Teaches it step by step, starting from nothing.
<code>/explain [topic]</code>	A clear explanation of how it actually works.
<code>/simple [topic]</code>	The same idea again, in plain words.
<code>/examples [topic]</code>	Real examples so the idea has something to hold onto.
<code>/analogy [topic]</code>	Explains it using something you already understand.
<code>/5minlesson [topic]</code>	The whole idea in 5 minutes.
<code>/deepdive [topic]</code>	Goes further than the syllabus asks you to go.



Go deeper

<code>/socratic [topic]</code>	Teaches by asking you questions instead of answering them.
<code>/mastery [topic]</code>	Teaches, tests, finds the gaps, then teaches again.
<code>/challenge [topic]</code>	Makes every question harder than the one before it.
<code>/studyplan [topic]</code>	Spreads the topic across the days you have left.



Capture it

<code>/summarise [text]</code>	Cuts it down to the ideas that matter.
<code>/keypoints [topic]</code>	Only the points you'd lose marks for missing.
<code>/notes [topic]</code>	Study notes in a shape you can copy out by hand.
<code>/stickynotes [topic]</code>	Breaks it into chunks small enough to remember.
<code>/studyguide [topic]</code>	A structured revision guide for the whole unit.
<code>/visual [topic]</code>	Draws it as a diagram, mind map, or flow chart.
<code>/cheatsheet [topic]</code>	One page you can scan the night before.
<code>/timeline [topic]</code>	Puts the events in order, with dates.

Connect and test



Connect it

<code>/compare [A] [B]</code>	Sets two things side by side, same and different.
<code>/causeeffect [topic]</code>	What caused it, and what it went on to cause.
<code>/definitions [topic]</code>	The terms you're expected to use correctly.
<code>/vocab [topic]</code>	The subject words worth knowing cold.



Remember it

<code>/flashcards [topic]</code>	Question and answer pairs to write onto cards.
<code>/remember [topic]</code>	Memory tricks for the parts that won't stick.
<code>/mnemonic [topic]</code>	One phrase that holds a whole list together.



Test it

<code>/quizzme [topic]</code>	One question at a time, answered on paper.
<code>/multiplechoice [topic]</code>	Multiple choice, with wrong answers that look right.
<code>/truefalse [topic]</code>	A fast true or false run to find the soft spots.
<code>/testme [topic]</code>	A full knowledge test across the whole topic.
<code>/exam [topic]</code>	Exam-style questions in the format you'll actually sit.
<code>/practice [topic]</code>	Practice questions until the method runs by itself.
<code>/hint</code>	A nudge, with the answer held back.
<code>/reveal</code>	The answer, once you've had a real go at it.
<code>/traps [topic]</code>	The mistakes students keep making, and how to dodge them.

Feedback and writing



Get feedback

<code>/mark [answer]</code>	Marks your answer and shows where the marks went.
<code>/feedback [answer]</code>	Turns a good answer into a better one.
<code>/tone [text]</code>	Tells you where your writing sounds too casual.



Write it

<code>/brainstorm [topic]</code>	Generates angles before you commit to one.
<code>/essayplan [question]</code>	A structure for the essay, built before you write.
<code>/paragraph [question]</code>	Builds one paragraph properly, point through to evidence.
<code>/argument [topic]</code>	Develops the case for.
<code>/counterargument [topic]</code>	Develops the case against, so your case holds up.
<code>/debate [topic]</code>	Both sides, fairly put.
<code>/homework [question]</code>	Walks you through it rather than handing you the answer.



Check it

<code>/factcheck [claim]</code>	Tests whether a claim holds up.
<code>/research [topic]</code>	Digs past the first paragraph of the internet.
<code>/sources [topic]</code>	Points you at sources worth citing.
<code>/spellcheck [text]</code>	Finds the spelling you've stopped noticing.
<code>/grammar [text]</code>	Fixes the grammar, and tells you why it was wrong.
<code>/translate [text]</code>	Translates it with the sense left intact.
<code>/rhetoric [text]</code>	Shows you how a speech or a text is working on you.

When it matters

A slash gets you moving in 2 seconds. Some nights that's all you need. When the assignment is worth real marks, type the full paragraph instead. Fill in every bracket before you press enter. The brackets are where your thinking goes, and an empty one gets you a generic answer.

01 The Questioner /socratic

Makes you do the thinking

I'm studying [the causes of World War II] and I need to [write 1,000 words]. Don't give me the answers. Ask me probing questions, one at a time, until I arrive at the core ideas myself.

02 The Opponent /counterargument

Finds the holes before your marker does

I'm arguing that [your thesis]. Act as a critical opponent. Name the 3 weakest points in my logic and give me the strongest counter-arguments a scholar would use against me. Don't rewrite my essay. Just give me the critique.

03 The Trap Spotter /traps

Names the mistakes before you make them

Explain how to solve [problem or concept] step by step, as if I'm a Grade 12 student preparing for a final exam. At each stage, name the mistakes students commonly make and tell me how to avoid them under exam conditions.

04 The Editor /tone

Tells you where you sound like a text message

Evaluate this text written for [type of assignment]: [paste text]. Analyse the tone and level of formality. Tell me where it's too casual or imprecise, and suggest specific ways to strengthen the vocabulary and clarity. Don't rewrite it for me.

05 The Planner /studyplan

Turns 14 days into a plan you can follow

Design a 14-day spaced study plan for [subject] covering these topics: [list them]. I'm a Grade 12 student preparing for a final exam. Specify what to review each day, and build it for deep understanding rather than short-term memory.

When it matters

06 The Translator /analogy

Explains it with something you already know

Explain [concept] using a detailed analogy from [your favourite show, sport, or game]. Connect each technical part of the concept to a specific part of the analogy.

07 The Examiner /multiplechoice

Tests you on your own notes

Based on these notes: [paste notes], write 5 multiple-choice questions and 1 long-answer challenge question. If I get one wrong, explain the reasoning behind the correct answer and give me a memory hack I can use in the exam.

08 The Deep End /deepdive

Goes past what the syllabus asks for

Explain [topic] to me as if I'm a Grade 12 student preparing for a final exam. Include the nuances, the debates academics still argue about, and how the topic connects to real-world issues today.

09 The Connector /compare

Links two subjects that look unrelated

Explain the connections between [topic A] and [topic B] as if I'm an advanced Grade 12 student. Show me the unexpected influences and what it means when these two shape each other, in academic study and in the real world.

10 The Close Reader /rhetoric

Shows you how the words are working on you

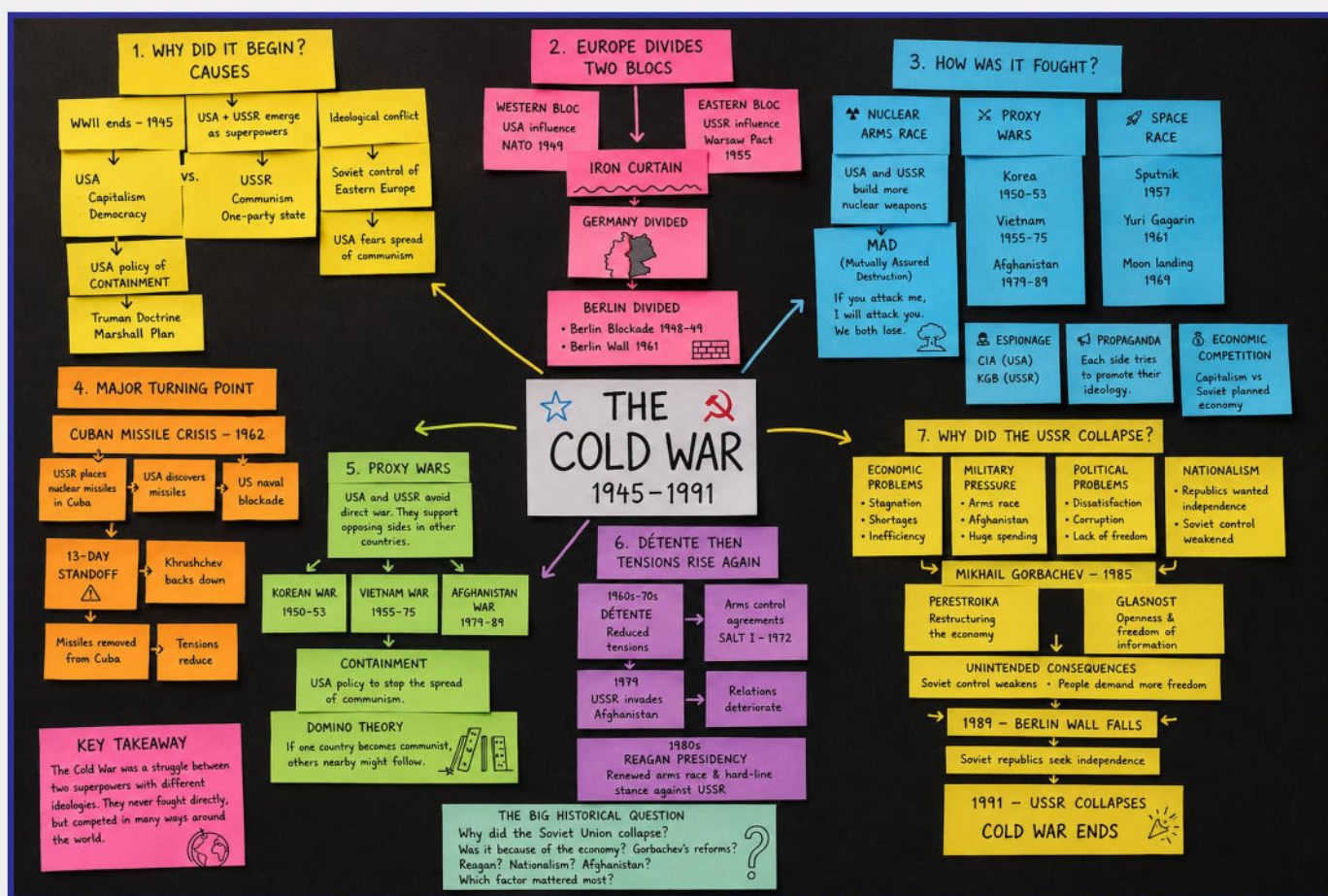
Analyse this speech or text: [paste it]. I'm a Grade 12 student preparing for a final exam. Identify the rhetorical devices used, such as ethos, pathos, and logos. Explain how each one works inside the argument and how it strengthens the persuasion.

What one prompt built

A whole unit of the Cold War, on one wall, from a single prompt. Here is the prompt that made it.

/stickynotes

Create an image of a sticky-note revision wall for [the Cold War], for a Grade 12 [Modern History] student in [Queensland]. Group it into 7 numbered themes, one colour per theme, with a title card in the centre and arrows showing what caused what. Keep every note under 20 words. Finish with a key takeaway note and one big exam question note.



Then do the part that counts

- Swap the brackets for your subject, your topic, and your state.
- Copy the wall onto real cards in your own handwriting. Reading a wall someone else built teaches you nothing.
- Check the dates and the names against your syllabus before you trust them.

For the staffroom

Your students are already using these tools. The only question left is whether they are using them to think or to skip thinking. This sheet is built to push them toward the first one.

Introduce it in 10 minutes

Put the sheet up. Pick a topic the class already knows well. Run `/teachme`, then `/quizzme`, then `/mark` on the projector, and feed it a deliberately weak answer so the class watches it get marked down. That one demonstration lands harder than a policy slide.

Set the guardrails before the tools

The panel on page 2 is written for students, so read it out loud. Then add your school's AI policy on top of it. This sheet defers to your policy every time.

Four prompts carry most of the value

`/socratic`, `/quizzme`, `/mark`, and `/hint`. Each one makes the student produce something before ChatGPT does. The other 48 are convenience. These 4 are the teaching. Pages 6 and 7 give the long-form version of 10 prompts, worth setting for assessment tasks.

Where it goes wrong

ChatGPT is fluent when it is wrong, and it is most often wrong on state-specific syllabus content, local case studies, and anything published in the last year. Send students to the syllabus document to check, never to a second chatbot.

Pen and paper still wins

Have students write the answers by hand before they type anything. The prompt generates the question. The student generates the thinking.

Copy it freely

Print this sheet for your class, put it in the student diary, or stick it on the wall. No permission needed inside your school.

A 4-prompt cycle you can set as homework

1. `/teachme` [tonight's topic]
2. `/examples` [tonight's topic]
3. `/quizzme` [tonight's topic], answered on paper
4. `/mark` [a photo or typed copy of what they wrote]

Ask them to hand in the handwritten answers, not the chat log. The prompt supplies the question. The student supplies the thinking.



The best students ask better questions.

The HopeFULL Institute runs in-person seminars for secondary students across Australia and beyond. 750 schools, 25 years, one message. Hope is a skill, and a skill can be taught.

Book a seminar

thehopefullinstitute.com/bookings

More free classroom resources

thehopefullinstitute.com/classroom-pack

