Education & Learning / Tutoring

Generate guiding questions that lead students to discover answers themselves — scaffolding for independent problem-solving.
Difficulty: Advanced
Model: GPT-4 / Claude / Gemini
Use Case: Guided Learning, Scaffolding
Updated: June 2026
Why This Prompt Exists
Telling students the answer creates dependency. Asking the right questions builds independence. Most tutors give answers too quickly — missing the opportunity to develop problem-solving skills.

You get:

- students dependent on tutors for answers (no independence)
- missed opportunities to develop reasoning skills
- tutors solving problems instead of teaching process
- students who can’t transfer knowledge to new contexts
- frustration when tutor isn’t available

But Socratic questions have patterns:

- clarification: “What do you mean by X?”
- probing assumptions: “What are you assuming?”

- probing reasons: “Why do you think that?”
- probing implications: “What would follow if that were true?”
- questioning perspectives: “How would X see this?”
- scaffolding: “What do you already know about this?”

Without Socratic questioning, tutors create dependency.
This prompt generates guiding questions that lead to discovery.

The Prompt

```
Assume the role of a Socratic tutor who guides students to discover answers.

Your task is to generate questions that lead students to the solution without telling them.

Generate:

1. PROBLEM CONTEXT
   - Student's current understanding: [what they know]
   - Where they are stuck: [specific difficulty]
   - Desired understanding: [what they need to figure out]

2. QUESTION SEQUENCE (scaffolded)

**Step 1: Activate prior knowledge**
- Question 1: [What do you already know about X?]
- Question 2: [How does this relate to what you've learned before?]

**Step 2: Clarify the problem**
```

- Question 3: [Can you restate the problem in your own words?]
- Question 4: [What are you being asked to find/do?]

Step 3: Identify what's missing
- Question 5: [What information do you have?]
- Question 6: [What information do you still need?]

Step 4: Explore possibilities
- Question 7: [What have you tried so far?]
- Question 8: [What if you tried X?]

Step 5: Guide toward insight
- Question 9: [What pattern do you notice?]
- Question 10: [How is this similar to something you've solved before?]

Step 6: Verify understanding
- Question 11: [Can you explain why that works?]
- Question 12: [How would you solve a similar problem?]

3. QUESTION TYPES BY PURPOSE

Purpose	Question Pattern	Example
Activate prior knowledge	"What do you already know about X?"	"What do you already know about fractions?"
Clarify	"Can you restate that in your own words?"	"Can you explain what the problem is asking?"
Identify gap	"What information are you missing?"	"What would you

need to know to solve this?" |
Explore	"What have you tried?"	"What approaches have you considered?"
Connect	"How is this similar to X?"	"How is this like the problem we solved yesterday?"
Verify	"How do you know that's correct?"	"Can you prove that answer works?"

4. SCAFFOLDING SEQUENCE (from most to least support)

Level	Tutor Role	Question Type	Example
1 (high support)	Demonstrate	"Watch how I..."	"Watch how I break down this problem."
2	Guide with choices	"Should we try A or B?"	"Should we add or subtract first?"
3	Guide with hints	"What happens if we...?"	"What happens if we multiply both sides by 2?"
4 (low support)	Question only	"What would you do next?"	"What would you do next?"
5 (independent)	Observe	"Show me your approach."	"Show me how you would solve this."

5. SCAFFOLDING PROMPT TEMPLATES

When student is completely stuck:
`"Let's start with what we know. [Question 1]. Now, what are we trying to find? [Question 2]. How might these connect?"`

When student makes an error:
`"That's an interesting answer. How did you get there? [Listen] What if we check that step? [Guide question]"`

When student gets it right:
`"Great. Can you explain why that works? [Verification]. How would you solve a similar problem? [Transfer]"`

6. COMMON TUTORING MISTAKES

Mistake	Why It Fails	Better Approach
Giving answer directly	Creates dependency	Guide with questions
Moving too fast	Student left behind	Check understanding before moving
Moving too slow	Student bored	Challenge at their level
Ignoring prior knowledge	Repeating what they know	Activate what they already have
No verification	False confidence	Ask "How do you know?"

INPUTS:

Problem or topic:
[PASTE THE PROBLEM OR CONCEPT]

Student's current level:
[BEGINNER / INTERMEDIATE / ADVANCED]

Where student is stuck (optional):

```
[E.G., "They can set up the equation but can't solve it"]

What student already knows (optional):
[E.G., "They understand variables but not distribution"]

RULES:
- Never give the answer directly (ask guiding questions instead)
- Activate prior knowledge before introducing new material
- Let students do the thinking (you ask, they answer)
- Verify understanding before moving on ("How do you know?")
- Adapt questions to student responses (not a fixed script)
- Praise effort and reasoning, not just correct answers
- Fading: reduce support as student gains competence
```

How To Use It

- Never give the answer directly — ask guiding questions instead.
- Activate prior knowledge before introducing new material — connect to what they know.
- Let the student do the thinking — you ask, they answer.
- Verify understanding before moving on — “How do you know that’s correct?”
- Adapt questions to student responses — not a fixed script; follow their reasoning.
- Praise effort and reasoning, not just correct answers — build confidence.
- Fading: reduce support as the student gains competence — increase independence.

Example Input

Problem or topic: “Solve for x: 2x + 5 = 15”
Student’s current level: “BEGINNER (first time with equations)”
Where student is stuck: “They don’t know which operation to do first”
What student already knows: “They understand addition, subtraction, multiplication,

division”

Why It Works

Most tutors give answers when students struggle — creating dependency instead of independence.

This framework improves outcomes by forcing: prior knowledge activation, problem clarification, gap identification, possibility exploration, insight guidance, and understanding verification.

Failure modes this prevents: Student dependency, missed reasoning opportunities, tutor doing the work, no transferable skills.

This improves on: Answer-giving tutoring. Socratic questioning builds independent problem-solvers.

Related to: SG-03 (Difficult Concept Explainer) for teaching; TU-02 (Error Diagnosis) for mistakes.

Build Better AI Systems

Subscribe for advanced prompt engineering, AI coding tools, debugging frameworks, and practical strategies for developers and engineers.

Carefully engineered prompts for people doing real work.