Education & Learning / Tutoring

Create problems that require applying learned skills in new contexts — transfer assessment for genuine mastery.
Difficulty: Advanced
Model: GPT-4 / Claude / Gemini
Use Case: Transfer Assessment, Mastery Testing
Updated: June 2026
Why This Prompt Exists
Students can solve problems identical to what they practiced — but fail when the context changes. Near transfer isn’t real mastery. Most assessments never test far transfer.

You get:

- students who can’t apply skills in new contexts
- false confidence from identical problem practice
- no assessment of genuine understanding
- frustration when real-world problems look different
- inability to adapt learned procedures

But transfer problems have structures:

- near transfer: same concept, different numbers or surface features
- medium transfer: same concept, different context or format
- far transfer: same concept, different domain entirely

- transformation: requires combining multiple concepts
- adaptation: requires modifying the learned procedure

Without transfer problems, you don’t know if they understand.
This prompt generates problems that test genuine transfer.

The Prompt

```
Assume the role of an assessment designer who creates transfer
problems.

Your task is to generate problems that test application in new
contexts.

Generate:

1. SOURCE SKILL/CONCEPT
   - Skill: [what student learned]
   - Example problem: [what they practiced]
   - Key principles: [underlying rules or patterns]

2. TRANSFER DISTANCE

| Distance | Description | Surface Change | Deep Structure | Example |
|----------|-------------|----------------|----------------|---------|
| Near | Same format, different numbers | Numbers only | Identical |
Same word problem, different values |
| Medium | Same concept, different context | Context, wording | Same |
Different scenario, same math |
| Far | Different domain, same principle | Domain, format, context |
```

Same underlying pattern | Physics problem using same algebraic structure |
| Transformation | Requires combination with other skills | Multiple changes | Adapted | Multi-step, multi-concept |

3. TRANSFER PROBLEM SET

Near Transfer (same format, new numbers)
Problem: [different numbers, identical structure]
Solution approach: [same steps, different values]

Medium Transfer (same concept, new context)
Problem: [different scenario, same underlying principle]
Solution approach: [recognize pattern, apply same method]

Far Transfer (different domain, same principle)
Problem: [unrelated domain, same logical or mathematical structure]
Solution approach: [abstract the principle, map to new domain]

Transformation (combine with other skills)
Problem: [requires adaptation or combination]
Solution approach: [modify procedure, integrate multiple concepts]

4. TRANSFER SUCCESS CRITERIA

Transfer Level	What Student Must Do	Evidence of Mastery
Near	Apply same steps to new numbers	Correct answer within time

Medium	Recognize underlying pattern	Correct setup without prompting
Far	Abstract principle to new domain	Justify why same approach works
Transformation	Adapt procedure to new constraints	Novel solution, not memorized

5. COMMON TRANSFER BARRIERS

Barrier	Why Students Struggle	Mitigation
Surface feature fixation	Focus on wording, not structure	Teach pattern recognition explicitly
Context dependence	Can only solve in learned context	Practice multiple contexts
Over-reliance on examples	Can't solve without template	Fade examples gradually
Procedural only	No understanding of why	Teach principles, not steps

6. TRANSFER DIAGNOSTIC (if student struggles)

If student fails near transfer: [issue with basic skill]
If student fails medium transfer: [issue with pattern recognition]
If student fails far transfer: [issue with abstraction]
If student fails transformation: [issue with integration]

INPUTS:

```
Skill/concept taught:
[E.G., "Solving two-step equations: 2x + 3 = 7"]

Example problem practiced:
[E.G., "A number times 2 plus 3 equals 7. Find the number."]

Student's current performance:
[CAN SOLVE NEAR / CAN SOLVE MEDIUM / CAN SOLVE FAR / CAN TRANSFORM]

Desired transfer level to test:
[NEAR / MEDIUM / FAR / TRANSFORMATION]

RULES:
- Test near transfer first (if they can't do near, don't test further)
- Medium transfer requires recognizing underlying pattern (not just
memorizing)
- Far transfer reveals genuine understanding (not procedural mimicry)
- Transformation is the highest level of mastery (requires adaptation)
- Provide different surface features for each transfer level
- Keep the deep structure constant across transfer levels
- Diagnose which transfer level fails to identify learning gaps
```

How To Use It

- Test near transfer first — if they can’t do near transfer, don’t test further; reteach the basics.
- Medium transfer requires recognizing the underlying pattern — not just memorizing procedures.
- Far transfer reveals genuine understanding — not procedural mimicry.
- Transformation is the highest level of mastery — requires adaptation, not just

application.

- Provide different surface features for each transfer level — change context, wording, numbers, domain.
- Keep the deep structure constant across transfer levels — the underlying principle should be the same.
- Diagnose which transfer level fails to identify specific learning gaps.

Example Input

Skill/concept taught: “Solving two-step equations: ax + b = c”

Example problem practiced: “A number times 2 plus 3 equals 7. Find the number.”

Student’s current performance: “CAN SOLVE NEAR (different numbers)”

Desired transfer level to test: “MEDIUM (same concept, different context)”

Why It Works

Students can solve problems identical to practice but fail when context changes — false mastery. Near transfer isn’t enough.

This framework improves outcomes by forcing: transfer distance classification, problem generation at each level, success criteria definition, barrier identification, and diagnostic mapping.

Failure modes this prevents: False confidence from identical practice, inability to apply in new contexts, no genuine mastery assessment.

This improves on: Identical problem testing. Transfer problems reveal genuine understanding.

Related to: TU-03 (Worked Example) for demonstration; TU-02 (Error Diagnosis) for transfer failures.

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.