Internal Report on
Prompt Design Patterns for AI in Education

Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Fundacja Rozwoju Systemu Edukacji. Neither the European Union nor the granting authority can be held responsible for them.

General Prompt Engineering Patterns

General Prompt Patterns are foundational strategies for interacting with AI. They are not tied to a specific subject or pedagogy but describe how prompts can be structured in any learning or problem-solving context. These patterns form the “building blocks” of effective AI use: they control how information is requested, how outputs are shaped, and how reasoning is revealed.

In education, General Patterns are valuable because they: Contextual

- Provide flexible methods that can be applied across subjects like mathematics, computer science, and beyond.
- Help instructors and students steer AI behavior — making responses more accurate, structured, or creative.
- Serve as the foundation for Education-Oriented Patterns, which adapt these general methods into explicit teaching and learning strategies.

The following General Patterns include core techniques. Each describes how AI can be guided, what it is most useful for, and what limitations to be aware of.

Zero Shot Pattern

In Zero-Shot Pattern [1] an AI system is asked to perform a task or answer a question without being given any examples, demonstrations, or prior context. The model relies entirely on its background knowledge and training data to generate a response. This pattern highlights the AI’s generalization ability, since it must infer the intended output structure and style from the prompt alone.

Use Cases

- Retrieving definitions or explanations of concepts quickly.
- Generating first-pass answers that can be refined later.
- Starting a brainstorming session without constraints.
- Getting baseline responses to compare with more structured prompts.

Limitations

- Outputs can be vague, shallow, or misaligned with expectations.
- Higher risk of factual errors, since no guidance is provided.
- Does not ensure adherence to structure, tone, or pedagogy.
- Unsuitable for complex or technical tasks where precision matters.

Few-Shot Pattern (Example-Driven)

The Few-Shot Pattern [2] enhances AI output by supplying a handful of examples of the desired task before requesting a new response. These examples act as mini-demonstrations, guiding the model's interpretation of the instructions, the structure of the output, and the level of detail expected. In effect, the examples "teach" the AI how to respond in context, reducing ambiguity and improving reliability.

Use Cases

- Showing sample answers to essay questions and generating similar ones.
- Training the AI to follow a particular tone, format, or reasoning style.
- Creating custom practice exercises aligned with prior examples.

Limitations

- Designing effective examples requires time and effort.
- Too few examples may not be enough to guide the AI; too many can overwhelm.
- Risk of biasing the AI output too narrowly toward the provided samples.
- Students may become passive if they only copy from examples rather than reasoning independently.

Chain of Thought Pattern

The Chain of Thought Pattern [3] prompts the AI to display its intermediate reasoning steps instead of jumping directly to a final answer. This makes the AI's process transparent, showing how it arrives at conclusions through logical progression. For learners, this pattern is valuable because it mirrors human problem-solving, revealing not only *what* the answer is but *why* it makes sense.

Use Cases

- Explaining math or logic problems step by step.
- Walking through the structure of a computer program.
- Helping students debug reasoning by examining each stage of thought.
- Demonstrating critical thinking or decision-making processes.

Limitations

- Responses may become long-winded or repetitive.

- If the AI makes a mistake, the entire chain of reasoning may still look convincing, creating false confidence.
- Not always necessary for simple factual queries where brevity is preferred.
- May slow down learning if students are overwhelmed by detail.

Persona-Based Patterns

The Persona-Based Pattern [8] directs the AI to adopt a specific identity, role, or perspective when generating responses. By framing the interaction through a persona (e.g., expert tutor, beginner student, coach, or historical figure), the AI tailors its tone, depth, and style to match the persona's expected behavior. This helps create more engaging and context-appropriate interactions, and allows learners to approach knowledge from multiple viewpoints.

Use Cases

- Acting as a supportive tutor who explains concepts at a beginner level.
- Simulating a peer who asks questions and debates ideas.
- Adopting the persona of a historical figure for contextualized learning.
- Role-playing as a critical reviewer or coach to provide constructive feedback.

Limitations

- Personas can drift if instructions are not clear or reinforced.
- Risk of reinforcing stereotypes if personas are oversimplified.
- Students may take persona responses too literally and confuse them with authoritative truth.
- Requires careful design to ensure educational value rather than entertainment only.

Constraint-Based Pattern

The Constraint-Based Pattern [5] shapes the AI's output by applying explicit rules or limitations. These constraints may include word count, output length, formatting requirements, or structural rules such as "respond in bullet points" or "use an outline format." By narrowing the scope, the AI is guided to produce responses that are concise, focused, and aligned with specific expectations. This pattern provides greater control over both the form and function of AI-generated responses.

Use Cases

- Producing short summaries or explanations within a fixed word limit.

- Requiring step-by-step guides, outlines, or tabular outputs.
- Designing exercises that enforce a specific coding or writing style.
- Helping students focus on clarity rather than verbosity.

Limitations

- Strict constraints may result in overly rigid or unnatural responses.
- Overuse of constraints can suppress creativity or depth.
- AI may still misinterpret unclear or conflicting rules.
- Not suitable for open-ended exploratory tasks where freedom is valuable

Negative Prompting Pattern

The Negative Prompting Pattern [6] directs the AI to avoid certain types of content, language, or approaches. Instead of focusing only on what should be included, the prompt specifies what must not appear in the response. This provides an additional layer of control, ensuring the AI stays aligned with the user's intentions by filtering out irrelevant, inappropriate, or overly advanced content.

Use Cases

- Simplifying explanations by excluding technical jargon for beginners.
- Avoiding biased, sensitive, or culturally inappropriate material.
- Removing unnecessary tangents so responses remain focused.
- Training students to recognize how constraints shape communication.

Limitations

- Negative instructions must be precise; vague "don'ts" may be ignored.
- AI might still include restricted content if not reinforced.
- Too many exclusions can limit richness and variety.
- Risk of responses becoming bland or oversimplified.

Instruction Based Patterns

The Instruction-Based Pattern [7] emphasizes clear, direct, and structured commands given to the AI. Instead of relying on the AI to infer goals, the user specifies exactly what is required, often breaking the task into steps or explicitly stating the format. This pattern reduces ambiguity and increases reliability, making it one of the most effective for producing actionable and consistent outputs.

Use Cases

- Generating tutorials or guided learning materials.
- Creating exercises with explicit directions and expected outcomes.
- Designing lesson content where consistent structure is critical.
- Assisting students in understanding how to write precise, goal-oriented prompts.

Limitations

- Overly prescriptive prompts can stifle flexibility and creativity.
- Requires time and skill to design good instructions.
- AI may still misinterpret poorly written steps.
- Not ideal for exploratory or creative tasks where openness is desired.

8. **Question- Refinement Patterns** – Prompt pattern uses the Generative AI service to help develop a better version of the prompt instead. For example – Whenever I ask a question to suggest an improved version to use instead.

Data-Driven Patterns

The Data-Driven Pattern [9] structures the AI's output around specific data provided in the prompt. This could be numerical data, tables, code snippets, or factual context that the AI must use to guide its response. By grounding the task in actual data, this pattern increases relevance, accuracy, and alignment with real-world scenarios.

Use Cases

- Analyzing a given dataset or SQL table.
- Generating explanations or visualizations based on raw numbers.
- Creating assignments or exercises tied to student-provided data.
- Encouraging evidence-based reasoning and output.

Limitations

- Output quality depends heavily on the accuracy of the input data.
- Larger datasets may exceed model input size limits.

- AI may misinterpret data structure or oversimplify analysis.
- Not suitable for purely theoretical or creative tasks without data anchors.

Contextual Patterns

The Contextual Pattern [8] provides the AI with **background information or a scenario** to shape its responses. Rather than asking in isolation, the user embeds the task in a context — such as a course topic, a dataset, or a learner's level of experience. This enables the AI to align its output more closely with the user's goals, producing responses that are relevant, situational, and personalized.

Use Cases

- Explaining SQL queries using a given database schema.
- Adapting explanations to a learner's background (e.g., beginner vs. advanced).
- Providing case-specific insights in fields like medicine, law, or engineering.
- Simulating real-world problem scenarios for applied practice.

Limitations

- Requires effort to craft accurate and sufficient context.
- AI may misinterpret or underuse provided context.
- Too much detail can overwhelm or confuse the model.
- Inconsistent use of context may lead to fragmented learning experiences.

Game-Play Patterns

The Game-Play Pattern [8] transforms AI interaction into a game-like experience by defining rules, challenges, scoring systems, or win conditions. By gamifying the learning process, it increases motivation, engagement, and interactivity. Instead of passively consuming explanations, students participate in playful, structured activities that make learning more memorable.

Use Cases

- Quiz-style competitions with points and scoring.
- Coding challenges framed as levels or puzzles.
- Gamified exercises to reinforce practice and review.

- Motivational scenarios that make routine drills more engaging.

Limitations

- Risk of oversimplifying content into “just a game.”
- Requires careful balance to maintain educational rigor.
- Poorly designed rules can frustrate learners.
- May prioritize fun over depth if not properly scaffolded.

Template Patterns

The Template Pattern [8] provides the AI with a structured format or skeleton containing placeholders that must be filled. By embedding the expected structure directly into the prompt, the AI produces responses that align neatly with the required format. This ensures consistency, comparability, and standardization across multiple outputs.

Use Cases

- Generating lesson plans or lab worksheets in a set structure.
- Producing SQL explanations with consistent sections (e.g., query, purpose, output).
- Designing rubrics, reports, or outlines with predictable formats.
- Teaching students the importance of structure in communication.

Limitations

- Rigid templates may not accommodate complex or nuanced tasks.
- Risk of repetitive and mechanical outputs.
- Requires carefully designed placeholders for effectiveness.
- Less flexibility for creative or exploratory learning tasks.

Meta Language Creation Pattern

The Meta-Language Creation [8] Pattern establishes a shared symbolic or shorthand language between the user and the AI. This meta-language can consist of custom symbols, labels, or definitions that the AI consistently interprets throughout the interaction. By creating a controlled vocabulary, complex ideas can be communicated more efficiently and consistently.

Use Cases

- Defining shorthand for SQL operators or mathematical notation.

- Establishing placeholder terms in collaborative coding exercises.
- Teaching the concept of controlled vocabularies or domain-specific languages.
- Streamlining repetitive prompts across long interactions.

Limitations

- New learners may struggle to distinguish meta-language from standard terms.
- Risk of misinterpretation if rules are not clearly defined or reinforced.
- May become confusing if the meta-language is too complex.
- Less useful for short or one-time queries where shared jargon isn't needed.

Alternatives Approaches Patterns

The Alternatives Approaches Pattern [11] asks the AI to produce multiple distinct solutions, strategies, or perspectives for the same problem. By comparing the differences, learners can better understand the flexibility of problem-solving, identify strengths and weaknesses of each method, and cultivate critical evaluation skills.

Use Cases

- Demonstrating different SQL queries that yield the same result.
- Presenting alternative problem-solving strategies in mathematics.
- Comparing writing styles or argument structures in essays.
- Encouraging divergent and creative thinking in project-based learning.

Limitations

- Not all alternatives will be valid or equally useful.
- Quality and depth may vary across outputs.
- Too many options can overwhelm learners.
- Requires guidance to help students critically evaluate differences.

Fact Check List Pattern

The Fact Check List Pattern [8] requires the AI to generate a list of factual claims or assumptions that underpin its response. The learner or instructor can then independently verify these claims. This shifts some responsibility back to the human user, encouraging critical engagement with AI outputs and fostering information literacy skills.

Use Cases

- Teaching students to verify information instead of blindly trusting it.
- Supporting academic writing and research tasks.
- Checking the validity of AI-generated explanations in technical subjects.
- Encouraging learners to practice evidence-based validation.

Limitations

- The “facts” generated may themselves be incorrect.
- Verification requires effort and external resources.
- Students may over-trust the checklist if not careful.
- Not all domains lend themselves easily to factual breakdowns.

Tail Generation Patterns

The Tail Generation Pattern [13] instructs the AI to append a recurring reminder, note, or reflection prompt at the end of each response. This acts as a “tail” that ensures continuity across a conversation, keeping the learner aligned with overarching goals or reinforcing study habits.

Use Cases

- Reminding students of learning objectives at the end of each explanation.
- Encouraging reflection questions after every session.
- Maintaining continuity in long-term projects or multi-session learning.
- Reinforcing consistent study behaviors and routines.

Limitations

- Repetition may become annoying or distracting.

- Overuse can cause students to ignore or skip the tail content.
- Less valuable for short, one-off tasks.
- Risk of cluttering outputs with redundant reminders.

Semantic Filter Patterns

The Semantic Filter Pattern [10] instructs the AI to include, highlight, or exclude information based on semantic rules. Instead of generating free-flowing responses, the AI processes information selectively, filtering content according to the user's requirements. This helps in refining responses to match specific needs or to ensure compliance with privacy and safety standards.

Use Cases

- Removing personally identifiable information (PII) from text.
- Extracting only the SQL commands from a larger explanation.
- Highlighting key concepts or keywords for study notes.
- Streamlining content for summaries or targeted reviews.

Limitations

- Filtering may be inconsistent; AI might miss key items or remove too much.
- Requires clear criteria for inclusion/exclusion.
- May lead to fragmented or incomplete explanations.
- Not effective if rules are vague or contradictory.

Flipped Interaction Patterns

The Flipped Interaction Pattern [8] inverts the usual dynamic: instead of the learner asking and the AI answering, the AI initiates by asking questions. This shifts responsibility toward the learner, who must articulate knowledge, reasoning, or problem-solving approaches. It fosters active learning by putting the student in the role of explainer, while the AI acts as a coach or examiner.

Use Cases

- Diagnostic questioning to reveal knowledge gaps.
- Simulating Socratic dialogue to stimulate reasoning.

- Interactive tutoring where students must “teach back” concepts.
- Engaging students in active, self-directed exploration.

Limitations

- AI may ask trivial, repetitive, or poorly sequenced questions.
- Students may feel frustrated if questions are unclear.
- Requires framing so the learner understands the educational purpose.
- Not suitable for students who lack baseline knowledge to engage.

Creativity-Enhancing Patterns

The Creativity-Enhancing Pattern [12] prompts the AI **to go beyond conventional or predictable responses**, encouraging novelty, imagination, and unconventional thinking. It shifts the focus from accuracy and structure to exploration and originality. For learners, this fosters creative problem-solving and helps them break out of rigid approaches to thinking.

Use Cases

- Brainstorming ideas for projects, essays, or designs.
- Generating novel analogies for difficult concepts.
- Exploring “out-of-the-box” solutions to open-ended problems.
- Inspiring curiosity and imagination in learning activities.

Limitations

- Responses may be unrealistic, impractical, or irrelevant.
- Harder to control for factual accuracy.
- Risk of prioritizing creativity at the expense of clarity.
- May not suit technical tasks that require precision.

Multi-Step Interaction

The Multi-Step Interaction Pattern [4] is a structured approach where the AI is guided through a series of prompts that build on each other, instead of being asked for the final answer immediately. Each step produces an intermediate output that becomes the basis for the next. This mirrors how humans approach complex tasks: breaking them into manageable parts, refining results along the way, and gradually building toward a polished final product. The strength of this pattern lies in scaffolding — it makes problem-solving processes transparent and avoids overwhelming learners.

Use Cases

- Complex writing tasks: brainstorming ideas → drafting → refining → polishing.
- Programming exercises: outlining logic → writing pseudocode → producing working code.
- Problem-solving in math/science: defining knowns → setting equations → solving → interpreting results.
- Curriculum design: generating learning objectives → structuring lessons → drafting assessments.
- Student projects: developing research questions → gathering sources → synthesizing findings → drafting reports.

Limitations

- Requires careful planning of the sequence of prompts; poorly designed steps can confuse rather than clarify.
- More time-consuming than one-shot prompting, especially for simpler tasks.
- Students may become over-reliant on the step structure, instead of learning to independently scaffold tasks.

Cognitive Verifier Pattern

The Cognitive Verifier Pattern [8] is a prompting strategy where the AI is instructed to check, evaluate, or validate reasoning and outputs against explicit rules or criteria. Instead of only producing an answer, the AI also performs a verification step — confirming whether the solution is correct, logically consistent, or aligned with specified guidelines. This mirrors the human process of self-checking work before finalizing it, and supports both accuracy and metacognitive reflection.

Use Cases

- Mathematics & Programming: Have the AI verify whether a solution follows the correct steps or whether a query/code snippet adheres to rules.
- Essay Writing: Ask the AI to review a draft against clarity, coherence, or rubric criteria.

- Student Self-Assessment: Learners can paraphrase their own understanding, then prompt the AI to check if it is accurate.
- Error Diagnosis: Pairing this pattern with others (e.g., Error Injection, Explain-Back) to confirm correctness of explanations or outputs.

Limitations

- The AI's "verification" may still miss subtle errors or introduce false confidence if flawed reasoning is judged as correct.
- Requires clear and specific criteria for checking; vague rules lead to unreliable validation.
- Can create extra cognitive load for students if verification becomes repetitive or overly detailed.
- Works best as a complement to human oversight, not a replacement for it.

Adaptive Quiz Prompting

The Adaptive Quiz Prompting Pattern [14] uses AI to create or adjust questions dynamically based on a learner's previous responses. Instead of delivering a static set of questions, the AI tailors the difficulty, type, or focus of subsequent questions to match the learner's current level of understanding. This mirrors adaptive testing approaches, where assessment becomes personalized and responsive, promoting engagement and targeted learning.

Use Cases

- Formative assessment: Providing learners with questions that adjust in real time to their performance.
- Skill-building: Moving from simple to complex tasks in programming, mathematics, or language learning.
- Differentiation: Supporting diverse learners by giving each a customized practice path.
- Exam preparation: Generating question sets that progressively challenge the student until mastery.

Limitations

- Relies on AI's ability to accurately judge performance — misinterpretations can lead to mismatched difficulty.
- Requires careful framing to avoid frustration (too easy or too hard).
- Students may become dependent on adaptive scaffolding rather than practicing persistence.

- Needs clear boundaries to ensure assessment validity and fairness.

Role-Play Scenario Prompt

The Role-Play Scenario Prompt [8] directs the AI to simulate multiple perspectives or stakeholders in a situation, creating an interactive environment where learners can engage in dialogue, negotiation, or debate. This pattern leverages the AI's flexibility to embody roles, helping students explore social, ethical, or professional dimensions of learning through immersive experiences.

Use Cases

- Ethics education: Simulating a debate between conflicting viewpoints (e.g., privacy vs. security).
- Professional training: AI takes the role of a client, manager, or patient for role-play practice.
- Peer learning: Simulating group discussions where AI takes on diverse student perspectives.
- Decision-making exercises: Exploring trade-offs by interacting with different roles.

Limitations

- Quality of simulation depends on clarity of role instructions; poorly framed roles may seem artificial.
- Risk of reinforcing stereotypes if roles are not carefully designed.
- Learners may confuse simulated perspectives with authoritative facts.
- Requires debriefing to connect role-play back to learning objectives.

Cross-Domain Transfer Prompt

The Cross-Domain Transfer Prompt [15] encourages the AI to apply knowledge or concepts from one subject area to another, supporting analogical reasoning and interdisciplinary learning. By bridging contexts, this pattern helps learners see connections across domains, deepening understanding and stimulating creative thinking.

Use Cases

- STEM integration: Applying concepts from physics to computer science, or from mathematics to economics.
- Creative analogies: Explaining programming loops using musical rhythms or sports strategies.

- Interdisciplinary projects: Linking historical events to modern technology developments.
- Critical thinking: Encouraging students to test whether knowledge can transfer meaningfully to new domains.

Limitations

- Transfers may produce weak or misleading analogies if not guided carefully.
- Learners may overgeneralize and misapply concepts.
- Risk of trivializing complex ideas if cross-domain links are oversimplified.
- Requires teacher intervention to validate and refine cross-domain insights.

Educational and Cognitive strategies/patterns using AI in Education

Education-Oriented Prompt Patterns are adaptations of general prompting techniques that are designed to directly support teaching and learning. While General Patterns describe how to shape AI behavior in any context, these patterns connect explicitly to pedagogical goals, such as building knowledge, practicing skills, stimulating creative thinking, fostering reflection, designing instruction, or providing assessment and feedback.

These patterns matter because they:

- Offer structured methods for students to learn with AI, not just receive answers.
- Help instructors design activities, scaffolds, and assessments that align with learning objectives.
- Promote active, reflective, and critical engagement with course content in mathematics, computer science, and beyond.

The Education-Oriented Patterns are organized into six groups:

- Knowledge Construction — building, explaining, and clarifying concepts.
- Active Practice — applying skills, testing knowledge, and debugging errors.
- Creative Thinking — stimulating analogies, alternative approaches, and divergent problem-solving.
- Reflection & Meta-Learning — fostering self-awareness, confidence, and critical evaluation.
- Instructor Design — supporting lesson planning, scaffolding, and differentiated instruction.
- Assessment & Feedback — enabling rubrics, adaptive testing, and peer-style feedback.

Together, these categories show how AI can be integrated not just as an information source, but as a collaborative partner in both teaching and learning.

I. Knowledge Construction Patterns

- **Socratic Reversal**: Is a pedagogical prompt pattern [16] where the learner evaluates, challenges, or teaches the AI — reversing the traditional dynamic of "student asks, AI explains." The student becomes the *teacher, critic, or tutor*, while the AI takes on a *flawed peer or novice* role. This creates a metacognitive learning loop that deepens understanding through reasoning and reflection.
    - Encourages active knowledge construction
    - Develops critical thinking and self-explanation
    - Exposes misconceptions in real-time

  Relation to General Patterns: [*Flipped Interaction Pattern*]: Core alignment: the AI now asks the questions, or the student assumes the teacher's role.
- **Feynman Prompt**: The Feynman Prompt [16] is based on the principle that if you can't explain something simply, you don't understand it well enough (attributed to physicist Richard Feynman). In this prompt pattern, the learner or instructor asks the AI to explain a concept in the simplest possible terms, often imagining the audience as a child, beginner, or someone unfamiliar with the topic. This promotes clarity, conceptual understanding, and foundational learning, especially useful in technical subjects like SQL, programming, or mathematics.
    - Clarifies foundational concepts
    - Reduces cognitive overload
    - Helps with scaffolded learning, especially in early labs

  Relation to General Patterns: [**Persona-Based Pattern**]: The AI adopts the persona of a teacher addressing a young or novice learner. [**Contextual Pattern**]: The learner sets a scenario — "explain this to someone with no background" — to shape output.
- **Explain-Back** : is a dialogic pattern [16] where students paraphrase or re-express what the AI just explained, in their own words. It extends the learning conversation by requiring active processing of AI-generated explanations. After the AI teaches something, the student says: "Let me try explaining that back to you. Tell me if I got it right." This not only reinforces learning but also activates metacognition — students think about how they understand the concept and receive feedback on their clarity.
  Relation to General Patterns: [**Cognitive Verifier Pattern**]: Student checks understanding by validating their paraphrased version against AI's logic. [**Flipped Interaction Pattern**]: Temporarily reverses the roles — the student becomes the explainer.
    - Strengthens comprehension and memory
    - Develops self-monitoring and confidence in verbalizing ideas
    - Trains students to not just receive, but to process and communicate knowledge
- **Gap Finder:** Gap Finder [17] is a prompt pattern where AI is used to diagnose what a learner does not yet know or fully understand. Instead of providing explanations or

answers directly, the AI acts like a diagnostic tutor, asking questions, posing challenges, or giving incomplete tasks to help the student (or teacher) discover knowledge gaps. This pattern supports formative assessment, metacognition, and targeted review, making it especially valuable in early- to mid-stage learning.

  - Personalized discovery: AI finds where the learner is stuck
  - Encourages curiosity and repair: Students want to fill the gaps once revealed
  - Reveals overconfidence: Forces learners to articulate and test what they think they know

  Relation to General Patterns: [**Flipped Interaction Pattern**]: The AI asks the student questions instead of answering. [**Test Me Pattern**]: Similar structure — but Gap Finder focuses on revealing gaps, not just assessing knowledge.

- **Concept Contrast**: is a prompt pattern[18] where the AI is asked to compare two related concepts, highlighting their differences, similarities, and use cases. This helps learners clarify subtle distinctions and build stronger mental models by seeing what makes concepts different but related.
  - Sharpens discrimination between look-alike concepts
  - Reduces conceptual confusion

  Relation to General Patterns: [**Contextual Pattern**]: Students provide two concepts and ask AI to clarify context-specific usage.

- **Progressive Elaboration**: Progressive Elaboration [19] is a prompt pattern where the AI is asked to explain a concept, solve a problem, or generate a solution step by step, instead of jumping directly to the final answer. It encourages sequential reasoning, scaffolding, and process awareness — making it ideal for tasks that require multi-step thinking.
  - Helps novices follow logic without being overwhelmed
  - Encourages debugging and understanding of intermediate steps
  - Reinforces the process, not just the product

  Relation to General Patterns: [**Chain of Thought Pattern**]: Strong alignment — both promote step-by-step reasoning instead of final-answer dumping. [**Multi-Step Interaction Pattern**]: Builds incrementally toward full understanding or final output.

II. Active Practice Patterns

- **Show-Then-Do**: Show-Then-Do [20] is a classic instructional pattern adapted for AI tutoring, where ChatGPT first demonstrates a task, and then the student is asked to complete a similar one independently. The AI provides a worked example, and the student mirrors or adapts that process — a strategy grounded in cognitive apprenticeship and scaffolded learning. This pattern is especially useful in teaching syntax-heavy tasks like programming, or math, where modeling a process before practice can boost confidence and accuracy.
  - Provides immediate modeling and feedback
  - Reduces cognitive load for beginners

    - Helps students focus on variations, not just construction

  Relation to General Patterns: [***Few-Shot Pattern***]: The "Show" phase is an example that the AI provides — modeling what good output looks like.

- **Error Injection:** Error Injection [21] is a prompt pattern where the AI intentionally introduces a mistake into a query, explanation, or process. The student's task is to spot, explain, and correct the error. This pattern is rooted in error-based learning, which helps learners develop diagnostic skills, increase attention to detail, and deepen conceptual understanding. In this approach, the mistake is a feature, not a bug — and it becomes the focus of analysis and reflection.
    - Trains debugging skills and attention to detail
    - Builds error resilience (students learn how to recover from failure)
    - Encourages active learning instead of passive reading

  Relation to General Patterns: [***Cognitive Verifier Pattern***]: Student must evaluate whether the AI's reasoning or output follows rules. [***Fact Check List Pattern***]: Student may verify the AI's claims or results by cross-referencing with known rules.

- **Prompt Me First**: Prompt Me First is a prompt pattern where the AI does not immediately give an answer — instead, it prompts the student to try first. The learner is asked to explain their approach, make a prediction, or attempt a solution before receiving help. This pattern builds initiative, confidence, and self-monitoring, while reducing over-reliance on AI for answers. It turns the AI into a guide, not a crutch, and promotes productive struggle — an essential aspect of deep learning.
    - Encourages ownership of ideas and problem-solving strategies
    - Prevents shortcut behavior (copy-pasting answers)
    - Improves long-term retention by reinforcing what students already know

  Relation to General Patterns: [***Flipped Interaction Pattern***]: The AI defers to the student — flipping from passive reception to active production. [***Cognitive Verifier Pattern***]: The AI acts as a reviewer or coach, offering verification after the student attempt.

- **Test Me:** Test Me is a prompt pattern where students ask the AI to quiz them, generate practice tasks, or pose challenge questions to check their understanding. This pattern shifts the AI into the role of a formative assessor or tutor, helping learners self-evaluate their mastery of concepts. It supports active recall, which has been proven to enhance long-term retention. Students use this pattern to practice safely, assess what they know, and reinforce learning through feedback loops.
    - Promotes retrieval practice, a highly effective study technique
    - Reinforces self-directed learning
    - Helps students identify weak areas early and safely
    - Encourages engagement through interactivity and gamified learning

  Relation to General Patterns: [***Fact Check List Pattern***]: Can be combined by asking for a list of key concepts tested in the quiz. [Cognitive Verifier Pattern]: Students verify their understanding and receive feedback after attempting answers.

- **Rewrite Challenge**: The Rewrite Challenge pattern invites students to transform a correct solution into an alternative form. The goal is not just to achieve the same result, but to do so using a different method, construct, or syntax. This enhances flexibility, structural awareness, and depth of understanding — especially in fields like SQL, programming, logic, and writing.
    - Encourages flexible problem solving
    - Builds deeper structural understanding
    - Encourages creative, modular thinking and efficiency

  Relation to General Patterns: [***Alternative Approaches Pattern***]: Promotes multiple valid ways to solve the same problem. [***Constraint-Based Pattern***]: Often includes rules like "Don't use …"
- **Step-by-Step Debugger Prompt**: The Step-by-Step Debugger Prompt [22] has AI act as a debugging assistant that identifies mistakes and explains them progressively, step by step. This helps learners trace logic, analyze errors, and understand why a solution fails, much like using a real debugger in programming or walking line by line through a proof.
    - Develops systematic error-analysis skills.
    - Strengthens logical tracing in programming and proofs.
    - Builds resilience by treating errors as learning opportunities.
    - Encourages precision and attention to detail.

  Relation to General Patterns: Builds on [**Chain of Thought**] (stepwise reasoning), related to [**Error Injection**] (but errors come from learner, not AI)., connects with [**Cognitive Verifier Pattern**](validating correctness step by step).
- **Mathematical Proof Explainer:** AI breaks down mathematical proofs [23] into small logical steps, explaining not only what is done but *why* each step follows. This makes formal reasoning more accessible and supports learners in building both intuition and rigor.
    - Builds stepwise reasoning in mathematics.
    - Strengthens logical rigor and justification skills.
    - Reduces cognitive overload by scaffolding complexity.
    - Encourages reflection on both how and why proofs work.

  Relation to General Patterns: Aligns with [**Progressive Elaboration]** (layered detail), uses [Chain **of Thought**] (stepwise reasoning).

III. Creative Thinking Patterns

- **Many Ways Prompt:** The Many Ways Prompt [24] is a flexible prompt pattern that asks the AI (or the student) to solve or explain the same problem using multiple distinct approaches. The goal is to show that there is more than one valid way to solve a problem — and to help learners compare and contrast the reasoning, advantages, and limitations of each. This pattern promotes divergent thinking, conceptual versatility, and deep comparative analysis — critical in subjects like SQL, programming, math, and writing.

  - Helps students break out of rigid thinking
  - Strengthens understanding of syntax equivalence
  - Encourages self-evaluation: "Which version would I use, and why?"

  Relation to General Patterns: [***Alternative Approaches Pattern***]: Direct mapping — it's designed to surface multiple valid options. [***Creativity-Enhancing Pattern***]: It encourages novel or less conventional ways of solving the problem.

- **Analogy Builder**: The Analogy Builder [25] is a prompt pattern where students or instructors ask the AI to explain a complex concept using a metaphor or analogy that relates to something familiar. This supports conceptual understanding by bridging abstract ideas with everyday experiences. It's especially powerful for novices, visual learners, and multidisciplinary thinkers.
  - Makes abstract concepts tangible
  - Helps novice learners break into complex topics
  - Sparks imaginative thinking and cross-domain connections

  Relation to General Patterns: [***Persona-Based Pattern***]: AI adopts a role (e.g., elementary school teacher) and crafts age-appropriate analogies. [***Creativity-Enhancing Pattern***]: Students think about how to connect different domains

- **Persona Teaching:** Persona Teaching [26] is a powerful prompt pattern where the student explains a concept to ChatGPT acting as a specific persona, such as a beginner, child, confused peer, or even a skeptical critic. This reverses the typical teacher–learner dynamic and puts the student in the role of the instructor. By teaching the AI, students clarify their own understanding and fill gaps.
  - Clarify and consolidate understanding by requiring students to teach concepts.
  - Strengthen communication skills through audience adaptation.
  - Encourage metacognitive reflection by exposing knowledge gaps.

  Relation to General Patterns: [***Flipped Interaction Pattern***]: The usual AI-as-teacher dynamic is flipped — the student becomes the explainer.

- **What-If Scenarios**: What-If Scenarios [27] is a prompt pattern where students or instructors ask AI to explore hypothetical changes, variations, or alternative situations. This pattern encourages learners to extend their understanding by analyzing how changing inputs, rules, or assumptions affects the outcome. It supports deeper conceptual learning, scenario-based thinking, and exploratory learning.
  - Develops predictive reasoning ("what would happen if…")
  - Fosters understanding of rules, boundaries, and exceptions

  Relation to General Patterns: [***Contextual Pattern***]: Changes the context slightly to test comprehension. [**Cognitive Verifier Patter**]: Promotes logical checking of assumptions and consequences

- **Misconception Reveal:** Misconception Reveal [21] is a prompt pattern where the AI is asked to simulate common student mistakes, misunderstandings, or incorrect logic. The student's task is to identify what's wrong, explain why, and correct it. This pattern strengthens error awareness, builds conceptual clarity, and prevents shallow understanding by focusing on what not to do — and why not. It leverages the fact that AI can generate realistic mistakes on demand, which makes it a powerful tool for targeted remediation and diagnostic instruction.

  - Strengthen learners' ability to detect and correct errors in reasoning or syntax.
  - Build conceptual clarity by contrasting misconceptions with correct logic.
  - Develop resilience and diagnostic skills when confronting mistakes.

  Relation to General Patterns: [***Fact Checklist Pattern***]: Students may cross-check AI's logic against known SQL rules. [***Cognitive Verifier Pattern***]: Students are verifying correctness through conceptual analysis.

- **Debate Simulation:** Debate Simulation is a prompt pattern where AI takes on two or more contrasting roles and conducts a structured debate on a technical or conceptual issue. Learners observe or engage with the dialogue to evaluate arguments, weigh trade-offs, and refine their own reasoning. This helps students move beyond single-answer thinking and develop a deeper appreciation of conditions where different approaches may be valid.
  - Promote critical evaluation of competing perspectives in math and computer science.
  - Strengthen reasoning by requiring justification of trade-offs and conditions.
  - Encourage learners to engage actively with alternative approaches.

  Relation to General Patterns: [**Persona-Based Pattern**]: Debate Simulation depends on AI adopting distinct personas (e.g., "recursion advocate" vs. "iteration advocate"), which is a direct extension of persona role-play. [**Concept Contrast Pattern**]: While Concept Contrast highlights similarities and differences, Debate Simulation enacts those distinctions dynamically through dialogue, making them more vivid. [**Alternative Approaches Pattern**]: The debate naturally surfaces multiple valid solutions or perspectives, aligning with the goal of presenting alternatives for evaluation.

IV. Reflection & Meta-Learning Patterns

- **Confidence Check**: Confidence Check [28] is a reflection pattern where learners state how certain they are about their answers, and the AI responds by probing, verifying, or offering feedback. This encourages students to monitor their own certainty, distinguish between guesswork and knowledge, and build metacognitive awareness.
  - Encourage self-monitoring of certainty and accuracy.
  - Develop metacognitive skills by linking confidence to evidence.
  - Support better calibration between "feeling correct" and "being correct."

  Relation to General Patterns: **[Cognitive Verifier Pattern]**: The AI helps validate whether the learner's reasoning aligns with rules or facts, reinforcing self-assessment. **[Flipped Interaction Pattern]**: Instead of the AI declaring correctness, the learner initiates by declaring confidence, and the AI adapts its response accordingly. **[Tail Generation**

Pattern]: Can append a recurring “confidence scale” check after tasks, reinforcing ongoing reflection.

- **What Did I Miss?:** What Did I Miss? [17] is a reflection patternwhere the learner presents their solution or explanation and asks the AI to identify elements that were overlooked or left incomplete. Instead of providing the whole answer, the AI highlights omissions, gaps, or missing reasoning, helping students refine and strengthen their work.
  - Train learners to review their own work for completeness.
  - Reduce overconfidence by highlighting overlooked steps or details.
  - Strengthen mastery by filling missing elements in reasoning or solutions.

  Relation to General Patterns: **[Cognitive Verifier Pattern]**: AI checks the learner’s solution against expected rules or key elements, surfacing what is absent. **[Gap Finder Pattern]**: Shares the diagnostic function of revealing knowledge gaps, but focuses specifically on omissions in student work. **[Contextual Pattern]**: AI tailors feedback to the specific subject area, e.g., SQL clauses, proof steps, or algorithm analysis.
- **AI-as-Coach**: AI-as-Coach is a reflection pattern where the AI provides constructive guidance, encouragement, and targeted suggestions for improvement. Instead of delivering direct solutions, the AI acts like a supportive coach — offering feedback on strategies, study habits, and areas for growth. This shifts focus from content delivery to learner development.
  - Encourage reflective learning and continuous improvement.
  - Build learner confidence through supportive, constructive feedback.
  - Guide students toward effective study strategies and skill development.

  Relation to General Patterns: **[Persona-Based Pattern]**: AI adopts the persona of a coach or mentor, shaping tone and feedback style. **[Instruction-Based Pattern]**: Feedback often follows structured steps (acknowledge strengths → suggest improvements → provide practice ideas). **[Tail Generation Pattern]**: Can be combined to provide ongoing motivational or coaching prompts at the end of sessions.
- **Study Partner Simulation**: Study Partner Simulation is a reflection pattern where the AI interacts with the learner as if it were a peer working on the same material. Instead of acting as an expert tutor, the AI takes the role of a study partner — asking clarifying questions, suggesting ideas, and comparing reasoning. This creates a collaborative atmosphere that encourages active dialogue and shared exploration.
  - Foster engagement by simulating collaborative peer learning.
  - Encourage articulation and comparison of reasoning.
  - Build confidence through practice in a low-stakes, partner-like setting.

  Relation to General Patterns: **[Persona-Based Pattern]**: AI adopts the persona of a peer learner rather than an expert. **[Flipped Interaction Pattern]**: Roles shift dynamically — sometimes the learner explains, sometimes the AI does, mimicking real peer collaboration. **[Game-Play Pattern]**: Can be extended into light challenges or turn-taking, making study sessions interactive.

- **Reverse Role Play**: Reverse Role Play [26] is a reflection pattern where the student adopts the role of teacher, and the AI deliberately plays a "confused" or "struggling" learner. The student must explain, clarify, and correct misunderstandings, which reinforces their own mastery while revealing hidden gaps in knowledge.
  - Strengthen mastery by requiring students to teach concepts clearly.
  - Reveal misconceptions through the act of correction.
  - Build metacognitive awareness by reflecting on teaching as a way of learning.

  Relation to General Patterns: **[Persona-Based Pattern]**: AI takes on the persona of a novice or confused learner, shaping the dialogue. **[Flipped Interaction Pattern]**: **The** teaching/learning roles are inverted — the student instructs while the AI asks questions. **[Error Injection Pattern]**: AI may introduce deliberate misunderstandings to give the student opportunities to correct them.
- **Learning Diary Prompts** : Learning Diary Prompts [29] guide students to reflect on what they learned, what was difficult, and what their next steps are. The AI helps structure these reflections, highlights missed elements, and suggests areas for improvement, turning reflection into a regular habit.
  - Develop metacognitive skills through structured reflection.
  - Identify recurring strengths and weaknesses.
  - Encourage continuous improvement and long-term learning habits.

  Relation to General Patterns: **[Tail Generation Pattern]**: Reflection prompts can be added at the end of sessions to ensure consistency. **[Contextual Pattern]**: AI adapts reflections to the subject (SQL, proofs, algorithms). **[Cognitive Verifier Pattern]**: AI checks reflections for accuracy and completeness.
- **Goal-Setting Prompts**: Goal-Setting Prompts [29] support learners in defining specific, realistic objectives for a study session or project. AI helps refine vague intentions into actionable goals and proposes strategies for achieving them, fostering self-regulation.
  - Strengthen planning and organizational skills.
  - Build learner autonomy and ownership of progress.
  - Support realistic and measurable goal-setting.

  Relation to General Patterns: **[Instruction-Based Pattern]**: Breaks goals into actionable steps. **[Constraint-Based Pattern]**: Encourages measurable, time-bound objectives. **[Adaptive Quiz Prompting Pattern]**: Can link goals to personalized practice tasks.
- **Bias Detective**: Bias Detective [30] prompts students to critically evaluate AI outputs for possible bias, error, or missing perspectives. Instead of passively accepting responses, learners are encouraged to challenge validity and fairness, building AI literacy.
  - Build critical evaluation skills for AI-generated content.
  - Raise awareness of fairness, bias, and accuracy.
  - Encourage responsible and reflective AI use.

Relation to General Patterns: **[Fact Check List Pattern]**: Verifies claims systematically against rules or facts. **[Cognitive Verifier Pattern]**: Learners actively test correctness of reasoning. **[Transparency Builder Pattern]**: Complements efforts to expose assumptions.

- **Transparency Builder**: Transparency Builder [30] requires the AI to reveal its reasoning, assumptions, and limitations. Instead of treating answers as authoritative, students see the logic and uncertainty behind them, which fosters critical engagement.
    - Encourage learners to question and validate AI outputs.
    - Strengthen understanding of reasoning processes.
    - Build awareness of AI limitations and uncertainty.

  Relation to General Patterns: **[Chain of Thought Pattern]**: Makes intermediate reasoning steps explicit. **[Fact Check List Pattern]**: Surfaces verifiable claims for review. **[Bias Detective Pattern]**: Works together to expose possible errors or blind spots.

V. Instructor/Design Patterns

- **Curriculum Generator**: Curriculum Generator [31] is a design pattern where AI helps instructors create structured lesson or course outlines. The AI can propose topics, order them logically, and align them with learning outcomes, providing a draft that teachers can refine.
    - Support efficient course and lesson planning.
    - Align content sequencing with learning objectives.
    - Save time by generating adaptable curriculum drafts.

  Relation to General Patterns: **[Instruction-Based Pattern]**: Relies on clear instructions for scope (topics, duration, level). **[Contextual Pattern]**: Uses background info (course goals, learner profile) to shape output. **[Template Pattern]**: Can generate content in predefined formats (week-by-week syllabus).
- **Misconception Map:** Misconception Map [33] leverages AI to surface and organize common errors or misunderstandings students may have about a topic. Instructors can use these insights to design targeted activities and address pitfalls proactively.
    - Anticipate and address student misconceptions.
    - Design targeted lessons to correct recurring errors.
    - Improve assessment and feedback by focusing on known pitfalls.

  Relation to General Patterns: **[Misconception Reveal Pattern]**: Similar logic — focusing on common errors to correct them. **[Data-Driven Pattern]**: AI compiles misconceptions from datasets or student inputs. **[Fact Check List Pattern]**: Provides lists of typical errors for teacher validation.
- **Scaffold Builder:** Scaffold Builder [19] helps instructors break complex learning tasks into smaller, progressive steps. The AI provides incremental stages of difficulty or layered explanations, supporting gradual mastery and reducing cognitive load.
    - Provide structured pathways for student learning.
    - Reduce cognitive overload by sequencing complexity.
    - Support differentiated instruction through tiered scaffolds.

Relation to General Patterns: **[Progressive Elaboration Pattern]**: Builds content layer by layer, from simple to complex. **[Multi-Step Interaction Pattern]**: Structures tasks into sequences of dependent steps. **[Constraint-Based Pattern]**: Can enforce limits (e.g., one new concept per stage).

- **Differentiation Designer:** Differentiation Designer [32] is a pattern where AI adapts tasks, explanations, or assessments to match different learner levels or needs. It enables instructors to create multiple entry points into the same content.
  - Personalize learning materials for diverse students.
  - Promote equity by offering multiple access points to the same concept.
  - Support teachers in managing mixed-ability classrooms.

  Relation to General Patterns: **[Persona-Based Pattern]**: AI can explain the same concept for beginners, intermediates, or experts. **[Constraint-Based Pattern]**: Tailors tasks to different levels of complexity. **[Adaptive Quiz Prompting Pattern]**: Adjusts practice dynamically based on performance.

V. Assessment & Feedback Patterns

- **Rubric-Based Feedback**: Rubric-Based Feedback [34] is a pattern where the AI evaluates student work against a predefined rubric or set of criteria. Instead of providing general comments, the AI gives targeted feedback tied to explicit standards, making evaluation transparent and structured.
  - Provide consistent, structured feedback aligned with learning goals.
  - Increase transparency of evaluation by linking comments to rubric criteria.
  - Help students identify specific areas for improvement.

  Relation to General Patterns: [**Instruction-Based Pattern**]: Relies on clear step-by-step criteria provided in the rubric. [**Constraint-Based Pattern**]: Structures feedback around fixed categories and performance levels. [**Cognitive Verifier Pattern**]: Checks whether the student’s work aligns with specific requirements.
- **Adaptive Test Generation**: Adaptive Test Generation [35] uses AI to create assessments that adjust in real time to student performance. Based on learner responses, the AI can increase or decrease difficulty, change question formats, or introduce targeted challenges, creating a personalized testing experience.
  - Personalize assessment to match learner progress and ability.
  - Support formative testing that identifies strengths and weaknesses.
  - Keep students engaged by maintaining an appropriate level of challenge.

  Relation to General Patterns: [**Adaptive Quiz Prompting Pattern]:** Builds directly on adaptive questioning at the general level. [**Multi-Step Interaction Pattern**]: Structures assessments as sequences that adapt step by step. [**Data-Driven Pattern**]: Uses student performance data to inform the next question.
- **AI as Peer Reviewer**: AI as Peer Reviewer [36] is a pattern where the AI provides feedback in the style of a fellow student rather than an expert teacher. It highlights

strengths, asks questions, and offers constructive suggestions in a more approachable, peer-like tone, encouraging learners to reflect without feeling judged.

- Encourage reflective learning through peer-style dialogue.
- Provide formative feedback in a less intimidating tone.
- Support collaboration and critical thinking by simulating peer evaluation.

Relation to General Patterns: [**Persona-Based Pattern**]: AI adopts the persona of a peer or classmate. [**Flipped Interaction Pattern**]: Encourages dialogue by prompting students to respond to peer-style comments. [**Game-Play Pattern**]: Can simulate structured peer-review activities, like turn-based exchanges.

References

1. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large Language Models are Zero-Shot Reasoners. ArXiv, abs/2205.11916.
2. Brown, T. B., Mann, B., Ryder, N., et al. (2020). Language Models are Few-Shot Learners. arXiv:2005.14165.
3. Wei, J., Wang, X., Schuurmans, D., et al. (2022). Chain of Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903.
4. Zhang, Z., et al. (2022). Automatic Prompt Engineering for Multi-Step Reasoning. arXiv:2210.03059.
5. Shen, X., et al. (2023). Controllable Text Generation for Education. Computers & Education.
6. Ban, Y., Wang, R., Zhou, T., Cheng, M., Gong, B., & Hsieh, C. (2024). Understanding the Impact of Negative Prompts: When and How Do They Take Effect? European Conference on Computer Vision.
7. Ouyang, L., Wu, J., Jiang, X., et al. (2022). Training language models to follow instructions with human feedback. arXiv:2203.02155.
8. White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., & Schmidt, D.C. (2023). A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT. ArXiv, abs/2302.11382.
9. Salesforce. (2024, April 17). What is prompt grounding? — A generative AI tutorial. Salesforce Blog. Retrieved September 30, 2025, from https://www.salesforce.com/blog/what-is-grounding/
10. Pruseth, D. (2023, July 22). Prompt Engineering: The Semantic Filter Pattern. Debabrata Pruseth — Data Science & AI Blog. Retrieved from https://debabratapruseth.com/prompt-engineering-the-semantic-filter-pattern/
11. Pruseth, D. (2023, August 7). Prompt Engineering: Alternative Approach Pattern. Debabrata Pruseth — Data Science & AI Blog. Retrieved from https://debabratapruseth.com/prompt-engineering-alternative-approach-pattern/

12. Mehrotra, P., Parab, A., & Gulwani, S. (2024). Enhancing Creativity in Large Language Models through Associative Thinking Strategies. ArXiv, abs/2405.06715.
13. Mantri, G. (2023, October 5). Prompt Patterns Every Generative AI Developer Must Know. “Tail Generation Pattern” section. Retrieved from https://gauravmantri.com/2023/10/05/prompt-patterns-every-generative-ai-developer-must-know/
14. Kabir, M.R., & Lin, F.O. (2023). An LLM-Powered Adaptive Practicing System. LLM@AIED.
15. WhiteBeard Strategies. (2025, January). Unlocking Cross-Domain Knowledge Transfer With Prompts. WhiteBeard Strategies Blog. Retrieved from https://whitebeardstrategies.com/blog/unlocking-cross-domain-knowledge-transfer-with-prompts/
16. Chi, M.T., Leeuw, N.D., Chiu, M., & LaVancher, C. (1994). Eliciting Self-Explanations Improves Understanding. Cogn. Sci., 18, 439-477.
17. Nicol, D.J., & Macfarlane-Dick, D. (2006). Formative assessment and self-regulated learning: a model and seven principles of good feedback practice. Studies in Higher Education,
18. Smith, W.M., Bruner, J.S., Goodnow, J.J., & Austin, G.A. (1956). A Study of Thinking. American Journal of Psychology, 71, 474. → Classic work on concept learning through contrast.
19. Wood, D.J., Bruner, J.S., & Ross, G.P. (1976). The role of tutoring in problem solving. Journal of child psychology and psychiatry, and allied disciplines, 17 2, 89-100 .
20. Merrill, D. (2012). First principles of instruction. Educational Technology Research and Development, 50, 43-59.
21. Borasi, R. (1994). Capitalizing on Errors as “Springboards for Inquiry”. Journal for Research in Mathematics Education.
22. Fitzgerald, S., et al. (2008). Debugging: Finding, Fixing and Learning. Computer Science Education.
23. Stylianides, A. J. (2007). Proof and Proving in School Mathematics. Journal for Research in Mathematics Education.
24. Rose, D.H., & Meyer, A. (2002). Teaching Every Student in the Digital Age: Universal Design for Learning.
25. Gentner, D. (1983). Structure-Mapping: A Theoretical Framework for Analogy. Cognitive Science.
26. Mazur, E. (1997). Peer Instruction: A User’s Manual. Prentice Hall.
27. Qin, L., Bosselut, A., Holtzman, A., Bhagavatula, C., Clark, E., & Choi, Y. (2019). Counterfactual Story Reasoning and Generation. Conference on Empirical Methods in Natural Language Processing.
28. Schraw, G., & Dennison, R. S. (1994). Assessing Metacognitive Awareness. Contemporary Educational Psychology.
29. Zimmerman, B. J. (2002). Becoming a Self-Regulated Learner. Theory Into Practice.
30. Bender, E. M., & Gebru, T. (2021). On the Dangers of Stochastic Parrots. FAccT Conference.

31. Tyler, R. W. (1949). Basic Principles of Curriculum and Instruction.
32. Tomlinson, C. A. (2001). How to Differentiate Instruction in Mixed-Ability Classrooms. ASCD.
33. Smith, J. P., diSessa, A. A., & Roschelle, J. (1994). Misconceptions Reconceived: A Constructivist Analysis. Journal of the Learning Sciences.
34. Brookhart, S. M. (2013). How to Create and Use Rubrics for Formative Assessment and Grading. ASCD.
35. Van der Linden, W. J., & Glas, C. A. W. (2000). Computerized Adaptive Testing: Theory and Practice. Springer.
36. Topping, K. J. (1998). Peer Assessment Between Students in Colleges and Universities. Review of Educational Research.