

Article | Received 10 May 2026; Revised 31 May 2026; Accepted 5 June 2026; Published 15 September 2026
<https://doi.org/10.55092/phess20260010>

Research on teaching methods of artificial intelligence courses for cultivating computational thinking

Lei Ding*, Mingxing Zeng, Zongshou Li, Yuanqiong Chen

College of Computer Science and Engineering, Jishou University, Jishou, China

* Correspondence author; E-mail: dinglei@jsu.edu.cn.

Abstract: Computational thinking is one of the core competencies for talents in the field of artificial intelligence. However, artificial intelligence courses aimed at cultivating computational thinking face three major predicaments: lack of authentic context, cognitive opacity, and weak motivation. Based on an analysis of scenario-based teaching, scaffolding instruction, and gamification, this paper proposes a collaborative teaching approach characterized by “scenarios continuously present, with scaffolding and games dynamically intervening and promptly returning to the scenario upon goal achievement.” Specifically, scenario-based teaching serves as the underlying cognitive field that runs throughout the entire course, while scaffolding instruction and gamification are introduced on demand into the scenario-based teaching process, targeting cognitive confusion and insufficient motivation respectively. When both cognitive confusion and insufficient motivation occur simultaneously, the principle of “cognitive priority” is followed. Furthermore, taking recommender systems as an example, this paper elaborates on key design points including scenario design covering seven elements, scaffold fading mechanisms, and gamification level design, providing operable methods for cultivating computational thinking in artificial intelligence courses.

Keywords: computational thinking; scenario-based teaching; scaffolding instruction; gamification teaching; collaborative mechanism.

1. Introduction

Computational Thinking(CT) is defined as the thought process of applying fundamental computer science concepts to problem solving, system design, and understanding human behavior [1], it has become one of the core literacies for talent in the intelligent era^[2]. In 2018, the Ministry of Education issued the “Action Plan for Artificial Intelligence Innovation in Higher Education Institutions”, urging universities to incorporate computational thinking cultivation into the core objectives of AI curricula [3]. For instance, the United States has prioritized computational thinking skills such as problem solving and abstraction in AI education [4]. However, after years of pedagogical practice, the academic community has generally recognized three major predicaments:



Copyright©2026 by the authors. Published by ELSP. This work is licensed under Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium provided the original work is properly cited.

- (1) Lack of context: Traditional instruction is detached from real-world engineering scenarios, resulting in unsatisfactory outcomes in thinking cultivation [5].
- (2) Cognitive opacity: Elements of computational thinking such as decomposition and abstraction are inherently tacit, making them difficult to observe and express directly; thus, scaffolding is needed to externalize implicit knowledge [6].
- (3) Weak motivation: There is a lack of organic integration among course content, game design, and thinking practice; gamification elements often remain disconnected from core thinking training, making it difficult for students to perceive their cognitive progression during gameplay, and their participation motivation significantly declines as the course proceeds [7].

However, existing studies tend to focus on a single strategy (e.g., scenario design alone, scaffolding alone, or gamification alone), which makes it difficult to simultaneously address the above three challenges. Therefore, it is necessary to systematically explore the connotations, design essentials, and synergistic mechanisms of the three approaches.

2. The connotation of computational thinking

Computational thinking practice is a collection of methodological elements for problem solving [8]. The U.S. K–12 Computer Science Framework explicitly cultivates practices such as pattern recognition and abstraction [9]. Shute *et al.*, after reviewing 45 papers, proposed a universal computational thinking framework comprising six elements: decomposition, abstraction, algorithm, debugging, iteration, and generalization [10]. Selby *et al.*, by screening the consistency in the use and interpretation of computational thinking elements across various literature, summarized five key elements with relatively high consensus: abstraction, decomposition, algorithm, evaluation, and generalization, where “generalization” is similar in meaning to “generalisation” [11]. Building on Selby’s definitions of computational thinking elements, Shen Shusheng *et al.* further noted that the requirements for computational thinking in the field of artificial intelligence are significantly different from those in general programming courses [12]. Zhang Yi *et al.* pointed out that when dealing with complex engineering problems, debugging and iteration are equally critical [13].

Considering the special demands of artificial intelligence courses for debugging and iteration, as well as the transfer value of generalization ability in engineering practice, this study synthesizes the above viewpoints and adopts seven elements—namely decomposition, abstraction, algorithm, debugging, evaluation, iteration, and generalization—as the core orientation for computational thinking cultivation.

3. Design of teaching methods for artificial intelligence courses

This paper integrates the advantages of scenario design, scaffolding, and gamification instruction to construct a collaborative teaching design characterized by “continuous presence of scenarios, with scaffolding and games dynamically intervening”.

3.1. Scenario-based instruction: a continuously operating cognitive field

Piaget's view [14] grounds CT in authentic scenarios. Shen *et al.* [12] create sequential, relational, or dynamic scenarios by regulating roles, tasks, resources, and events, adapting abstraction, decomposition, algorithm, evaluation, and generalization to AI paradigms. Scenario instruction designs real-world problems for applying these elements. Design principles: (1) set project cycle (e.g., 4 weeks) with real data (e.g., MovieLens) and emergent requirements; (2) scenario runs continuously, with scaffolding/gamification triggered within; (3) solving authentic problems covers decomposition, abstraction, algorithm, evaluation, debugging, iteration, and generalization; (4) generalization is key—students generalize via new datasets/domains in cross-context tasks; (5) flexibly construct sequential, relational, or dynamic scenarios per problem nature. Example: a 4-week video recommendation project with five stages (requirements, modeling, testing, optimization, cross-domain transfer). Table 1's 'Scenario events/problems' and 'Thinking activities' show emergent task contexts and invoked CT elements.

Table 1. Application of scenario-based instruction (using a recommendation system as an example).

CT Element	Scenario Event / Problem	Thinking Activities Provided to Students
Abstraction	Obtain large volumes of heterogeneous data such as browsing, clicks, and ratings	Build the model in the form of a user-item rating matrix, and abstract user preferences into latent vectors.
Decomposition	Simultaneously satisfy the needs of new and existing users as well as popular and long-tail items	Decompose the work into five modules: data preprocessing, user profiling, collaborative filtering, cold start, and evaluation.
Algorithm	New users with no historical data (e.g., cold start)	Adopt recommendation strategies based on high-probability events, such as combining demographic features with content popularity
Debugging	Unexpected recommendations (e.g., recommending childish animated content to adults)	Debug separately for similarity computation, weight allocation, missing data, etc
Evaluation	Recommendation accuracy fails to meet the preset requirements	Combine offline evaluation with online evaluation
Iteration	Accuracy still below the target	Adopt an iterative cycle of "evaluation→diagnosis→adjustment→reevaluation" until the target is met
Generalization	Can the system be transferred to a food-delivery recommendation system	Analyze the similarities and differences between food-delivery data and video data, and derive a general model framework

3.2. Scaffolding instruction: on-demand cognitive support

Grizioti *et al.* [15] define CT as extracting problem structure; Zhu *et al.* [6] note scaffolding concretizes tacit knowledge. Zhang *et al.* [13] designed conceptual scaffolding (via cognitive chains: decomposition→abstraction→debugging, and algorithm design→iterative optimization→algorithm improvement) and procedural scaffolding (full engineering process). These cover abstraction, decomposition, algorithm, debugging, and iteration. To

align with scenario-based instruction, we add generalization and a withdrawal mechanism. Design principles: (1) covers six elements (excluding generalization); (2) scaffolding only aids generalization (e.g., cross-domain tables), with scenario instruction primarily cultivating it; (3) triggered by student request or repeated failures; (4) removal differs: procedural scaffolding goes from full template $\rightarrow$ 80% $\rightarrow$ 50% $\rightarrow$ interfaces $\rightarrow$ independent; conceptual from full diagram $\rightarrow$ hidden info $\rightarrow$ hints $\rightarrow$ independent framework. Table 2 shows specific application.

Table 2. Application of Scaffolding Instruction.

Triggering Scenario	CT Element	Scaffolding Type	Scaffolding Form	Thinking Activities Provided to Students
Unsure how to extract effective features, and confused about ideas	Abstraction	Conceptual scaffolding	Principle diagram	Flowchart for constructing the rating matrix; visualization explanation of latent vectors
Don't know where to start, and uneven decomposition granularity	Decomposition	Procedural scaffolding	Module decomposition template	Three-layer framework of "Input $\rightarrow$ Processing $\rightarrow$ Output"; interface definition templates for the five modules
Unclear understanding of algorithm principles, and unsure which to choose	Algorithm	Conceptual scaffolding	Strategy comparison table	comparison of applicable conditions, advantages, and disadvantages of three recommendation strategies; checklist of cold-start solutions
Don't know how to proceed next; blindly tuning parameters	Debugging	Procedural scaffolding	Debugging checklist	Four-step method for error localization: data input $\rightarrow$ intermediate results $\rightarrow$ similarity computation $\rightarrow$ weight configuration
Unsure how to select evaluation metrics	Evaluation	Conceptual scaffolding	Evaluation metric cards and trade-off matrix	Provides definitions and applicable scenarios for metrics such as precision, recall, coverage, novelty, diversity, etc.
Don't know how to proceed next; blindly tuning parameters	Iteration	Procedural scaffolding	Iteration log template	Standardized log table recording "Evaluation $\rightarrow$ Diagnosis $\rightarrow$ Adjustment $\rightarrow$ New Results"
Transferring to a food-delivery recommendation system; unsure how to analyze similarities and differences	Generalization	Conceptual scaffolding	Transfer comparison table	Blank table for comparing data-format dimensions

3.3. Gamification instruction: On-demand embedded motivation and assessment tool

Garris *et al.*'s IPO model [16] and Zhang *et al.*'s integration of CT, game elements, and AI content [17] allow students to learn via playing/designing games. In AI courses, gamification inserts challenges, feedback, and competition into key thinking nodes to boost engagement and thinking; levels are catalysts, not substitutes, covering abstraction, decomposition, algorithm, pattern recognition, and debugging. Scenario-based design principles: (1) covers six elements (excluding generalization); (2) generalization is only tested (failure sends back to scenario); (3) triggered by decreased participation and increased completion time; (4) removed when participation and scores recover, with final withdrawal based on performance

in subsequent scenario tasks. When gamification instruction is initiated, its specific application is shown in Table 3.

Table 3. Application of gamification instruction.

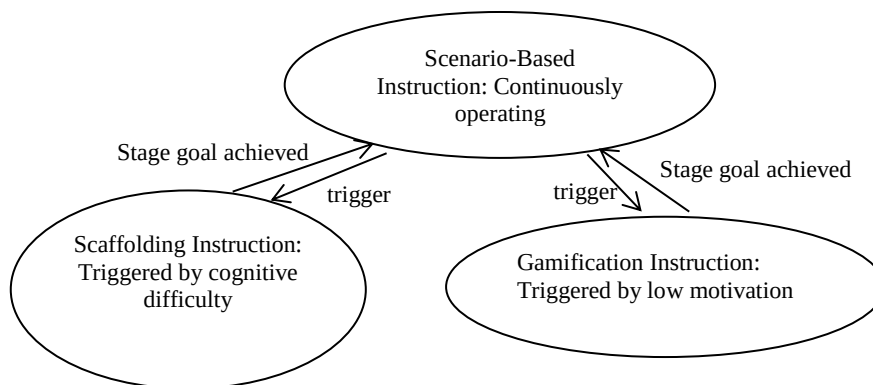
Triggering Scenario	Level	CT Element	Game Mechanism	Thinking Activities Provided to Students
Faced with massive log data and lacking motivation for matrix construction tasks	Data Detective	Abstraction	Progressive Hints	Under step-by-step hints, identify key fields from logs, construct the rating matrix, and gain a sense of achievement
Faced with complex decomposition tasks and lacking confidence and patience	Task Decomposition Master	Decomposition	Time-limited Unlocking	Within a time limit, sort out module boundaries and decompose into 5 modules, experiencing a sense of control
Faced with complex coding implementation and lacking confidence and patience	Algorithm Designer	Algorithm	Challenge-based Mode	Write the relevant core code; pass only after reaching the goal, enhancing self-efficacy
Faced with complex debugging work and lacking patience	Debugging Expert	Debugging	Bug-fixing Competition	Fix preset bugs and earn corresponding points.
Feeling bored with evaluation work	Evaluation Analyst	Evaluation	Leaderboard Race	Select the optimal model within 3 minutes and provide reasons. A leaderboard is set up to stimulate enthusiasm for evaluation through competition
After multiple iterations, feeling dull and uninteresting	Iteration Optimizer	Iteration	Time-limited Parameter Tuning	Find the optimal parameter combination and experience a sense of rhythm through immediate feedback
Performing passively in the generalization process	Transfer Challenger	Generalization	Cross-domain Challenge	Transfer to a food-delivery recommendation system; the system evaluates the transfer effect and stimulates motivation for improvement

3.4. Synergistic mechanism of the three teaching methods

The scenario runs continuously as the foundational field. Scaffolding and games are triggered on demand: scaffolding when cognitive confusion arises, gamification when motivation is low. If both are needed simultaneously, cognition first—scaffolding before gamification. Generalization ability is cultivated solely by scenario instruction; scaffolding provides auxiliary tools, and gamification provides checkpoint tests for generalization. Both are temporary interventions with clear objectives; once achieved, students return to the main scenario track, ensuring continuous presence. Synergies are shown in Table 4 and Figure 1.

Table 4. Synergistic application of the three methods.

Triggering Situation	Judgment Condition	Action	Example	Matching Principle for Scaffolding/ Levels
Smooth cognition and normal motivation	Smooth cognition and normal motivation	Maintain scenario operation	Students steadily advance the recommendation system development; no additional support needed	---
Scenario → Scaffolding	Two consecutive failures or active request for help	Push matching scaffolding	Student fails consecutively in collaborative filtering → push code template	Scaffolding type matches the current scenario stage
Scenario → Gamification	No interaction for 15 minutes and abnormal task completion time	Only push motivation levels matching the current stage	Student remains inactive after completing algorithm design → push the “Algorithm Designer” challenge	Game levels are synchronized with scenario stages
Scenario → Scaffolding → Gamification	Both cognitive difficulty and low motivation are present	Scaffolding first, then gamification	Consecutive failures and no interaction → first provide a debugging checklist; if still no progress, push the Bug-fixing Competition	Scaffolding and game elements are consistent, forming a chain of “cognitive repair → motivation consolidation”
Scaffolding/ Gamification → Scenario	Stage goal achieved	Return to scenario tasks	Complete debugging checklist tasks → return to the recommendation system debugging phase	The return point connects with the intervention point

**Figure 1.** Synergistic relationships among the three teaching methods.

4. Conclusions

This study proposes a collaborative teaching model for AI courses in higher education, characterized by continuous scenarios and dynamically intervening scaffolding and games. Its main innovation, compared with existing studies that focus on single strategies, is a dynamic scheduling mechanism based on students' cognitive and motivational states. Contributions: (1) It analyzes relationships among the three methods: they are not all applied

simultaneously throughout the entire course; scenario-based instruction is foundational, while scaffolding and gamification are scheduled based on states. (2) It defines functions: scenario-based instruction covers all seven CT elements; scaffolding enhances six via cognition and provides auxiliary support for generalization; gamification enhances six via motivation and tests generalization. (3) It provides operable designs and their synergy: scenario covers all seven; trigger and removal conditions for conceptual and procedural scaffolding; trigger contexts, tasks, and removal for gamification levels; scaffolding and games play only auxiliary or testing roles for generalization. Future research can explore automated triggering via learning analytics to reduce teachers' operational burden and verify the model's effectiveness in enhancing CT elements, especially generalization ability

Declaration of generative AI and AI-assisted technologies

During the preparation of this manuscript, the authors used generative AI tools only to improve language and readability. The authors take full responsibility for the content of the manuscript.

Acknowledgments

Funding: This work was supported by the Key Project of the 2022 Hunan Provincial Ordinary Higher Education Teaching Reform Research Project under Grant No. HNJG-2022-0187, the Key Project of the 2025 Hunan Provincial Ordinary Higher Education Teaching Reform Research Project under Grant No. 202502000868, and the Project of the 2022 Hunan Provincial Ordinary Higher Education Teaching Reform Research Project under Grant No. HNJG-2022-0815.

Authors' contribution

Ding is responsible for overall conception and initial drafting; Zeng is responsible for checking the logical structure; Li is responsible for examining logical errors; Chen is responsible for ensuring fluent and coherent wording.

Conflicts of interests

The authors declare no conflicts of interest.

References

- [1] Wing J. Computational thinking. *Commun. ACM*. 2006, 49(3): 33–35.
- [2] Zhu Z, Dai L, Zhao X, Shen S. Cultivating Novel-Quality Talents: A New Mission for Education in the Age of Digital Intelligence. *e-Educ. Res*. 2024, 1: 52–60.
- [3] Wang L, Wang J. New Literacy for Attention in the Era of Artificial Intelligence: Computational Thinking (In Chinese). *Comp. Educ. Rev*. 2021, 43(3): 24–30.

- [4] Han Q, Cai L. Research on the Planning Goals and Characteristics of Artificial Intelligence Education in American Primary&Secondary Schools. *Stud. Foreign Educ.* 2021, 1: 115–128.
- [5] Lou Y, Li Y, Wang Y, Wang Y, Ma J, *et al.* Development Framework and Implementation Strategy for Global Mapping of AI Basic Education Curricula: An Analysis of UNESCO Report of K–12 Artificial Intelligence Curriculum Map. *Distance Educ.* 2022, 3: 3–15.
- [6] Zhu G, Su X, Du J, Chen Q, Xiong B, *et al.* A quasi–experimental study on the influence of different media scaffolds toward physics problem–solving processes. *Interact. Learn. Environ.* 2023, 31(2): 980–993.
- [7] Asbell J, Rowe E, Almeda V, Edwards T, Bardar E, *et al.* The development of students’ computational thinking practices in elementary– and middle– school classes using the learning game, Zoombinis. *Comput. Hum. Behav.* 2021, 115: 1–46.
- [8] Yang X, Chang L. Approaching Artificial Intelligence: Activating Computational Thinking (In Chinese). *Educ. Pract. Res.* 2023, 10: 4–6.
- [9] K–12 Computer Science Framework Steering Committee. K–12 computer science framework. 2016.
- [10] Shute V, Sun C, Asbell J. Demystifying computational thinking. *Educ. Res. Rev.* 2017, 22: 142–158.
- [11] Selby C, Woollard J. Computational thinking: The developing definition. University of Southampton, 2013, pp. 6.
- [12] Shen S, Weng Z. Research on Scenario Design to Promote the Cultivation of Computational Thinking in Artificial Intelligence Course (In Chinese). *Distance Educ. J.* 2023, 41(6): 94–103.
- [13] Zhang Y, Lin Y, Yang Y, Ren X, Gao Z. Effects of Conceptual Scaffolding and Procedural Scaffolding on Students’ Computational Thinking in an Artificial Intelligence Course (In Chinese). *Distance Educ. J.* 2024, 42(5): 102–112.
- [14] Piaget J. The psychogenesis of knowledge and its epistemological significance. In *Piatelli–Palmarini M. Language and learning: The debate between Jean Piaget and Noam Chomsky*. Cambridge: Harvard University Press, 1980. pp. 1–23.
- [15] Kynigos C, Grizioti M. Modifying games with ChoiCo: Integrated affordances and engineered bugs for computational thinking. *Br. J. Educ. Technol.* 2020, 51(6): 2252–2267.
- [16] Garris R, Ahlers R, Driskell J. Games, motivation, and learning: A research and practice model. *Simul. Gaming.* 2002, 33(4): 441–467.
- [17] Zhang Y, Ma J, Zhou P, Liu J, Wang K, *et al.* Research on Gamification Learning in Artificial Intelligence Course to Cultivate High School Students’ Computational Thinking Practices—Take “Challenge Alpha Tic Tac Toe” as An Example (In Chinese). *e–Educ. Res.* 2022, 43(9): 63–72.