ENGENHARIAS, TECNOLOGIA, GESTÃO E TURISMO
ENGINEERING, TECHNOLOGY, MANAGEMENT AND TOURISM
INGENIERÍA, TECNOLOGÍA, ADMINISTRACIÓN Y TURISMO

millenium

Millenium, 2(30)

en

DESIGN INSTRUCIONAL BASEADO EM PROMPTS NO ENSINO DE PROGRAMAÇÃO APOIADO POR INTELIGÊNCIA ARTIFICIAL

PROMPT-BASED INSTRUCTIONAL DESIGN IN ARTIFICIAL INTELLIGENCE-SUPPORTED PROGRAMMING EDUCATION

DISEÑO INSTRUCCIONAL BASADO EN PROMPTS EN LA ENSEÑANZA DE PROGRAMACIÓN APOYADA POR INTELIGENCIA ARTIFICIAL

Mohamad Rani[1] *https://orcid.org/0000-0002-4370-3342*
Vinothini Kasinathan[1] *https://orcid.org/0000-0002-3401-8175*

[1] Universidade Ásia-Pacífico de Tecnologia e Inovação, Kuala Lumpur, Malásia

Mohamad Rani – firdaus@apu.edu.my | Vinothini Kasinathan – vinothini@apu.edu.my

Corresponding Author:
Mohamad Rani
MRANTI Park
57000 – Kuala Lumpur - Malásia
firdaus@apu.edu.my

RECEIVED: 25th February, 2026
REVIEWED: 15th April, 2026
ACCEPTED: 20th May, 2026
PUBLISHED: 12th June, 2026

DOI: https://doi.org/10.29352/mill0230.45667

RESUMO

Introdução: A rápida integração da inteligência artificial (IA) generativa no ensino de programação transformou o uso de prompts de uma técnica de interação em uma prática instrucional relevante. Entretanto, a pesquisa no ensino superior permanece fragmentada.
Objetivo: Sintetizar estudos empíricos sobre instrução baseada em prompts no ensino de programação apoiado por inteligência artificial, analisando conceitualizações, estratégias de implementação, resultados de aprendizagem e lacunas de pesquisa.
Métodos: Realizou-se uma revisão sistemática conforme as diretrizes PRISMA 2020. As buscas nas bases Scopus e Web of Science identificaram publicações entre 2023 e 2025. Vinte estudos atenderam aos critérios de inclusão e foram analisados por meio de síntese temática.
Resultados: O prompting foi conceituado como habilidade técnica, processo iterativo, mecanismo de autorregulação, apoio ao pensamento computacional e recurso instrucional incorporado ao sistema. Andaimes estruturados, como modelos de prompt, progressão orientada e ciclos de depuração, associaram-se a melhorias no pensamento computacional e nos comportamentos de interação. Evidências de ganhos sustentados permanecem limitadas.
Conclusão: A instrução baseada em prompts constitui um campo pedagógico emergente cujo impacto depende do alinhamento com princípios de pensamento computacional e autorregulação. São necessários estudos longitudinais e teoricamente fundamentados.

Palavras-chave: instrução baseada em prompts; inteligência artificial generativa; ensino de programação; pensamento computacional; autorregulação da aprendizagem

ABSTRACT

Introduction: The rapid integration of generative artificial intelligence (AI) into programming education has transformed prompting from a technical interaction technique into a relevant instructional practice. However, research in higher education remains fragmented.
Objective: To synthesize empirical studies on prompt-based instruction in AI-supported programming education, examining conceptualizations, implementation strategies, learning outcomes, and research gaps.
Methods: A systematic review following PRISMA 2020 guidelines was conducted. Searches in Scopus and Web of Science identified peer-reviewed publications between 2023 and 2025. Twenty studies met the inclusion criteria and were analyzed through thematic synthesis.
Results: Prompting was conceptualized as a technical skill, iterative workflow process, self-regulatory mechanism, computational thinking scaffold, and system-embedded instructional feature. Structured scaffolds, including prompt templates, progressive prompting, and debugging cycles, were associated with improvements in computational thinking and interaction behaviors. Evidence for sustained learning gains remains limited.
Conclusion: Prompt-based instructional design is an emerging pedagogical domain whose effectiveness depends on alignment with computational thinking and self-regulated learning principles. Further longitudinal and theory-driven research is needed.

Keywords: prompt-based instruction; generative artificial intelligence; programming education; computational thinking; self-regulated learning

RESUMEN

Introducción: La rápida integración de la inteligencia artificial (IA) generativa en la enseñanza de programación ha transformado el uso de prompts de una técnica de interacción en una práctica instruccional relevante. No obstante, la investigación en educación superior sigue siendo fragmentada.
Objetivo: Sintetizar estudios empíricos sobre instrucción basada en prompts en programación apoyada por inteligencia artificial, analizando conceptualizaciones, estrategias de implementación, resultados de aprendizaje y vacíos de investigación.
Métodos: Se realizó una revisión sistemática conforme a las directrices PRISMA 2020. Las búsquedas en las bases de datos Scopus y Web of Science identificaron publicaciones entre 2023 y 2025. Veinte estudios cumplieron los criterios de inclusión y fueron analizados mediante síntesis temática.
Resultados: El prompting se conceptualizó como habilidad técnica, proceso iterativo, mecanismo de autorregulación, andamiaje del pensamiento computacional y recurso instruccional integrado en el sistema. Las intervenciones estructuradas, como plantillas de prompt, progresión guiada y ciclos de depuración, se asociaron con mejoras en el pensamiento computacional y en los comportamientos de interacción. La evidencia sobre mejoras sostenidas en el aprendizaje es limitada.
Conclusión: La instrucción basada en prompts constituye un campo pedagógico emergente cuyo impacto depende de su alineación con principios de pensamiento computacional y autorregulación. Se requieren estudios longitudinales y teóricamente fundamentados.

Palabras clave: instrucción basada en prompts; inteligencia artificial generativa; enseñanza de programación; pensamiento computacional; autorregulación del aprendizaje

DOI: https://doi.org/10.29352/mill0230.45667

INTRODUCTION

Artificial intelligence (AI), particularly large language models (LLMs), is rapidly reshaping programming education in higher education (Kasneci et al., 2023; Zhai, 2023). AI-powered coding assistants now provide real-time code generation, debugging support, conceptual explanations, and conversational feedback, transforming how students approach problem-solving and algorithmic reasoning (Becker et al., 2023; Finnie-Ansley et al., 2022). While early research has focused primarily on performance outcomes and user perceptions (Kazemitabaar et al., 2024), increasing attention is being directed toward how learner–AI interactions are pedagogically structured (Holmes et al., 2019).

Central to AI-mediated interaction is the prompt, the structured input that guides system responses (Brown et al., 2020). Although often discussed under the technical label of “prompt engineering” (Liu et al., 2023), prompting in educational contexts extends beyond output optimization. In programming education, it increasingly functions as an instructional mechanism (Finnie-Ansley et al., 2022; Kazemitabaar et al., 2024). Carefully designed prompts can guide problem decomposition, scaffold debugging, structure explanations, and encourage evaluation of generated code (Becker et al., 2023). Prompting therefore operates as a form of instructional scaffolding that shapes both learner cognition and the instructional behavior of AI systems (Holmes et al., 2019; Wood et al., 1976).

Programming education has long presented challenges for novice learners, including abstraction difficulties, debugging barriers, and cognitive overload (Lahtinen et al., 2005; Robins et al., 2003). Previous instructional research has addressed these challenges through scaffolding, worked examples, peer instruction, intelligent tutoring systems, and automated feedback (Sweller, 1988; vanLehn, 2011). LLM-based systems introduce a new layer of conversational, adaptive support (Kasneci et al., 2023). However, their pedagogical effectiveness depends not only on technical capability but on how prompting practices are designed and integrated into instruction. Prompt design thus emerges as a critical instructional variable.

Despite the rapid adoption of AI-supported programming tools, research on prompt-based instructional design remains fragmented (Zhai, 2023). Studies vary in how prompting is conceptualized. Some frame it as a learner competency, while others treat it as an embedded instructional scaffold (Becker et al., 2023; Kazemitabaar et al., 2024). Theoretical grounding is inconsistent, and connections to established learning frameworks such as scaffolding, cognitive load theory, and self-regulated learning (SRL) are not always explicitly articulated (Sweller, 1988; Zimmerman, 2002). Methodological approaches also differ substantially in terms of design, duration, and outcome measurement, limiting cumulative knowledge building (Holmes et al., 2019).

Reported prompt-based strategies include decomposition prompts, debugging prompts, explanation prompts, comparative prompts, reflective prompts, and Socratic prompting (Becker et al., 2023; Finnie-Ansley et al., 2022). Although these practices suggest significant pedagogical potential, they are often described descriptively rather than analyzed systematically (Zhai, 2023). Evidence regarding learning outcomes is similarly heterogeneous, ranging from improvements in coding efficiency and confidence (Kazemitabaar et al., 2024) to concerns about overreliance, shallow processing, and reduced independent problem-solving (Becker et al., 2023). Longitudinal evidence on sustained learning transfer remains scarce (Kasneci et al., 2023).

The literature reveals significant conceptual and methodological gaps, underscoring the need for systematic synthesis. Accordingly, this study presents a systematic review of prompt-based instructional design in AI-supported programming education in higher education. It addresses the following research questions:

- How is prompting conceptualized within instructional design research in AI-supported programming education?
- What prompt-based instructional strategies and scaffolding approaches are reported?
- How are prompt-based practices operationalized in higher education programming courses?
- What learning outcomes are associated with these interventions?
- What conceptual, theoretical, methodological, and empirical gaps characterize the current research landscape?

By synthesizing existing scholarship, this review clarifies conceptualizations, categorizes instructional strategies and implementation models, and identifies priorities for future research, providing a structured foundation for evidence-informed prompt-based instructional design.

1. METHODS

This study employed a systematic literature review (SLR) to synthesize empirical research on prompt-based instructional design in AI-supported programming education within higher education. The review followed PRISMA 2020 guidelines to ensure transparent identification, screening, and inclusion procedures (Figure 1).

DOI: https://doi.org/10.29352/mill0230.45667

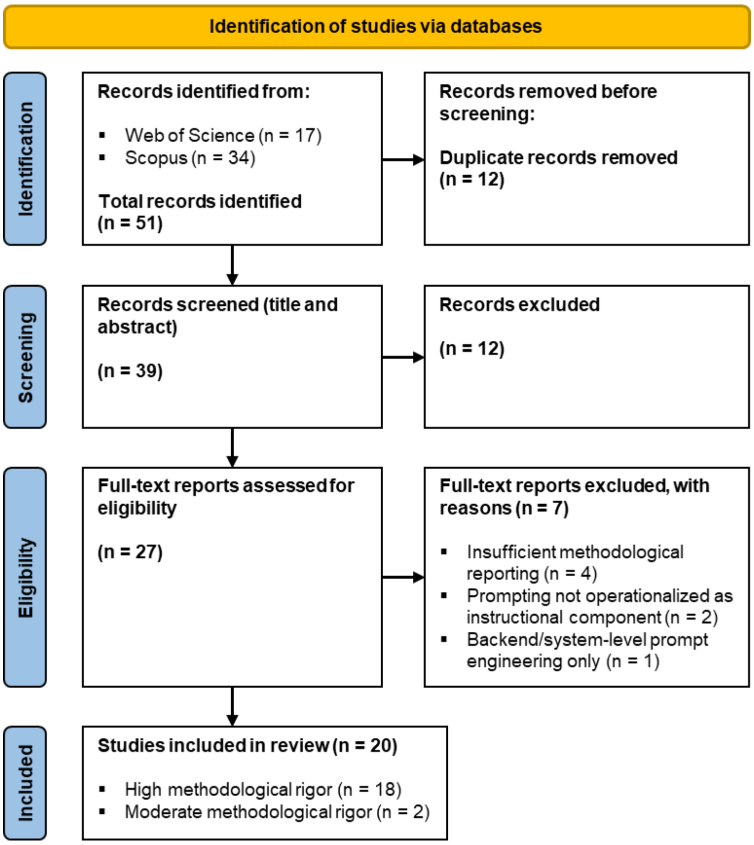

Figure 1 - PRISMA flow diagram of study selection process adapted from Page et al. (2021)

1.1 Search Strategy and Identification

A structured search was conducted in Scopus and Web of Science (WoS), selected for their comprehensive coverage of peer-reviewed research in education and computer science. Consistent with computing research norms, peer-reviewed conference proceedings were included.

Search queries were applied to title, abstract, and keyword fields using Boolean operators. The strategy targeted studies at the intersection of three domains: (a) prompt-related constructs (e.g., prompt engineering, prompt literacy, generative AI prompting), (b) programming and computer science education, and (c) higher education contexts (e.g., undergraduate, university, postsecondary).

The search was limited to English-language journal articles and conference papers published between January 1, 2023 and December 31, 2025. The final database search was conducted in January 2026. The search yielded 51 records (Scopus = 34; WoS = 17). After removal of 12 duplicates, 39 unique studies proceeded to screening.

1.2 Screening and Eligibility

Titles and abstracts were screened against predefined inclusion and exclusion criteria. The eligibility framework was developed to ensure conceptual alignment with prompt-based instructional design in AI-supported programming education and to maintain consistency during study selection. The full set of criteria is presented in Table 1. Following title and abstract screening, 27 studies were retained for full-text eligibility assessment.

Table 1 - Inclusion and exclusion criteria for study selection

Domain	Inclusion Criteria	Exclusion Criteria
Educational Context	Higher education (undergraduate, graduate, postsecondary)	K–12 or informal learning settings
Disciplinary Focus	Programming, coding, computer science education	AI use outside programming education
AI Integration	Use of generative AI or LLM-based systems in instructional activities	Backend model optimization without pedagogical implementation
Prompt Component	Explicit incorporation of prompt-based instructional elements (e.g., structured prompt tasks, scaffolded prompting, prompt literacy instruction)	Use of “prompt” unrelated to LLM interaction
Study Type	Empirical studies reporting implementation data	Editorials, commentaries, purely conceptual papers

DOI: https://doi.org/10.29352/mill0230.45667

1.3 Quality Appraisal

A methodological quality appraisal was conducted to ensure minimum rigor. Five criteria were applied: clarity of objectives, methodological transparency, data source adequacy, analytical rigor, and reporting transparency (Table 2). Each criterion was rated on a three-point scale (1–3), yielding total scores between 5 and 15. Studies scoring ≥9 were retained. Two independent reviewers conducted the appraisal. Discrepancies were resolved through consensus, and inter-rater reliability (Cohen's κ = 0.74) indicated substantial agreement. Seven studies did not meet the quality threshold and were excluded. The final sample comprised 20 studies (18 high rigor; 2 moderate rigor).

Table 2 - Methodological Quality Appraisal Rubric

Criterion		High (3)	Moderate (2)	Low (1)
1.	**Clarity of Objectives**	Clearly stated, aligned with prompt-based instructional design in programming education	Objectives stated but partially aligned	Unclear or not aligned with scope
2.	**Methodological Transparency**	Design clearly described and appropriate for study purpose	Design described but lacks detail	Design unclear or poorly described
3.	**Data Source Adequacy**	Data source appropriate for research question (e.g., transcripts, experiments, surveys) and clearly described	Data source somewhat limited but acceptable	Data source unclear or insufficiently described
4.	**Analytical Rigor**	Analysis methods appropriate and systematically applied; coding or statistical procedures explained	Basic analysis conducted but limited detail	Analysis insufficient or poorly explained
5.	**Reporting Transparency**	Results clearly presented; limitations acknowledged	Partial reporting; some missing clarity	Poor reporting; limited interpretability

1.4 Characteristics of Included Studies

The 20 included studies were published between 2023 and 2025 and employed experimental, quasi-experimental, exploratory, and log-based research designs across introductory and advanced programming contexts. Table 3 summarizes study characteristics and quality ratings.

Table 3 - Characteristics of included studies (N = 20)

Study	Design	Context	AI Tool	Prompt Type	Implementation Mode	Sample (N)	Key Outcomes
(Arora et al., 2025)	Log analysis + survey	Advanced programming	GPT-3.5, GPT-4, Copilot, Gemini	Iterative, decomposition, CoT	Optional AI coursework use	411	Productivity gains; reliance risk
(Gunturi et al., 2025)	Mixed-method	Competitive programming	ChatGPT, Copilot, Claude	Zero-shot, few-shot, CoT	AI-assisted competition	100+	Limited advanced prompting
(Aruleba et al., 2025)	Qualitative longitudinal	Undergraduate Java	ChatGPT	Reflective, reformulated prompts	Structured programming tasks	21	Strategic growth; ethical awareness
(Garg et al., 2025)	Experimental (3-group)	Engineering programming	ChatGPT	Structured prompt training (CLEAR)	Controlled intervention	157	Higher-order gains (Bloom)
(Mutanga et al., 2025)	Mixed-method + prompt corpus	Software development diploma	ChatGPT, Gemini, Claude	Copy-paste, reformulated, multi-question	Naturalistic classroom use	140	Prompt depth linked to learning depth
(Gong et al., 2025)	Experimental + analytics	Undergraduate Python	Kimi LLM	Progressive structured prompts	Prompt-sheet intervention	44	CT gains; interaction shifts
(Alves & Pereira Cipriano, 2025)	Log analysis + survey	CS1 (DSA project)	ChatGPT	Comparative prompting	AI-assisted assignment	69	Partial critical evaluation
(Roy et al., 2025)	Case study (curriculum redesign)	Advanced SE & capstone	GPT-4, Copilot	Zero-shot, CoT, refinement	Curriculum integration	83	Productivity gains; hallucination risk
(Menon, 2025)	Survey (SRL gap analysis)	Undergraduate CS	ChatGPT, Copilot	Planning & verification prompts	Naturalistic use	50	SRL gaps; weak verification
(López-Pernas et al., 2025)	Mixed-method + sequence analysis	Web programming	ChatGPT	Metacognitive/regulatory prompts	Naturalistic AI assignment use	120	Surface regulation; offloading risk
(Rachmat et al., 2025)	Mixed-method intervention	Intro programming	ChatGPT	Structured reflective prompts	Lecture-based intervention	86	Reflection gains; no RT effect
(Mueller et al., 2025)	Experimental Intelligent Tutoring Systems (ITS) comparison	First-year programming	GPT-4o ITS	System-mediated feedback prompts	LLM-based tutoring system	9	History-based feedback preferred
(Sun et al., 2024)	Quasi-experimental	Undergraduate Python	ChatGPT	Structured PbL prompts	Prompt-based learning intervention	30	Deeper interaction patterns
(Tripaldelli et al., 2024)	Pilot experimental	Engineering (Verilog)	ChatGPT-3	Stepwise debugging prompts	Prompt manipulation study	Small pilot	Improved debugging

DOI: https://doi.org/10.29352/mill0230.45667

Study	Design	Context	AI Tool	Prompt Type	Implementation Mode	Sample (N)	Key Outcomes
(Majumdar et al., 2024)	Process mining + analytics	Programming problem-solving	ChatGPT	Embedded SRL prompts	Structured SRL scaffold	42	Stronger epistemic evaluation
(Urhan & Kocadere, 2024)	Qualitative case study	Computational problem solving	ChatGPT-3.5	Pair-programming iterative prompts	Student–AI pair task	3	CT support via guidance
(Wightman, 2024)	Classroom intervention + survey	First-year programming	Multiple LLMs	Requirement refinement prompts	AI-assisted game project	23	Confidence gains; skill gap
(Servín et al., 2024)	Classroom experiment	CS1 (community college)	ChatGPT, Copilot	Prompt programming (zero-shot)	Team-based AI integration	17	Prompting as emerging skill
(Scholl et al., 2024)	Large-scale protocol analysis	Intro programming	ChatGPT-3.5	Diverse iterative prompting	Required AI-assisted assignment	213	High variability; mixed effects
(Maher et al., 2023)	Mixed-method experiment	Introductory Java	ChatGPT, Copilot	Concept clarification & debugging prompts	Between-subjects experiment	3	Mixed engagement; reliance risk

1.5 Methodological Limitations

Although this review followed PRISMA 2020 procedures and applied explicit inclusion, exclusion, and quality appraisal criteria, several methodological limitations should be acknowledged. First, the review was limited to Scopus and Web of Science, which may have excluded relevant studies indexed in ACM Digital Library, IEEE Xplore, ERIC, or other education-focused databases. Second, only English-language peer-reviewed journal articles and conference papers were included, potentially excluding grey literature and non-English research. Third, because the field is recent and rapidly evolving, many included studies used small samples, short intervention periods, or self-report measures, limiting the strength of claims about sustained learning outcomes. Finally, the heterogeneity of study designs, AI tools, prompt strategies, and outcome measures limited the possibility of meta-analysis and required thematic synthesis.

2. DISCUSSION

2.1 From Prompt Engineering to Prompt-Based Instructional Design

A central finding of this review is a conceptual shift in how prompting is framed within programming education research. Early discourse treated prompting primarily as a technical optimization technique for eliciting higher-quality responses from large language models (LLMs) (Liu et al., 2023). Across studies, however, prompting was conceptualized through five overlapping lenses: as a technical literacy, an iterative workflow process, a self-regulatory mechanism, a computational thinking scaffold, and a system-embedded instructional feature.

Framing prompting as a literacy positions it as a workforce-relevant skill, akin to writing precise specifications or managing development tools. Several reviewed studies explicitly emphasized structured prompt crafting and strategy development as competencies to be cultivated (Alves & Pereira Cipriano, 2025; Gunturi et al., 2025). While this perspective highlights skill acquisition aligned with professional practice, it risks isolating prompting from broader pedagogical objectives if reduced to tool proficiency.

Several studies instead conceptualized prompting as an iterative workflow embedded within programming practice (Arora et al., 2025; Scholl et al., 2024). Rather than a one-shot query, prompting was described as a cyclical process of requirement articulation, generation, debugging, refinement, and evaluation. Log-based and process-oriented analyses showed that students' interactions often evolved from broad solution-seeking requests to targeted debugging and optimization (López-Pernas et al., 2025; Majumdar et al., 2024). This framing situates prompting within authentic development cycles and repositions AI interaction as part of sustained reasoning, consistent with prior research on iterative problem-solving in programming (Robins et al., 2003).

A third conceptualization positioned prompting as a self-regulated learning (SRL) mechanism. Prompt sequences were analysed as traces of planning, monitoring, and evaluation behaviours, revealing regulatory patterns such as problem framing and solution verification (López-Pernas et al., 2025; Menon, 2025). In these accounts, prompting becomes an externalization of metacognitive activity and aligns with established SRL frameworks emphasizing planning, monitoring, and reflection (Zimmerman, 2002).

Relatedly, experimental studies treated prompting as a scaffold for computational thinking (CT), using structured templates and progressive strategies to support decomposition and algorithmic reasoning (Garg et al., 2025; Gong et al., 2025; Sun et al., 2024). In these contexts, prompting functions as intentional instructional scaffolding rather than a neutral interaction channel, resonating with classical scaffolding theory (Wood et al., 1976) and contemporary computational thinking research.

Finally, some studies described prompting as a built-in system feature. They integrated LLM-based tutoring systems that partially automate prompt construction (Mueller et al., 2025; Roy et al., 2025). This highlights prompt agency, meaning whether prompts are written by students or generated by the system. It also expands the focus from individual prompt skills to overall instructional design, similar to earlier research on intelligent tutoring systems (vanLehn, 2011). These perspectives indicate a shift from "prompt engineering" toward prompt-based instructional design, where prompting is understood as a pedagogical construct rather than a technical technique.

DOI: https://doi.org/10.29352/mill0230.45667

For conceptual clarity, this review distinguishes among four related but distinct constructs. Prompting refers broadly to the act of formulating inputs to guide AI-generated responses. Prompt engineering refers more narrowly to the technical optimization of prompts to improve model output quality. Prompt literacy refers to learners' broader capacity to formulate, evaluate, revise, and ethically use prompts in support of learning goals. Prompt-based instructional design refers to the pedagogical organization of prompt practices, scaffolds, feedback cycles, and learning tasks within a curriculum. This distinction is important because educational value does not arise from prompt optimization alone, but from the alignment of prompting with disciplinary reasoning, self-regulation, and instructional objectives.

2.2 Instructional Strategies and the Central Role of Scaffolding

Across the reviewed studies, six major categories of prompt-based instructional strategies were identified: structured templates, progressive or step-by-step prompting (including chain-of-thought approaches), iterative refinement and debugging prompts, role- or persona-based prompts, metacognitive/SRL-aligned scaffolds, and system-generated prompting.

Structured templates were among the most common strategies (Gunturi et al., 2025; Tripaldelli et al., 2024). These required learners to specify context, constraints, desired output format, and evaluation criteria. In experimental settings, template use was associated with clearer task articulation and more precise LLM outputs (Garg et al., 2025). Progressive prompting frameworks encouraged learners to request intermediate reasoning steps or structured decompositions, thereby aligning AI interaction with computational thinking processes (Gong et al., 2025; Sun et al., 2024). Such approaches parallel chain-of-thought prompting in generative AI research (Liu et al., 2023).

Iterative refinement and debugging-oriented prompting emerged as the most consistently observed naturalistic practice (Arora et al., 2025; Scholl et al., 2024). Extended prompt sequences often followed patterns resembling software development cycles: articulate requirements, generate code, test output, request corrections, and optimize solutions. Process-analytic studies further documented sequences of testing, verification, and revision (López-Pernas et al., 2025; Majumdar et al., 2024). This iterative model suggests that the pedagogical value of prompting lies not in single interactions but in sustained cycles of evaluation and revision, consistent with established models of programming problem-solving (Robins et al., 2003).

Metacognitive scaffolds explicitly prompted learners to reflect on goals, evaluate solution correctness, or articulate reasoning (López-Pernas et al., 2025; Menon, 2025). Studies employing process analytics found that such scaffolds were associated with increased planning and evaluation behaviours in interaction logs, aligning with theoretical models of self-regulated learning (Zimmerman, 2002). Similarly, role-based or Socratic prompting strategies encouraged dialogic engagement rather than passive answer retrieval (Servín et al., 2024; Urhan & Kocadere, 2024), reflecting interactive learning principles (Chi & Wylie, 2014).

A consistent synthesis finding is that instructional effectiveness depends less on prompt presence and more on scaffold intentionality. Structured, progressive, and metacognitively aligned prompt frameworks were associated with deeper engagement and measurable gains in controlled contexts (Garg et al., 2025; Gong et al., 2025). In contrast, minimally guided or purely student-driven prompting often resulted in one-shot solution requests and limited verification behaviours (Scholl et al., 2024). This pattern reinforces that instructional design mediates the educational impact of learning tools, consistent with cognitive load theory (Sweller, 1988). LLMs provide affordances, but without deliberate scaffolding aligned to domain-relevant cognitive processes, those affordances may not translate into durable understanding.

2.3 Operationalization and Prompt Agency in Higher Education

Prompt-based practices were operationalized across diverse course contexts, including introductory programming (CS1), advanced software engineering, digital system design, and project-based computing courses (Roy et al., 2025; Wightman, 2024). Implementation formats ranged from short-term experimental interventions (Garg et al., 2025; Gong et al., 2025) to semester-long curriculum integrations (Arora et al., 2025).

Most studies relied on student-authored prompting within authentic coursework, frequently analysing chat transcripts or interaction logs to model interaction dynamics (Majumdar et al., 2024; Scholl et al., 2024). A smaller subset implemented LLM-powered tutoring systems that partially automated prompt generation (Mueller et al., 2025; Roy et al., 2025). This distinction highlights the emerging dimension of prompt agency.

When prompting is student-authored, instructional focus shifts toward developing prompt literacy and regulatory competence (Gunturi et al., 2025; Menon, 2025). When prompting is system-mediated, emphasis shifts toward designing environments that embed scaffolds automatically. Each approach carries distinct pedagogical implications. Student-authored prompting may foster transferable AI literacy, but risks inequity if prompt skill varies substantially across learners. System-mediated prompting may reduce cognitive load and provide consistent support, but it can limit learner agency, reflecting ongoing debates in intelligent tutoring systems research (vanLehn, 2011).

Hybrid models, in which structured system scaffolds are gradually withdrawn to foster independent prompting skills, offer a promising approach. This progression aligns with classical scaffolding models that emphasize contingent support and gradual release (Wood et al., 1976).

DOI: https://doi.org/10.29352/mill0230.45667

Operationally, the field remains characterized by short-term, single-course implementations (Garg et al., 2025; Sun et al., 2024). Multi-institutional and longitudinal integrations remain limited, constraining understanding of scalability and sustained impact.

2.4 Learning Outcomes: Promise and Limitations

Evidence regarding learning outcomes was heterogeneous but suggestive of promising directions. The strongest empirical support emerged from controlled interventions targeting computational thinking (CT). Structured and progressive prompt scaffolds demonstrated statistically significant gains in CT sub-dimensions such as algorithmic reasoning and decomposition (Garg et al., 2025; Gong et al., 2025; Sun et al., 2024).

Behavioural shifts were also observed. In scaffolded contexts, learners exhibited increased debugging, verification, and evaluative prompting behaviours compared to minimally guided conditions (Arora et al., 2025; Scholl et al., 2024). Process-analytic studies further documented heightened planning and evaluation sequences when metacognitive scaffolds were embedded (López-Pernas et al., 2025; Majumdar et al., 2024). These findings suggest that prompt scaffolds can influence interaction quality and regulatory engagement, aligning with models of self-regulated learning (Zimmerman, 2002).

Confidence, perceived productivity, and professional readiness gains were frequently reported (Servín et al., 2024; Wightman, 2024), though these outcomes were often measured through self-report instruments rather than standardized performance assessments. Improvements in reflective thinking were supported qualitatively (Menon, 2025), but not consistently demonstrated quantitatively across controlled comparisons.

Several studies identified risks of surface-level engagement and cognitive offloading when scaffolding was absent (Aruleba et al., 2025; Scholl et al., 2024). Overreliance on AI-generated code and limited conceptual reasoning were recurring concerns, echoing broader cautions in generative AI education research (Kasneci et al., 2023). These findings underscore that prompt-based instruction without explicit design mediation may inadvertently reinforce shallow learning behaviours, consistent with cognitive load principles (Sweller, 1988). Overall, prompt-based interventions are most effective when aligned with computational thinking and self-regulation within iterative workflows, though long-term transfer remains underexplored.

The limited longitudinal evidence should therefore be interpreted not only in terms of continued proficiency with specific AI tools, but also in terms of transferability and sustainability. A key question is whether learners can transfer prompt-supported strategies, such as decomposition, debugging, verification, and reflective evaluation, to new programming problems, unfamiliar AI systems, or contexts where AI support is reduced. Sustainable learning gains would be indicated by students' increasing ability to internalize these strategies, critically evaluate AI-generated outputs, and solve problems independently. From this perspective, longitudinal research should examine the gradual movement from externally scaffolded prompting toward autonomous computational reasoning.

2.5 Theoretical Integration and Persistent Gaps

The convergence around self-regulated learning and epistemic action frameworks represents a theoretically significant development (López-Pernas et al., 2025; Majumdar et al., 2024; Zimmerman, 2002). Prompt sequences can function as externalized representations of internal reasoning processes, offering new opportunities for fine-grained learning analytics. Integrating SRL theory with prompt-based instructional design provides a coherent explanatory foundation and situates prompting within established pedagogical traditions rather than as a standalone innovation.

Despite promising developments, several gaps remain. Conceptual definitions of "effective prompting" vary widely across studies, limiting cross-study comparability. Sample sizes are often small, interventions are short-term, and longitudinal evidence is scarce. Heavy reliance on self-report measures constrains claims about durable cognitive gains (Servín et al., 2024; Wightman, 2024). Curriculum-level redesign beyond isolated activities remains underdeveloped.

Methodologically, stronger integration of sequence analytics, standardized assessments, and mixed methods is needed (López-Pernas et al., 2025; Majumdar et al., 2024). Conceptually, clearer definitions of prompting effectiveness that distinguish output quality, behavioural sophistication, and regulatory engagement would support stronger cumulative knowledge building.

2.6 Toward a Coherent Framework for Prompt-Based Instructional Design

Synthesizing the reviewed studies, a layered framework for prompt-based instructional design in AI-supported programming education emerges. As shown in Figure 2, how prompting is conceptualized shapes the design of instructional strategies, which are then implemented in specific course models. The impact of these strategies on learning outcomes depends on scaffold intentionality, meaning how well prompt-based designs align with domain-relevant cognitive processes and self-regulatory support (Wood et al., 1976; Zimmerman, 2002). Meanwhile, ongoing conceptual and methodological gaps highlight priorities for future research.

In this framework, structured scaffolding functions as the central mediating mechanism between prompting practices and learning gains. Evidence from controlled interventions suggests that prompt-based instructional design is most effective when it deliberately targets computational thinking processes and self-regulated learning behaviours within iterative development workflows (Garg et al., 2025; López-Pernas et al., 2025). In this sense, prompting becomes educationally powerful not by virtue of

DOI: https://doi.org/10.29352/mill0230.45667

model capability alone, but through principled instructional alignment, consistent with established learning science principles (Sweller, 1988).

The framework positions prompt-based instruction as an extension of established pedagogical principles enabled by AI. Rather than introducing a wholly new paradigm, prompt-based design reconfigures existing scaffolding, metacognitive, and domain-aligned strategies within AI-mediated learning environments.

Figure 2 should be interpreted as a dynamic rather than linear framework. Prompt-based instructional design involves iterative feedback loops in which learning outcomes and observed risks inform subsequent instructional decisions. For example, evidence of dependency, shallow prompting, cognitive overload, or weak verification should lead instructors to adjust scaffold intensity, prompt structure, task complexity, and opportunities for independent reasoning. As learners develop competence, scaffolds should be gradually faded to support autonomy. In this dynamic model, risk identification is not external to instruction but functions as an ongoing diagnostic mechanism that guides instructional adjustment across the learning trajectory.

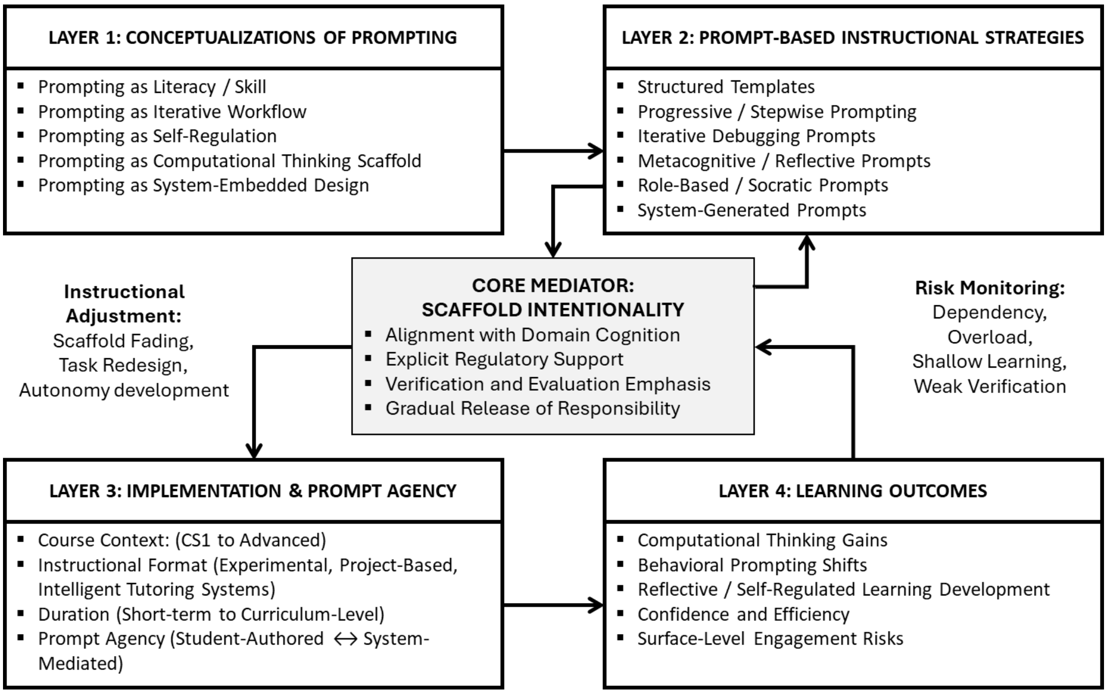

Figure 2 - Dynamic framework for prompt-based instructional design in AI-supported programming education

CONCLUSION

Prompt-based instructional design is evolving beyond technical optimization into a multidimensional pedagogical construct that integrates literacy, workflow practice, and metacognitive scaffolding. While early empirical evidence demonstrates promising effects on computational thinking and interaction quality under structured conditions, broader and longitudinal impacts remain insufficiently established.

Risks associated with AI-supported programming should be treated as instructional design signals rather than merely as limitations. When learners show dependency on AI-generated code, instructors may increase requirements for explanation, testing, and code justification. When cognitive overload appears, prompts may need to be simplified, sequenced, or embedded in worked examples. When shallow learning is observed, tasks should require comparison between AI-generated and student-generated solutions, error diagnosis, and reflection on alternative approaches. Thus, risk monitoring should inform decisions about when to provide scaffolds, when to fade them, and when to require independent problem-solving.

The field now requires theoretical consolidation, methodological rigor, and curriculum-level innovation. By situating prompting within established frameworks of computational thinking and self-regulated learning, future research can move beyond tool-centric experimentation toward principled, evidence-informed instructional design in AI-supported programming education.

DOI: https://doi.org/10.29352/mill0230.45667

AUTHORS' CONTRIBUTION

Conceptualization, M.R.; data curation, M.R. and V.K.; formal analysis, M.R. and V.K.; funding acquisition, M.R. and V.K.; investigation, M.R. and V.K.; methodology, M.R. and V.K.; project administration, M.R.; resources, M.R. and V.K.; software, M.R.; supervision, M.R. and V.K.; validation, V.K.; visualization, M.R.; writing – original draft, M.R.; writing – review & editing, M.R. and V.K.

CONFLICT OF INTERESTS

The authors declare no conflict of interests.

REFERENCES

Alves, P., & Pereira Cipriano, B. (2025). “Give me the code”: Log analysis of first-year CS students’ interactions with GPT. *Proceedings of the International Conference on Computer Supported Education (CSEDU), 2*. https://doi.org/10.5220/0013194400003932

Arora, U., Garg, A., Gupta, A., Jain, S., Mehta, R., Oberoi, R., Prachi, Raina, A., Saini, M., Sharma, S., Singh, J., Tyagi, S., & Kumar, D. (2025). Analyzing LLM usage in an advanced computing class in India. *Proceedings of the 27th Australasian Computing Education Conference (ACE 2025).* https://doi.org/10.1145/3716640.3716657

Aruleba, K., Sanusi, I. T., Obaido, G., Ogbuokiri, B., & Mienye, I. D. (2025). Beyond the prompt: Student strategies, ethical reflections, and learning with ChatGPT in computer science. *Technology, Knowledge and Learning*. https://doi.org/10.1007/s10758-025-09899-7

Becker, B. A., Denny, P., Finnie-Ansley, J., Luxton-Reilly, A., Prather, J., & Santos, E. A. (2023). Programming Is hard - or at least it used to be: Educational opportunities and challenges of AI code generation. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education (SIGSE 2023) 1*. https://doi.org/10.1145/3545945.3569759

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*.

Chi, M. T. H., & Wylie, R. (2014). The ICAP framework: Linking cognitive engagement to active learning outcomes. *Educational Psychologist*, *49*(4). https://doi.org/10.1080/00461520.2014.965823

Finnie-Ansley, J., Denny, P., Becker, B. A., Luxton-Reilly, A., & Prather, J. (2022). The robots are coming: Exploring the implications of OpenAI Codex on introductory programming. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3511861.3511863

Garg, A., Nisumba Soodhani, K., & Rajendran, R. (2025). Enhancing data analysis and programming skills through structured prompt training: The impact of generative AI in engineering education. *Computers and Education: Artificial Intelligence*, *8*. https://doi.org/10.1016/j.caeai.2025.100380

Gong, X., Xu, W., & Qiao, A. (2025). Exploring undergraduates’ computational thinking and human-computer interaction patterns in generative progressive prompt-assisted programming learning. *International Journal of Educational Technology in Higher Education*, *22*(1). https://doi.org/10.1186/s41239-025-00552-y

Gunturi, S. V., Blum, J., & Love, T. (2025). *An evaluation of prompt engineering strategies by college students in competitive programming tasks*. https://doi.org/10.18260/1-2--57608

Holmes, W., Bialik, M., & Fadel, C. (2019). *Artificial intelligence in education: Promises and implications for teaching and learning.* The Center for Curriculum Redesign.

Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., Krusche, S., Kutyniok, G., Michaeli, T., Nerdel, C., Pfeffer, J., Poquet, O., Sailer, M., Schmidt, A., Seidel, T., … Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, *103*, 102274. https://doi.org/10.1016/j.lindif.2023.102274

Kazemitabaar, M., Ye, R., Wang, X., Henley, A. Z., Denny, P., Craig, M., & Grossman, T. (2024). CodeAid: Evaluating a classroom deployment of an LLM-based programming assistant that balances student and educator needs. *Proceedings of the CHI Conference on Human Factors in Computing Systems.* (60), 1-20. https://doi.org/10.1145/3613904.3642773

Lahtinen, E., Ala-Mutka, K., & Järvinen, H.-M. (2005). A study of the difficulties of novice programmers. *ACM SIGCSE Bulletin*, *37*(3). https://doi.org/10.1145/1151954.1067453

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Computing Surveys*, *55*(9). https://doi.org/10.1145/3560815

DOI: https://doi.org/10.29352/mill0230.45667

López-Pernas, S., Misiejuk, K., Oliveira, E., & Saqr, M. (2025). The dynamics of the self-regulation process in student-AI interactions The case of problem-solving in programming education. *Proceedings of 25th International Conference on Computing Education Research (Koli Calling 2025)*. https://doi.org/10.1145/3769994.3770043

Maher, M. Lou, Tadimalla, S. Y., & Dhamani, D. (2023). An exploratory study on the impact of AI tools on the student experience in programming courses: An intersectional analysis approach. *Proceedings of the IEEE Frontiers in Education Conference (FIE)*, 1–5. https://doi.org/10.1109/FIE58773.2023.10343037

Majumdar, R., Prasad, P., & Sane, A. (2024). Mining epistemic actions of programming problem solving with Chat-GPT. *Proceedings of the International Conference on Educational Data Mining*. https://doi.org/10.5281/zenodo.12729902

Menon, P. (2025). Quantifying shortfalls in students' AI-supported programming practices. *Issues in Information Systems*, *26*(1). https://doi.org/10.48009/1_iis_134

Mueller, M., List, C., & Kipp, M. (2025). The power of context: An LLM-based programming tutor with focused and proactive feedback. *Proceedings of the European Conference on Software Engineering Education (ECSEE 2025)*. https://doi.org/10.1145/3723010.3723034

Mutanga, M. B., Msane, J., Mndaweni, T. N., Hlongwane, B. B., & Ngcobo, N. Z. (2025). Exploring the impact of LLM prompting on students' learning. *Trends in Higher Education*, *4*(3). https://doi.org/10.3390/higheredu4030031

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., … Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *The BMJ*, *372.* https://doi.org/10.1136/bmj.n71

Rachmat, A., Watterson, C., & Lundqvist, K. (2025). The impact of chatbots on students' reflective thinking in introductory programming course. *Proceedings of the IEEE Global Engineering Education Conference (EDUCON)*. https://doi.org/10.1109/EDUCON62633.2025.11016417

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. Computer *Science Education*, *13*(2). https://doi.org/10.1076/csed.13.2.137.14200

Roy, N., Horielko, O., & Omojokun, O. (2025). Integrating AI tools in advanced computer science curricula: A case study of course redesign. *Proceedings of the ACM Global Computing Education Conference (CompEd 2025)*, *1*. https://doi.org/10.1145/3736181.3747130

Scholl, A., Schiffner, D., & Kiesler, N. (2024). Analyzing chat protocols of novice programmers solving introductory programming tasks with ChatGPT. *Lecture Notes in Informatics (LNI), P-356*. https://doi.org/10.18420/delfi2024_05

Servín, C., Karichev, N. V., & Pagel, M. (2024). Unfolding programming: How to use AI tools in introductory computing courses. *Proceedings of the Annual Conference on Information Technology Education (SIGITE 2024)*. https://doi.org/10.1145/3686852.3687073

Sun, D., Boudouaia, A., Yang, J., & Xu, J. (2024). Investigating students' programming behaviors, interaction qualities and perceptions through prompt-based learning in ChatGPT. *Humanities and Social Sciences Communications*, *11*(1). https://doi.org/10.1057/s41599-024-03991-6

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, *12*(2), 257–285. https://doi.org/10.1207/s15516709cog1202_4

Tripaldelli, A., Pozek, G., & Butka, B. (2024). Leveraging prompt engineering on ChatGPT for enhanced learning in CEC330: Digital system design in aerospace. Proceedings of the *IEEE Global Engineering Education Conference (EDUCON)*. https://doi.org/10.1109/EDUCON60312.2024.10578705

Urhan, S., & Kocadere, S. A. (2024). Problem-solving through pair-programming: The mediational role of ChatGPT. *Proceedings of the International Conference in Electronic Engineering, Information Technology and Education (EEITE 2024)*. https://doi.org/10.1109/EEITE61750.2024.10654434

vanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. In *Educational Psychologist 46*(4). https://doi.org/10.1080/00461520.2011.611369

Wightman, P. (2024). Twisted games: A first experience of inclusion of AI tools in first year programming classes. *Proceedings of the IEEE Latin American Conference on Computational Intelligence (LA-CCI 2024).* https://doi.org/10.1109/LA-CCI62337.2024.10814750

Wood, D., Bruner, J. S., & Ross, G. (1976). The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry*, *17*(2). https://doi.org/10.1111/j.1469-7610.1976.tb00381.x

Zhai, X. (2023). ChatGPT for next generation science learning. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4331313

Zimmerman, B. J. (2002). Becoming a self-regulated learner: An overview. *Theory into Practice 41*(2). https://doi.org/10.1207/s15430421tip4102_2