Volume 3 Issue 3
www.westlandpublishers.com

ISSN: 3066-3822

WestLand Publishers

Digital Journal of Science (DJS)

Opinion

Building Thinkers, Not Just Coders: Reimagining Computational Thinking in the Age of Artificial Intelligence

Anika Saxena*
The Education University of Hong Kong, China

***Corresponding author:** Anika Saxena, The Education University of Hong Kong, China.

Received Date: 20 May, 2026 ; Published Date: 05 June, 2026

Keywords:
Computational Thinking (CT); Artificial Intelligence (AI); Computational Literacy; Critical Thinking; STEM Education

Citation:
Saxena A* (2026) Building Thinkers, Not Just Coders: Reimagining Computational Thinking in the Age of Artificial Intelligence. Digital J Sci 3(3): 158.

DOI: 10.63592/DJS/158

Abstract

As we move into an era characterised by rapid automation and significantly influenced by artificial intelligence (AI), Computational Thinking (CT) has expanded beyond its origins in computer science (CS) to become a fundamental literacy for the 21st century [1,2] This article argues for a shift in how we approach CT in education—moving away from predefined connections between CT and syntax and coding toward a focus on cognitive adaptability, critical thinking, creative thinking, and problem-solving. Contemporary research from 2022–2026 indicates that the best CT interventions combine ‘plugged’ digital applications (such as robotics and screen-based tools) with ‘unplugged’ conceptual reasoning to develop CT skills through hands-on problem-solving. While the remaining research body is grounded in cross-disciplinary integration of CT, along with equity to make it accessible to all students [3,4] Teaching these skills helps students learn to evaluate and build technology, rather than merely using it.

Introduction

Beyond the Code

For too long, public discourse and educational literature have treated CT and coding as one and the same thing. Such an approach is not only inaccurate but also does an injustice to the field’s cognitive depth, which is grounded in CS but extends far beyond it [4]. At its core, CT is a “habit of mind,” specifically an approach to problem formulation that enables individuals to solve problems using computational thinking and algorithms [5]. It is a way of seeing the world through the lenses of logic, structure, and efficiency. There are two reasons why the aforementioned change is so crucial now. First, the development of AI requires us to rethink the relationship between humans and computing systems [6, 7]. As automated large language models become ubiquitous in our workflows, we will shift from purely manual coding skills to a broader perspective that emphasises systems knowledge and the ethics of AI. This leads to the second reason – “human-AI collaboration” will soon become the norm.

Defining the Core Pillars of Computational Thinking

To understand the power of CT, we must look at its foundational pillars—decomposition, pattern recognition, abstraction, and algorithmic design—through a modern lens.

- **Decomposition** is no longer just about breaking down a CS program; it is about managing the cognitive load of complex socio-technical challenges [4, 8].
- **Abstraction** allows learners to filter out the noise of big data to find the essential signal, a critical skill in the age of AI [4].
- **Algorithmic Design** is the blueprint for automation, whether carried out by a machine or by a human process [15].

Recent studies in early childhood and primary education highlight that when these CT pillars are taught as transferable skills, students show improved reasoning across STEM disciplines [4, 8]. The goal is to build a "transferable cognitive toolkit" that serves a student whether they are debugging a script or analysing a historical trend.

Unplugged Approaches: Lowering the Barrier

One of the most encouraging trends in the last five years is the emergence of "unplugged" strategies, i.e. teaching CT skills without using digital equipment. A recently published meta-analysis demonstrated that unplugged activities are highly effective for presenting logical sequencing and abstraction to children as an easy entry point into CS concepts without involving a computer [9]. What makes unplugged approaches so attractive is their ingrained inclusivity. The approach eliminates the barriers posed by expensive equipment and Internet access, thereby enabling schools in poor areas to teach CT effectively to their students [3]. Additionally, studies conducted in 2023 demonstrated that children do not perceive any substantial difference in the difficulty or importance of CT aptitudes, whether presented in a screen-based learning environment [9] or through physical movement, hands-on games and puzzles [10]. This indicates that the "magic" of computation lies in the logic, not the electronics.

Plugged Approaches: Deepening Engagement through Creation

Although it is essential to begin with unplugged activities, 'plugged approaches' that involve using CS tools suitable for skill transfer, such as ScratchJr, Scratch, robots, and microcontrollers, are needed to establish the feedback loop essential for deep learning. These tools allow students to see their CT thinking come to life. As shown in a comparative study in 2026, the ability of "plugged coding" to provide feedback has been shown to enhance student engagement and make 'debugging' a positive experience for learners, an iterative process [11]. The current trend in the field of plugged computer science education is 'co-design', i.e. designing a project in collaboration with both students and teachers [12]. This approach shifts the teacher from being a "sage on the stage" to a facilitator, which is particularly effective in reducing the intimidation that the technology might have caused the learner [13].

Cross-Disciplinary Integration

The true strength of CT lies in its integration into the broader curriculum areas, moving beyond the confines of a traditional CS department. In mathematics, CT helps students move from numerical skills to algebra; In science, it forms the basis of data modelling and simulation. The latest models of integration of "STEM-CT" point out that as students use technological methods to solve real scientific problems, their understanding of technology and science becomes more sophisticated [4]. The integration ensures that CT is regarded as a universal language of inquiry rather than a specialised set of technical skills.

Equity, Inclusivity, and the Human Element

Any discussion of the future of CT cannot ignore the enduring achievement gap in CS [3]. CT will only help with democratisation if it is culturally responsive. The current literature on CT, as published by Scharff et al. (2025), underscores the need to integrate equity into AI and CT learning. This involves using students' environmental data from their immediate surroundings to inform the creation of algorithms [14] By grounding CT in students' lived experiences, we move away from a 'one-size-fits-all' model and embrace diversity, which is seen as an enabler of innovation rather than a hindrance [3].

The Teacher's Role and Professional Development

Sometimes, what limits progressive CT is not students' interest but rather a lack of confidence in teachers. Many educators, especially non-CS experts, do not feel confident enough to teach students about computational concepts. What is needed is professional development grounded in "conceptual frameworks," not expertise in particular CS languages. Teachers do not need to be expert programmers to teach the logic of an algorithm or the importance of decomposition [3]. Co-design pedagogies allow teachers to develop skills in embedding CT across different subject areas [12].

Challenges and Future Horizons: The Era of AI

As we move towards 2030, one of our greatest challenges and opportunities is AI. We are seeing the emergence of 'AI-enhanced CT', in which students use AI to write code that they must audit, refine and evaluate ethically. This shifts the focus of CS education from the conventional approach of 'writing code' to 'prompt engineering' and 'system auditing.' In a 2025 review article, there have been concerns about the need to teach students about the ethical issues surrounding their development and interactions with AI [15]. These issues include understanding data biases, the privacy implications of automation, and the social impact of AI systems [14]. It is through CT that we develop such literacies. CT is the foundation upon which this broader AI literacy is built.

Conclusion

Building "thinkers" rather than "coders" requires adopting a holistic, intentional, and more comprehensive approach to education. We need to embrace the value of 'unplugged' logic alongside 'plugged' output, while ensuring equal access to such computational thinking (CT) tools for all students, irrespective of their socio-economic status and background [9, 14]. As the boundary between human intelligence and AI continues to blur, the ability to think computationally will be our most enduring asset. By centring equity, creativity, and critical reasoning, we can ensure that the next generation is not just prepared for a digital future shaped by CS but is capable of leading it. Ultimately, fostering a comprehensive understanding of CT will empower individuals to navigate and ethically influence the increasingly automated and data-driven landscapes of the 21st century [16, 17]. This necessitates a re-evaluation of current pedagogical frameworks, moving towards curricula that emphasise adaptive problem-solving and systemic understanding over rote acquisition of technical skills [18].

Financial Support and Sponsorship: None

Conflict of Interest: The author declares no conflict of interest.

References

1. Kong SC, Hsu TC, Zhao J (2025) Proceedings of the 9th International Conference on Computational Thinking and STEM Education in the Era of AI. The Education University of Hong Kong, Conference Proceedings.
2. Rahimi AR, Ghorbani BD (2025) Developing and validating the scale of language teachers' computational thinking competency in Computer Assisted Language Learning (LTCCTCALL): Empowering language teaching by cultivating the heart of the 21st-century digital skill. Asian-Pacific Journal of Second and Foreign Language Education 10(1): 46.
3. Goldberg ER, Darwin T, Esquibel JS, Noble S, Allen, MM, et al. (2023) Contemporary Debates on Equity in STEM Education: Takeaways From a Doctoral Seminar in Equity in STEM Education. Journal of Research in Science Mathematics and Technology Education 6: 69.
4. Govender I (2022) Research Status in Computational Thinking in STEM Education. Advances in Research in STEM Education. IntechOpen.
5. Dewa A (2024) A Bibliometric Review Leveraging the use of Computational Thinking for educational sustainability in STEM subjects: A Framework for Teaching and Learning in the 21st Century. [Review of A Bibliometric Review Leveraging the use of Computational Thinking for educational sustainability in STEM subjects: A Framework for Teaching and Learning in the 21st Century.] Research Square, USA.
6. Ge YT, Mei L, Duan ZH, Li T, Zheng Y, et al. (2025) A Survey of Vibe Coding with Large Language Models. arXiv (Cornell University).
7. Meske C, Hermanns T, Weiden EVD, Loser K, Berger T (2025) Vibe Coding as a Reconfiguration of Intent Mediation in Software Development: Definition, Implications, and Research Agenda. IEEE Access 13: 213242.
8. Lee J, Joswick C, Pole K (2022) Classroom Play and Activities to Support Computational Thinking Development in Early Childhood. Early Childhood Education Journal 51: 457-468.
9. Chen P, Yang D, Metwally AHS, Lavonen J, Wang X (2023) Fostering computational thinking through unplugged activities: A systematic literature review and meta-analysis. IJ STEM Ed 10: 47.

10. YILMAZ T, İzmirli S (2023) Effect of unplugged and plugged coding activities on secondary school students' computational thinking skills. Journal of Educational Technology and Online Learning *6*(4): 1180.
11. García-Marqués ME, Pérez-Suay A, García-Bayona I (2026) Plugged or Unplugged? A Comparative Study of Computational Thinking Development in Early Childhood. Educ Sci 16(2): 333.
12. Sunday AO, Agbo FJ, Suhonen J (2024) Co-design Pedagogy for Computational Thinking Education in K-12: A Systematic Literature Review. Technology, Knowledge and Learning. 30: 63-118.
13. Wahyu FS, Ningrum MA, Malaikosa YML, Kristanto AY, Istiq'faroh N, et al. (2026) Teacher-Facilitated Plugged Coding for Early Childhood Computational Thinking. Technology, Knowledge and Learning 30: 63-118.
14. Scharff C, Cotoranu A, Wautelet Y, Brusseau J (2025) Systematically incorporating equity into design thinking for AI education. AI Magazine 46: e70008.
15. Chick JC (2025) AI-Enhanced Computational Thinking: A Comprehensive Review of Ethical Frameworks and Pedagogical Integration for Equitable Higher Education. Educ Sci 15(11): 1515.
16. Kafai YB (2023) Empowering the Next Generation of Computational Thinkers. CompEd 2023: Proceedings of the ACM Conference on Global Computing Education 1: 3-4.
17. Mäkitalo K, Laru J, Dagienë V (2023) Editorial: Looking ahead: computational thinking in K12 education. Front Educ 8: 1248509.
18. Das S, Mitra A (2024) Enhancing Logic and Problem-Solving Skills in the Digital Age. In M. Fonkam & N. Vajjhala (Eds.), Revolutionising Curricula Through Computational Thinking, Logic, and Problem Solving (pp. 40-59), IGI Global Scientific Publishing, USA.