Review Article ISSN 2767-3901

International Journal of Mathematatic Exploration and Computer Education

From Code to Thinking: Why Every Educator Should Know a Bit of Programming

Heng Chen

Chairman, Professor & Independent Education Researcher,
Fuzhou Bohui AI Technology Co., Ltd., China.

***Correspondence author**
Heng Chen,
Chairman, Professor & Independent Education Researcher,
Fuzhou Bohui AI Technology Co., Ltd.,
China.
.

Submitted : 16 Jul 2025 ; Published : 13 Aug 2025

Citation: Chen, H. (2025). From Code to Thinking: Why Every Educator Should Know a Bit of Programming. *Int J Math Expl & Comp Edu.2*(2):1-2.

Introduction

In the digital age, programming is no longer an exclusive skill reserved for computer scientists. Instead, it has emerged as a core literacy that all educators should acquire. This does not imply that every teacher must become a programming expert; rather, it emphasizes the importance of the computational thinking that underpins programming. Increasingly recognized as the "fourth literacy," alongside reading, writing, and numeracy (Chaabi et al., 2025), computational thinking has been embedded into national curricula across numerous countries as a strategic priority for equipping students with future-oriented skills (Mills et al., 2025). This article argues for the essential role of programming literacy in education by examining its impact from four key dimensions:

1. enhancing educators' cognitive development,
2. fostering students' future competencies,
3. enabling interdisciplinary integration, and
4. aligning with global curriculum reform trends.

Enhancing Educators' Cognitive Development

Programming cultivates rigorous logical reasoning and structured problem-solving skills among educators. These skills mirror the processes involved in designing lesson plans and addressing classroom challenges. Programming requires the decomposition of complex tasks and the development of algorithmic solutions, which are transferable to instructional planning and pedagogical decision-making (Yang, 2025).

Research consistently highlights computational thinking as a foundational 21st-century skill that can be applied across academic disciplines and real-world scenarios (Tongal et al., 2024). For example, an empirical study involving early childhood educators demonstrated that participation in robotics-based programming training significantly enhanced their professional competence and pedagogical confidence (MacCallum, 2025). These educators not only expanded their technological and instructional knowledge but also developed a stronger appreciation for the relevance of computational thinking in teaching.

Furthermore, scholars have proposed extensions to the TPACK (Technological Pedagogical Content Knowledge) framework—such as the (TPAC)2K model—which integrates creative digital competencies into teachers' professional development (Rehman et al., 2025). In this context, acquiring programming skills encourages educators to reflect critically on their instructional practices, enrich their pedagogical content knowledge, and adopt more logical, innovative, and evidence-informed approaches to teaching.

Fostering Students' Future Competencies

Educators' understanding of programming directly influences their ability to cultivate students' future-ready competencies. As societies increasingly demand digital literacy, creative problem-solving and technological fluency, programming education serves as a key enabler. Scholars argue that education systems must go beyond teaching digital tools to embedding the processes of computational thinking, enabling students to understand the science behind technology.

Core elements of computational thinking—such as abstraction, algorithmic reasoning, and debugging—equip students to become creators rather than consumers of technology. When these concepts are embedded into classroom practices, students are more likely to develop critical thinking, problem-solving abilities, and creativity in applying technology. A large-scale international review by Mills et al. (2025) provides strong evidence that participation in programming and computational thinking activities enhances students' academic performance and creative capacity across disciplines.

As artificial intelligence and automation transform the job market, proficiency in programming will become indispensable for many careers. Teachers who are familiar with computational thinking can better guide students toward these emerging demands, preparing them not only for academic success but also for lifelong adaptability and innovation.

Enabling Interdisciplinary Integration

Programming and computational thinking possess high transferability and offer significant potential for interdisciplinary applications. While often associated with computer science or information technology, programming can enrich the teaching and learning processes in mathematics, science, art, and beyond. Through interdisciplinary programming projects, students develop systems thinking and gain a holistic understanding of complex problems.

For instance, a mathematics teacher proficient in coding might introduce functions by having students write simple programs that generate graphs, thereby deepening conceptual understanding and fostering innovation. Programming enables educators to design project-based learning environments that bridge multiple subjects, making learning both engaging and contextually relevant.

Contemporary research on STEM and STEAM education strongly supports this integrative approach. Programming is increasingly viewed as a “universal language” that facilitates interdisciplinary instruction and helps students build a more coherent knowledge framework for solving real-world challenges.

Aligning with Global Education Policies and Curriculum Reform

Globally, education systems are responding to the demands of the digital economy by integrating programming into national curricula. According to international comparative studies, approximately two-thirds of countries now include programming or computational thinking in primary and secondary education standards.

Countries such as the United States, United Kingdom, Finland, Germany, Belgium, and Australia have made programming a compulsory component of K–12 education. Initiatives like “CSforALL” in the U.S. and the UK’s revised computing curriculum reflect the strategic priority placed on preparing students for digital futures.

China has also recently launched significant curriculum reforms, introducing programming education in basic education and investing in teacher training. Research on global STEM teacher development indicates that interdisciplinary innovation and digital competence are emerging as universal trends.

For individual educators, this global shift implies that programming literacy is becoming a baseline professional requirement. Only with a firm grasp of computational thinking and basic programming capabilities can teachers effectively implement new curriculum standards, align with policy reforms, and nurture students’ innovative potential.

Conclusion

The educational value of programming goes far beyond technical proficiency. It represents a transformative cognitive and pedagogical paradigm. Educators who possess programming literacy are better equipped to analyze instructional challenges logically, design innovative teaching strategies, and foster students’ creativity and digital fluency. Moreover, such educators are capable of breaking down disciplinary boundaries and designing learning environments that are interdisciplinary, inclusive, and future-oriented. They are also more responsive to curriculum reforms and policy shifts, assuming leadership roles in modern educational systems.

As educational researchers have emphasized, only by continuously updating their professional knowledge structures and embracing computational thinking can teachers effectively prepare the next generation for the uncertainties of the future. Therefore, it is both timely and necessary to advocate: regardless of subject or grade level, every educator should learn a bit of programming—to expand the boundaries of their own thinking, and to illuminate the learning pathways of tomorrow’s students.

References

1. Chaabi, H., Azmani, A., & Dodero, J. M. (2025). Bridging computational thinking research and practice of pre-service and in-service teachers. *Humanities and Social Sciences Communications, 12*(1), 1–16. DOI: http://dx.doi.org/10.1057/s41599-025-04895-9
2. Mills, K. A., Cope, J., Scholes, L., & Rowe, L. (2025). Coding and computational thinking across the curriculum: A review of educational outcomes. *Review of Educational Research, 95*(3), 581–618. DOI: https://doi.org/10.3102/00346543241241327
3. Yang, W. (2025). A three-phase professional development approach to improving robotics pedagogical knowledge and computational thinking attitude of early childhood teachers. *Computers & Education*, 231, 105282. DOI: https://doi.org/10.1016/j.compedu.2025.105282
4. Tongal, A., Yıldırım, F. S., Özkara, Y., Say, S., & Erdoğan, S. (2024). Examining teachers’ computational thinking skills, collaborative learning, and creativity within the framework of sustainable education. *Sustainability, 16*(22), 9839. DOI: https://doi.org/10.3390/su16229839
5. MacCallum, K. (2025). Integrating computational thinking through digital creation: The (TPAC)2K model. *Teaching and Teacher Education*, 161, 105056. DOI: https://doi.org/10.1016/j.tate.2025.105056
6. Rehman, N., Huang, X., Mahmood, A., Mohammad, N. K., & Zafeer, H. M. I. (2025). Emerging trends and effective strategies in STEM teacher professional development: A systematic review. *Humanities and Social Sciences Communications, 12*(1), 32. DOI: https://doi.org/10.1057/s41599-024-04272-y.