

Introduction: Computational Thinking and Computer Science Special Issue

Craig E. Shepherd and Cecil R. Short, Editors-in-Chief

As computers become more functional and ubiquitous, societies are placing greater emphasis on programming and development skills. Computer science credentials and degree programs have long existed in higher education. Many high schools have also offered computer science courses like coding, computer graphics, game development, and cybersecurity. However, the desire to push computer science training to younger audiences is increasing. Currently, a dozen states require computer science instruction as a prerequisite for high school graduation (Barack, 2025). Many others provide opportunities for computer science experiences within PK-12 curricula. However, computer science topics may intimidate educators and students. Coding languages like Python, C#, and Javascript feel cryptic and take time to learn. Block options like Scratch provide easier entries into coding but still require sustained effort and attention.

Yet, computer science is broader than coding. The processes involved in computer science often involve components of problem solving labeled as computational thinking. These processes involve approaches like problem recognition, decomposition, pattern recognition, abstraction, and algorithmic thinking. These processes can also be introduced in myriad grade levels and subject areas. This special issue includes nine articles that describe various low and high-tech approaches to teach computational thinking and computer science.

The first three articles focus on the use of unplugged and high-tech tools to foster computational thinking. “Puzzling our way into computational thinking” by David Mulder leverages puzzle pieces to help practicing P-8 teachers consider the processes needed to solve a puzzle, recognize patterns, and deconstruct the challenge. “Unplugged and plugged: An introduction to coding for elementary school children” by Jessica Kerr and Theodore Kopcha leverages low-tech tools in a 4-5th grade afterschool program to help students transition to block-based coding. “Integrating CT in science methods:

Advancing practice and pedagogy” by Ugur Kale and Yuahnua Wang moves the focus to preservice teacher preparation where unplugged activities and block coding help students develop computational thinking lessons implemented during practicum experiences.

Computational thinking is relevant to various subjects. “The plot thickens: Literacy through computational thinking” by Bataul Alkhateeb and Eiman Abushihab and “Artificial intelligence-enhanced digital storytelling: Empowering young creators in a summer STEM camp” by Bulent Dogan and Amani Itani describe how computational thinking aligns with secondary and elementary English Language Arts curricula. “Transformations throwdown: Extending mathematics knowledge with Assemblr” by Brian Johnson et al. aligns computational thinking with middle-school mathematics standards. “Scratch day: Hands-on computational thinking activities for youth and adult learners” by Ayanna Perkins et al. used a day-long event to help learners of all ages gain awareness and experience with block coding.

The remaining two articles “Scratch Encore: Creating and sustaining culturally responsive computer science education” by Rasha Alkhateeb et al. and “Integrating CT/CS into a teacher education program: A year-long PD” by Irene Bal et al. focus on sustained curricula. All articles in this issue focus on in-person instruction. However, the articles about sustained curricula also included hybrid and online activities or guidance about how to make activities suitable for online and hybrid delivery.

These articles are by no means exhaustive on the subject. However, we hope they will provide ideas, perspectives, and activities to inspire educators who are considering how to leverage computational thinking and computer science concepts in their classrooms.

GET INVOLVED

The JTILT editors and editorial board continue to examine how they can disseminate materials that are immediately useful, simplify processes, and increase their relevance to all teaching practitioners. As JTILT continues into its fourth year, we hope to see greater emphases on international and underrepresented voices. We welcome your voice! Reach out to the editorial team to get involved. Share your ideas and suggestions. Submit an article. Volunteer to review. JTILT is growing! We openly invite you to become part of that growth.

REFERENCES

Barack, L. (2025, May 14). *Significant momentum for computer science curriculum*. K-12DIVE.
<https://www.k12dive.com/news/significant-momentum-computer-science-curriculum-ai/748103/>

SHARING & MODIFICATION PERMISSIONS

Unless otherwise noted, this article and its resources are published under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International license](https://creativecommons.org/licenses/by-nc-sa/4.0/):



You can freely share the article and its resources if you indicate the original authors, identify the Creative Commons license, and use them non-commercially.

You may also make and share modifications by:

- [Identifying the original authors](#).
- Using the resources non-commercially.
- Licensing modifications under the CC BY-NC-SA 4.0 license (and including a link to it).
- Indicating what modifications were made.