

ISTE STANDARDS FOR EDUCATORS

Computational Thinking Competencies

Leaders and educators around the world have the enormous responsibility of preparing all students for success in a future where computing power underpins every aspect of the systems we encounter in our daily lives. Ensuring that every student understands and is able to harness the power of computing to improve their success in their personal, academic or professional lives is an ambitious goal. The **ISTE Standards for Educators: Computational Thinking Competencies** is intended to help all educators contribute to making that goal a reality.

In 2019, ISTE will release educator standards specifically for computer science discipline teachers in collaboration with the Computer Science Teachers Association. The Computational Thinking Competencies, however, focus on the educator knowledge, skills and mindsets to integrate computational thinking (CT) across the K-12 content areas and with students of every age. These competencies augment and hone in on the competencies embedded in the ISTE Standards for Students and the ISTE Standards for Educators.

Computational thinking is a powerful ingredient for solving ambiguous, complex and open-ended problems by drawing on principles and practices central to computer science (CS). CT is at the core of computer science and a gateway to sparking student interest and confidence in learning computer science. In these competencies, we use the definition of CS found in the K12 CS Framework, "the study of computers and algorithmic processes, including their principles, their hardware and software designs, their [implementation], and their impact on society," (*Tucker et. al, 2003, p. 6*), and describe computational thinking as involving designing solutions that leverage the power of computing.

Similarly to how technology is used by educators to deepen content area learning while building digital learning skills, teachers can integrate CT practices in their instruction to introduce computational ideas. This will enhance student content knowledge and build confidence and competence in CT. By integrating computational thinking into the classroom, educators can support students to develop problem-solving and critical-thinking skills, and empower them for success as CS learners and computational thinkers.

ISTE recognizes that the CS concepts framed in current standards and frameworks are not only new to students, but educators as well. Standard 1. Computational Thinking (Learner) is not an expectation of current

knowledge, but instead the beginning of a road map to help educators identify strengths and weaknesses, and seek out professional development opportunities to increase their mastery.

This document is not a one-size-fits-all list of expectations, but a recognition that competencies present different opportunities for growth and goal-setting for educators. Educators are doing powerful work to integrate CT across other disciplines to enable students to learn, use and apply CS concepts and CT practices across different contexts. ISTE seeks to help educators recognize where this work is already happening, identify opportunities to make these connections more explicit, and develop new ways to deepen student learning in CS, using computational thinking to drive that work.

1. Computational Thinking (Learner)

Educators continually improve their practice by developing an understanding of computational thinking and its application as a cross-curricular skill. Educators develop a working knowledge of core components of computational thinking: such as decomposition; gathering and analyzing data; abstraction; algorithm design; and how computing impacts people and society. Educators:

- a. Set professional learning goals to explore and apply teaching strategies for integrating CT practices into learning activities in ways that enhance student learning of both the academic discipline and CS concepts.
- b. Learn to recognize where and how computation can be used to enrich data or content to solve discipline-specific problems and be able to connect these opportunities to foundational CT practices and CS concepts.
- c. Leverage CT and CS experts, resources and professional learning networks to continuously improve practice integrating CT across content areas.
- d. Develop resilience and perseverance when approaching CS and CT learning experiences, build comfort with ambiguity and open-ended problems, and see failure as an opportunity to learn and innovate.
- e. Recognize how computing and society interact to create opportunities, inequities, responsibilities and threats for individuals and organizations.

2. Equity Leader (Leader)

All students and educators have the ability to be computational thinkers and CS learners. Educators proactively counter stereotypes that exclude students from opportunities to excel in computing and foster an inclusive and diverse classroom culture that incorporates and values unique perspectives; builds student self-efficacy and confidence around computing; addresses varying needs and strengths; and addresses bias in interactions, design and development methods. Educators:

- a. Nurture a confident, competent and positive identity around computing for every student.
- b. Construct and implement culturally relevant learning activities that address a diverse range of ethical, social and cultural perspectives on computing and highlight computing achievements from diverse role models and teams.
- c. Choose teaching approaches that help to foster an inclusive computing culture, avoid stereotype threat and equitably engage all students.
- d. Assess and manage classroom culture to drive equitable student participation, address exclusionary dynamics and counter implicit bias.
- e. Communicate with students, parents and leaders about the impacts of computing in our world and across diverse roles and professional life, and why these skills are essential for all students.

3. Collaborating Around Computing (Collaborator)

Effective collaboration around computing requires educators to incorporate diverse perspectives and unique skills when developing student learning opportunities, and recognize that collaboration skills must be explicitly taught in order to lead to better outcomes than individuals working independently. Educators work together to select tools and design activities and environments that facilitate these collaborations and outcomes. Educators:

- a. Model and learn with students how to formulate computational solutions to problems and how to give and receive actionable feedback.
- b. Apply effective teaching strategies to support student collaboration around computing, including pair programming, working in varying team roles, equitable workload distribution and project management.
- c. Plan collaboratively with other educators to create learning activities that cross disciplines to strengthen student understanding of CT and CS concepts and transfer application of knowledge in new contexts.

4. Creativity & Design (Designer)

Computational thinking skills can empower students to create computational artifacts that allow for personal expression. Educators recognize that design and creativity can encourage a growth mindset and work to create meaningful CS learning experiences and environments that inspire students to build their skills and confidence around computing in ways that reflect their interests and experiences. Educators:

- a. Design CT activities where data can be obtained, analyzed and represented to support problem-solving and learning in other content areas.
- b. Design authentic learning activities that ask students to leverage a design process to solve problems with awareness of technical and human constraints and defend their design choices.
- c. Guide students on the importance of diverse perspectives and human-centered design in developing computational artifacts with broad accessibility and usability.
- d. Create CS and CT learning environments that value and encourage varied viewpoints, student agency, creativity, engagement, joy and fun.

5. Integrating Computational Thinking (Facilitator)

Educators facilitate learning by integrating computational thinking practices into the classroom. Since computational thinking is a foundational skill, educators develop every student's ability to recognize opportunities to apply computational thinking in their environment. Educators:

- a. Evaluate and use CS and CT curricula, resources and tools that account for learner variability to meet the needs of all students.
- b. Empower students to select personally meaningful computational projects.
- c. Use a variety of instructional approaches to help students frame problems in ways that can be represented as computational steps or algorithms to be performed by a computer.
- d. Establish criteria for evaluating CT practices and content learning that use a variety of formative and alternative assessments to enable students to demonstrate their understanding of age-appropriate CS and CT vocabulary, practices and concepts.

For more information, contact standards@iste.org. ISTE Standards for Educators: Computational Thinking Competencies, ©2018, ISTE® (International Society for Technology in Education), iste.org. All rights reserved.