WHAT IS COMPUTATIONAL THINKING AND WHAT DOES IT HAVE TO DO WITH K-12 EDUCATION?

What is computational thinking (CT)?

CT refers to the thought processes involved in defining a problem and its solution so that the solution can be effectively carried out by a computer.[1] People don't need computers to engage in CT, but CT can capitalize on the power of computers to solve a problem. Computational artifacts are the things that computer scientists design, build, and study.[2] Computational artifacts are developed by using an iterative process that includes reflecting on and modifying a plan, taking into account key features, time and resource constraints, and user expectations.[3]

Computational thinking helps build these skills:

DECOMPOSITION

Recognizing and defining computational problems and their component parts.

ABSTRACTION

Extracting common features, recognizing patterns, and reducing unnecessary detail.

ALGORITHMIC THINKING

Designing a set of steps to accomplish a specific task.

DEBUGGING AND EVALUATION

Testing and refining computational artifacts and ensuring the solution is the best fit for the problem.[4]

These competencies relate to critical thinking and problem solving skills across subject matter, underscoring how the concepts of computing can be combined with other fields of study to assist in problem-solving.

What does CT have to do with computer science (CS)?

CT is a way of describing the many problem solving skills involved in CS, including those needed to design, develop, and debug software. However, CS is more than just skills, it also includes concepts about the Internet, networking, data, cybersecurity, artificial intelligence, and interfaces. CT can be relevant beyond CS, overlapping with skills also used in other STEM subjects, as well as the arts, social sciences, and humanities.

Why is CT important?

CT can apply the problem-solving techniques of CS to other subjects. For example, CT is established as one of the Science and Engineering Practices in the Next Generation Science Standards and can be found in several math state standards. CT encourages us to use the power of computing beyond the screen and keyboard.

Photo Courtesy of Riverside Unified School District

[1] Cuny, J., Snyder, L., and Wing, J. M. (2010). "Demystifying Computational Thinking for Non-Computer Scientists".
[2] Turner, R. (2020). Computational Artifacts: the Things of Computer Science. Philosophy & Technology, 33(2), 357-367.
[3] California Computer Science Content Standards (2018) & K–12 Computer Science Framework (2016).
[4] Ibid.

ADVANCING EQUITY IN COMPUTER SCIENCE

Importantly, highlighting CT skills offers an opportunity to consider equity while bringing a social justice perspective to the forefront of CS. First, CT centers the problem-solving skills that are at the heart of CS, promoting its integration with other subject areas, and exposing more students to the possibilities of CS. Secondly, CT, calls upon us to examine the limitations and opportunities of computational artifacts as they are being developed, by whom, and for what purposes and to think critically about the ways in which these artifacts impact society and our diverse communities of users.

What are examples of projects that foster computational thinking in the classroom?

Middle School Earth Sciences

Students locate the epicenter of an earthquake.
https://teacherswithguts.org

Middle School Math

Students debug a program that is designed to calculate the volume of any size rectangular prism.
https://ctrl.education.illinois.edu/ltec/lessons

High School Biology

Students use computational models to understand the connection between natural selection, and speciation.
https://ct-stem.northwestern.edu

Where can I find other resources on computational thinking?

- **K-12 CS Framework's Core Practices** (2016):
  This framework for CS for K-12 places CT at the core of its practices, and is what the California standards are based on.
  https://k12cs.org/navigating-the-practices

- **Computing at School** (Csizmadia et al., 2015):
  Part of the British Computing Society, Computing at School put forth resources to assist teachers in the UK in embedding CT in their classrooms.
  https://community.computingatschool.org.uk/resources/2324/single

- **Operational Definition of Computational Thinking** (ISTE & CSTA, 2011):
  This is one of the earliest definitions of CT for educators, and noteworthy for its inclusion of certain dispositions as being essential for effective CT.
  https://cdn.iste.org/www-root/ct-documents/computational-thinking-operational-definition-flyer.pdf

- **New frameworks for studying and assessing the development of computational thinking** (Brennan & Resnick, 2012) (pdf):
  The developers of Scratch divide CT into concepts, practices, and perspectives, and focus on the expressive and creative nature of computing.
  https://dam-prod.media.mit.edu/x/files/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf

- **Computational Thinking Competencies** (ISTE, 2018):
  Instead of focusing solely on standards for students, ISTE compiled a set of knowledge, skills, and mindsets needed for educators to be successful in integrating CT across the K-12 content areas and grade bands.
  https://www.iste.org/standards/computational-thinking

- **Bebras**:
  Bebras began as an international competition to promote CT for students, regardless of programming experience. It is now increasingly being used as a form of CT assessment.
  https://www.bebras.org