K-12 BLUEPRINT

What is Computational Thinking?

In short, Computational Thinking is a systematic way for students to learn complex problems. It uses the language, logic, and constraints of computers to apply computational methods to problems.

Computational thinking teaches students to think like a computer program so that whatever solution the student is formulating can be designed efficiently.

Many students find it hard to stay focused on the same task for a long period of time. Computational thinking breaks down a complex problem into smaller, more comprehensible tasks. It's a highly interactive methodology where students learn by doing: gaining hands-on practice in building problem-solving applications. It also allows for immediate feedback, which allows students to link this feedback to the immediate task at hand (rather than after the fact, which is often the case with more traditional forms of teaching).

Above all, computational thinking helps students to recognize that some tasks that might seem very difficult at first are actually very doable. This is a vital life-skill that can increase both capabilities and confidence across a wide range of disciplines. Computational thinking also emphasizes collaboration and student-centered engagement.

Computational Thinking Elements

- **Decomposition:** breaking down a problem into a smaller, more manageable, hierarchy of problems.
- **Abstraction:** creating a logical representation of a problem and its physical, spatial, or temporal constraints. Once logically represented, it is possible to manipulate, study, or interact with the constraints as well as the logical model representation of the problem.
- **Algorithmic Representation:** a step-by-step or procedural representation of the solution to a problem.
- **Pattern Recognition:** This is the process of looking for similarities across sets of problems, or similarities within the problem itself. Pattern Recognition can be applied to the solution as well.

For computational thinking to become a fundamental academic discipline in the classroom, teachers must demonstrate an openness and excitement for its pillars—both in their teaching as well as their professional development—and teach in an engaging way that is relevant to the experiences and interests of all students. Teacher and students must also have access to the technology, resources and tools that best support teaching and learning in the foundational pillars of computational thinking.

Benefits of Computational Thinking

Computational Thinking is used extensively across most if not all industries such as financial, genomics, energy, automotive, space, and the arts. With Computational Thinking, students develop higher- order thinking skills such as critical thinking, analytical thinking, evaluation, and application. Also, students learn to solve problems in creative and innovative ways.

A Brief History of Computational Thinking

While the phrase "computational thinking" is credited to computer science professor Jeannette Wing from a 2006 Association for Computing Machinery ("ACM") essay, computational thinking itself goes back much farther.

The history of computational thinking as a concept dates back to the 1950s, while its specific components are older. Computational thinking is typically considered an amalgamation of abstraction, data representation, and the logical organizing of data. The phrase is predated by terms popularized by computing pioneers Donald Knuth and Alan Perlis such as algorithmizing, procedural thinking, and computational literacy. In 1974, Knuth wrote: "A person does not really understand something until he can teach it to a computer." Perlis, likewise, maintained that programming should be integrated into liberal higher education.

Computational thinking was first referenced by mathematician Seymour Papert as "procedural thinking" in 1980—and again in 1996—while researching computer usage, claiming that computational thinking helped define the relationship between a problem and its solution. While working at MIT (the Massachusetts Institute of Technology), Papert worked to bring computational thinking into K-12 education by helping to create a pedagogical foundation.

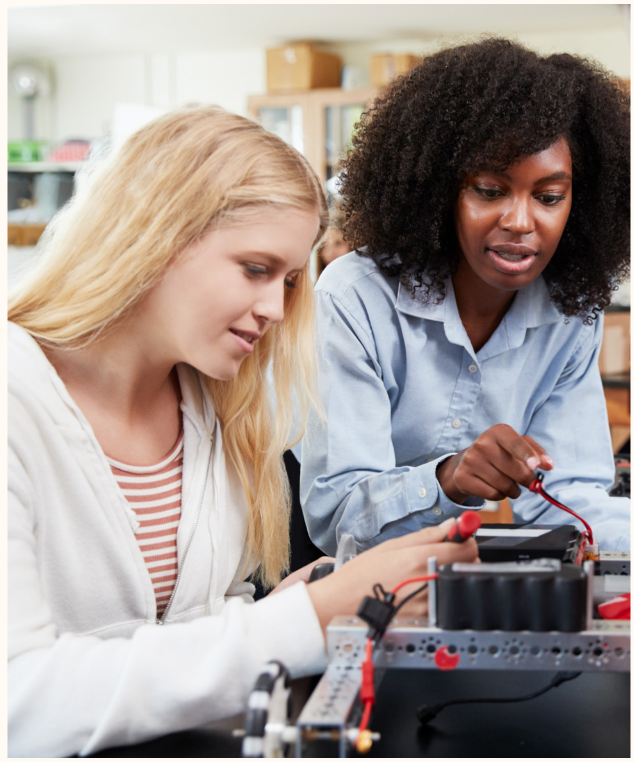
Though the term was already known throughout the computational science world, Jeannette Wing introduced the phrase to a broader academic audience in 2006. Wing's description of computational thinking was as a problem-solving tool that reduces problems to smaller parts, abstracts out some concerns, and chooses appropriate presentations.

The essay positioned computational thinking as a fundamental skill for everyone—not simply scientists and professors within the computer science and academia worlds—and maintained that computational ideas should be integrated into other school subjects. By stressing the educational value of informatics (a branch of information engineering) for all learners, Wing had built upon the principles and beliefs of various computer scientists before her.

Wing later evolved her definition of computational thinking to "the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent".

It was also a case of the right place at the right time in that STEM education was top of mind with many schools and even political leaders wishing to push forward computer science objectives. Computational thinking became something of a rallying cry, with organizations such as CSTA (Computer Science Teachers Association) establishing their own computational thinking frameworks. CSTA's framework, for example, combining data organization and analysis, abstractions, and algorithmic thinking.

Technologist brothers Stephen and Conrad Wolfram both wrote about computational thinking in 2016, resulting in the launch of the Wolfram Computational Thinking Initiative. The initiative offers programs and resources to improve computational thinking skills among students everywhere. Stephen Wolfram presents computational thinking as a defining feature of the future: "It's an incredibly important thing to be teaching to kids today.... Computational thinking is going to be needed everywhere. And doing it well is going to be a key to success in almost all future careers."



Technology has become a vital aspect of our modern lives. And, to efficiently design, program and control this technology, human-computer interaction must therefore be as efficient as possible.

And while computational thinking is not a new idea, it is one that needs more attention and support—both financially and philosophically—so that it can successfully become an intrinsic part of the modern classroom.