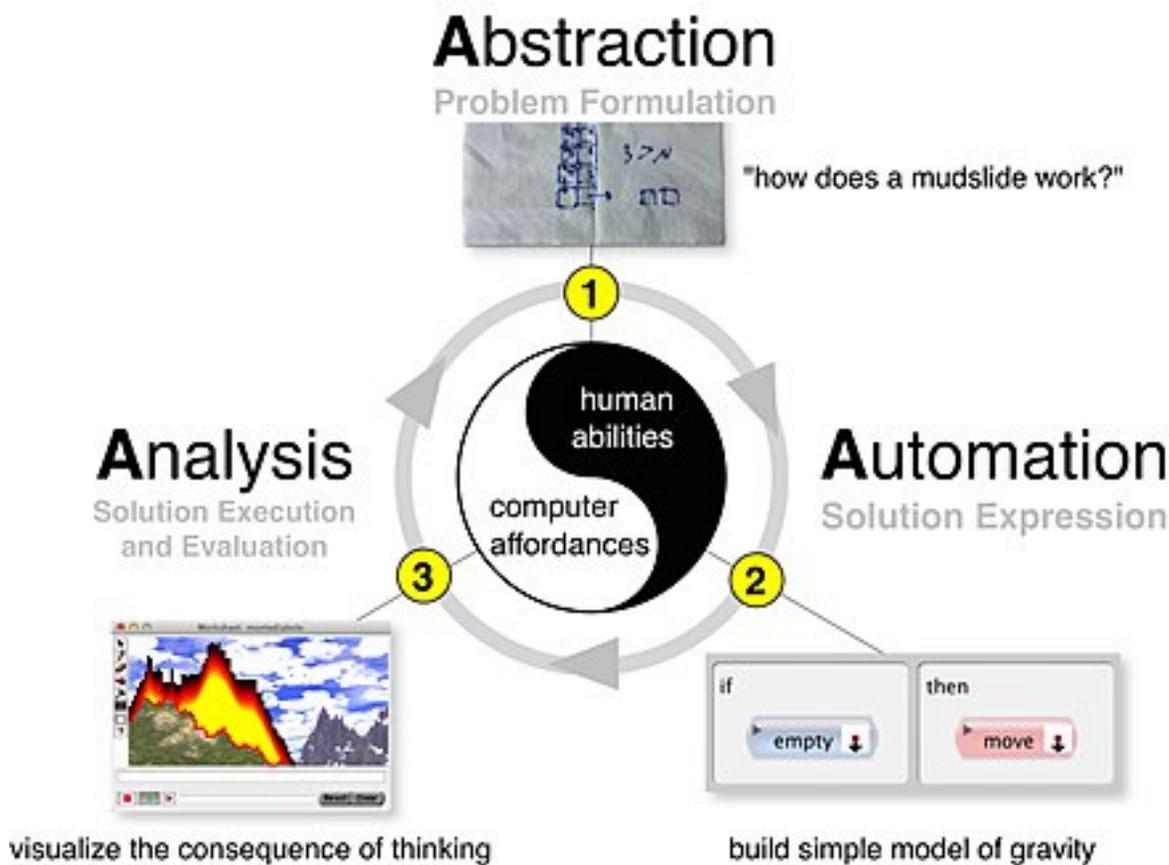


Computational thinking



Three stage process (the "AAA CT process") describing Computational Thinking

By one of the many definitions, **computational thinking** is the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer—human or machine—can effectively carry out.^[1]

Another characterization of computational thinking is an iterative process based on three stages (captured by the AAA Computational Thinking Process figure to the right):

1. **Abstraction:** Problem formulation;
2. **Automation:** Solution expression;
3. **Analyses:** Solution execution and evaluation.

The history of computational thinking dates back at least to the 1950s but most ideas are much older.^[2] The term *computational thinking* was first used by [Seymour Papert](#) in 1980^[3] and again in 1996.^[4] Computational thinking can be used to algorithmically solve complicated problems of scale, and is often used to realize large improvements in efficiency.^[5]

Overview

The characteristics that define computational thinking are [decomposition](#), [pattern recognition](#) / [data representation](#), [generalization/abstraction](#), and [algorithms](#).^{[6][7]} By decomposing a problem, identifying the variables involved using data representation, and creating algorithms, a generic solution results. The generic solution is a generalization or abstraction that can be used to solve a multitude of variations of the initial problem.

The phrase *computational thinking* was brought to the forefront of the [computer science](#) community as a result of an [ACM](#) Communications article on the subject by [Jeannette Wing](#). The article suggested that thinking computationally was a fundamental skill for everyone, not just computer scientists, and argued for the importance of integrating computational ideas into other disciplines.^[8]

Computational thinking in K-12 education

Jeannette Wing envisioned computational thinking becoming an essential part of every child's education.^[8] However, since her article (published in 2006) integrating computational thinking into the K-12 curriculum has faced several challenges including the agreement on the definition of computational thinking.^{[9][10]} Currently Computational Thinking is broadly defined as a set of cognitive skills and problem solving processes that include (but are not limited to) the following characteristics:^{[10][11]}

- Using abstractions and pattern recognition to represent the problem in new and different ways
- Logically organizing and analyzing data
- Breaking the problem down into smaller parts
- Approaching the problem using programmatic thinking techniques such as iteration, symbolic representation, and logical operations
- Reformulating the problem into a series of ordered steps (algorithmic thinking)
- Identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps

and resources

- Generalizing this problem-solving process to a wide variety of problems

Current integration computational thinking into the K-12 curriculum comes in two forms: in computer science classes directly or through the use and measure of computational thinking techniques in other subjects. Teachers in Science, Technology, Engineering, and Mathematics ([STEM](#)) focused classrooms that include computational thinking, allow students to practice [problem-solving](#) skills such as [trial and error](#) (Barr, et al., 2011).^[12] [Valerie Barr](#) and Chris Stephenson describe computational thinking patterns across disciplines in a 2011 [ACM Inroads](#) article^[9] However [Conrad Wolfram](#) has argued that computational thinking should be taught as a distinct subject.^[13]

Center for Computational Thinking

[Carnegie Mellon University](#) in [Pittsburgh](#) has a Center for Computational Thinking. The Center's major activity is conducting PROBEs or **PROBLEM**-oriented **Explorations**. These PROBEs are experiments that apply novel computing concepts to problems to show the value of computational thinking. A PROBE experiment is generally a collaboration between a computer scientist and an expert in the field to be studied. The experiment typically runs for a year. In general, a PROBE will seek to find a solution for a broadly applicable problem and avoid narrowly focused issues. Some examples of PROBE experiments are optimal kidney transplant logistics and how to create drugs that do not breed drug-resistant viruses.^[14]

Computational Thinking Resources

There are a handful of on-line institutions which provide curriculum, and other related resources to build and strengthen pre-college students with Computational Thinking, Analysis and Problems Solving. One prominent one is the [Carnegie Mellon Robotics Academy](#). It offers training sessions for both pre-college students, as well as teachers. CMU's programs exercise

[instructional scaffolding](#) methods via [engineering process](#). There is also another online site named [legoengineering.com](#).^[15] offering similar resources.

As far as a physical facility, in Central New Jersey, there is a small institution, named [Storming Robots](#), offering technology programs to Grade 4 to 12 with an emphasis on Algorithmic and Computational Thinking via robotics projects throughout the school year. Students may follow its road map^[16] starting from Grade 4 until they graduate to college.

Criticism

The concept of Computational Thinking has been criticized as too vague, as it's rarely made clear how it is different from other forms of thought.^{[2][17]}

Some computer scientists worry about the promotion of Computational Thinking as a substitute for a broader computer science education, as computational thinking represents just one small part of the field.^[18]

Others worry that the emphasis on Computational Thinking encourages computer scientists to think too narrowly about the problems they can solve, thus avoiding the social, ethical and environmental implications of the technology they create.^{[19][2]}

Computational Thinking and 4 Cs of 21st Century Learning

The 4 Cs of 21st century consist of Communication, Critical Thinking, Collaboration, and Creativity. The fifth C could be Computational Thinking which entails the capability to resolve problems algorithmically and logically. It includes tools that produce models and visualize data^[20].

Computational Thinking is applicable across subjects beyond Science, Technology, Engineering, and Mathematics (STEM) which include the social sciences and language arts. Students can engage in activities where they identify patterns in grammar as well as sentence structure and use models for studying relationships^[21].

Since its inception, the 4 Cs have gradually gained acceptance as vital

elements of many school syllabus. This development triggered a modification in platforms and directions such as inquiry, project-based, and more profound learning across all K-12 levels. Many countries have introduced computer thinking to all students. The United Kingdom has CT in its national curriculum since 2012. Singapore calls CT as “national capability”. Other nations like Australia, China, Korea, and New Zealand embarked on massive efforts to introduce Computational Thinking in schools^[22]. In the United States, President Barack Obama created this program, Computer Science for All to empower this generation of students in America with the proper computer science proficiency required to flourish in a digital economy^[23]. Computational Thinking means thinking or solving problems like computer scientists. CT refers to thought processes required in understanding problems and formulating solutions. CT involves logic, assessment, patterns, automation, and generalization. Career readiness can be integrated into learning and teaching environments in multiple ways^[24].

- [Computer-Based Math](#)

References

1. ^ Wing, Jeannette (2014). "[Computational Thinking Benefits Society](#)". *40th Anniversary Blog of Social Issues in Computing*.
2. ^ a b c Tedre, Matti; Denning, Peter J. (2016). "The Long Quest for Computational Thinking". [Proceedings of the 16th Koli Calling Conference on Computing Education Research \(PDF\)](#).
3. ^ Papert, Seymour. *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc., 1980.
4. ^ [Papert, Seymour](#) (1996). "[An exploration in the space of mathematics educations](#)". *International Journal of Computers for Mathematical Learning*. **1**. [doi:10.1007/BF00191473](#).
5. ^ Computational thinking:
 - o [Repenning, A.; Webb, D.; Ioannidou, A.](#) (2010). "Scalable game design and the development of a checklist for getting computational thinking into public schools". *Proceedings of the*

41st ACM technical symposium on Computer science education - SIGCSE '10. p. 265. [doi:10.1145/1734263.1734357](https://doi.org/10.1145/1734263.1734357).

[ISBN 9781450300063](https://www.amazon.com/dp/9781450300063).

- [Guzdial, Mark \(2008\). "Education: Paving the way for computational thinking" \(PDF\). *Communications of the ACM*. **51** \(8\): 25. \[doi:10.1145/1378704.1378713\]\(https://doi.org/10.1145/1378704.1378713\).](#)
 - [Wing, J. M. \(2008\). "Computational thinking and thinking about computing". *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*. **366** \(1881\): 3717. \[Bibcode:2008RSPTA.366.3717W\]\(https://doi.org/10.1098/rsta.2008.0118\). \[doi:10.1098/rsta.2008.0118\]\(https://doi.org/10.1098/rsta.2008.0118\). \[PMC 26961023\]\(https://pubmed.ncbi.nlm.nih.gov/26961023/\).](#)
 - [Center for Computational thinking at Carnegie Mellon \[cmu.edu\]\(http://cmu.edu\)](#)
 - [Exploring Computational Thinking, Google.com](#)
 - [How to Teach Computational Thinking by Stephen Wolfram, Stephen Wolfram Blog, September 7, 2016.](#)
 - [Conrad Wolfram: Teaching kids real math with computers on YouTube, TED talk 2010-11-15](#)
 - [What is Computational Thinking? CS4FN at CS4FN](#)
 - [The Sacramento Regional CPATH Team has created a Think CT website with information on CT developed by this NSF-funded project. Archived 31 August 2011 at the Wayback Machine.](#)
 - [A short introduction to Computational Thinking by the Open University](#)
6. [^] ["Introduction to computational thinking". BBC Bitesize. Retrieved 25 November 2015.](#)
 7. [^] ["Exploring Computational Thinking". Google for Education. Retrieved 25 November 2015.](#)
 8. [^] ^a ^b [Wing, Jeanette M. \(2006\). "Computational thinking" \(PDF\). *Communications of the ACM*. **49** \(3\): 33. \[doi:10.1145/1118178.1118215\]\(https://doi.org/10.1145/1118178.1118215\).](#)
 9. [^] ^a ^b [Barr, Valerie; Stephenson, Chris \(2011\). "Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?". *Acm Inroads*. **2**.](#)
 10. [^] ^a ^b [Grover, Shuchi; Pea, Roy \(2013\). "Computational Thinking in](#)

- K–12 A Review of the State of the Field". Educational Researcher. 42. doi:10.3102/0013189x12463051.*
11. ^ Stephenson, Chris; Valerie Barr (May 2011). "Defining Computational Thinking for K-12". *CSTA Voice*. 7 (2): 3–4. [ISSN 1555-2128](#). "CT is a problem solving process..."
 12. ^ Barr, David; Harrison, John; Leslie, Conery (2011-03-01). "Computational Thinking: A Digital Age Skill for Everyone". *Learning & Leading with Technology*. 38 (6): 20–23. [ISSN 0278-9175](#).
 13. ^ [Computational thinking is the code to success](#) Conrad Wolfram, The Times Educational Supplement
 14. ^ [PROBE Experiments](#)
 15. ^ ["LEGO Engineering"](#). Retrieved 30 Dec 2013.
 16. ^ ["Roadmap for learning path"](#). Retrieved 30 Dec 2013.
 17. ^ Jones, Elizabeth. ["The Trouble with Computational Thinking"](#) (PDF). ACM. Retrieved 30 November 2016.
 18. ^ Denning, Peter J. (1 June 2009). "Beyond computational thinking". *Communications of the ACM*. 52 (6): 28. [doi:10.1145/1516046.1516054](#).
 19. ^ Easterbrook, Steve (2014). ["From Computational Thinking to Systems Thinking: A conceptual toolkit for sustainability computing"](#). *Proceedings of the 2nd international conference ICT for Sustainability*. [doi:10.2991/ict4s-14.2014.28](#).
 20. ^ ["Should the 4 Cs of 21st century skills make room for one more?"](#). *Education Dive*. Retrieved 2018-06-11.
 21. ^ ["http://educationworld.in//EWIssue.aspx"](#). *EducationWorld.in*. Retrieved 2018-06-11. External link in |title= ([help](#))
 22. ^ ["The 5th 'C' of 21st Century Skills? Try Computational Thinking \(Not Coding\) - EdSurge News"](#). *EdSurge*. 2018-02-25. Retrieved 2018-06-11.
 23. ^ ["CSforALL"](#). *www.csforall.org*. Retrieved 2018-06-11.
 24. ^ ["Future-forward: How to incorporate the 5th 'C' of 21st Century learning"](#). *eSchool News*. 2017-10-25. Retrieved 2018-06-11.

External links

- [The Robotics Academy, Carnegie Mellon University](#)
- [Rise of the Machines, Larry Wasserman](#)
- [What is Computational Thinking ?](#)
- [How to Teach Computational Thinking](#)
- [Cos'è il Pensiero Computazionale, In Italian](#)
- [L'alfabetizzazione al Pensiero Computazionale, In Italian](#)