Computer Science and K–12 Mathematics

Our Position

NCSM believes that both computer science and mathematics are essential domains of study for 21st century students. We acknowledge that there are areas of overlap between the domains, which should be used to further advantage, but also recognize that there are important differences. Both involve computations and the development of thinking processes and practices, but computer science does not target the specific mathematics to the breadth and depth expected in state and provincial mathematics standards. Given this, NCSM believes that computer science and mathematics courses are not equivalent and takes the position that a computer science course that does not intentionally target a full course of mathematics standards should not count as a mathematics course. NCSM also cautions that the use of a computer science course as a substitution for a mathematics requirement must never adversely impact preparation for college or career readiness.

Background

In 2016 several organizations worked together to create and publish the K–12 Computer Science Framework.[1] The purpose was to establish a common vision for use in the development of computer science standards and curriculum and to support the teaching and implementation of computer science pathways. Although there is no single definition of computer science, the framework describes computer science as "the study of computers and algorithmic processes, including their principles, their hardware and software designs, their applications, and their impact on society."[2] Computer science involves much more than programming, as indicated by the five core concepts in the K–12 Computer Science Framework:

1. Computing systems

   Devices, hardware and software, and troubleshooting.

2. Networks and the internet

   Network communication and organization, and cybersecurity.

3. Data and analysis

   Collection, storage, visualization and transformation, and inferences and models.

4. Algorithms and programming

   Algorithms, variables, control, modularity, and program development.

5. Impacts of computing

   Culture, social interactions, and safety, law, and ethics.

In practice, there are at least two types of classes under the computer science umbrella. Consider the two

1 K–12 Computer Science Framework, 2016.

2 Tucker, 2006, as cited in the K–12 Computer Science Framework, 2016, p. 13.

computer science courses offered by the College Board for Advanced Placement (AP): AP Computer Science Principles and AP Computer Science A. The Computer Science Principles course focuses on the "fundamentals of computing, including problem solving, working with data, understanding the Internet, cybersecurity, and programming" while the Computer Science A course focuses on the "fundamentals of programming and problem solving using the JAVA language."[3] The first is aimed at students going into a variety of majors while the second targets students heading toward computer science or other STEM fields.

Courses that serve as a survey of computer science, such as AP Computer Science Principles, may align to the core concepts, but they do not teach or apply mathematical topics to the breadth and depth of state and provincial mathematics standards, such as the Common Core State Standards for Mathematics (CCSS-M).[4] This is reasonable as mathematics is not the primary goal of such a course. Some mathematics may be taught and applied when studying *Data and analysis* or *Algorithms and programming*, but the mathematics may be minimal or not aligned to grade-level mathematics standards. Students in the AP Computer Science Principles course, for example, will at some point "use models and simulations to represent phenomena"[5] or "[e]xtract information from data to discover and explain connections or trends,"[6] but nothing is stated about the level of mathematics that is involved or required.

Other computer science courses, such as AP Computer Science A, focus on programming. Programming, a term sometimes used interchangeably with coding (though with debate),[7] requires students to think logically and communicate with precision. Programming includes a broad range of skills, activities, and proficiencies, including such things as problem solving, analysis, data organization and encoding, code writing, verification/testing, and computational thinking. Yet these courses, as the above, do not focus primarily on mathematics. Although it may seem that the connection to the breadth and depth of mathematics content in such courses is stronger, it still is not enough to warrant equating a computer science class as a mathematics class. In discussing computer science courses, including the AP Computer Science A course, the National Council of Teachers of Mathematics (NCTM) Emerging Issues Committee noted the lack of mathematics in the courses:

> "While acknowledging exceptions, it seems fair to say that even the highly regarded high school computer science courses mentioned above teach very little new mathematics. They may fill in gaps, such as teaching function composition, recursion, iteration, sets, and some other discrete mathematics, but most of the mathematical content in these courses has been encountered by students in middle school together with many of the of Algebra I topics, a few geometry topics, and statistics topics relative to the CCSS-M"[8]

LeadCS.org notes that the College Board provides a matrix that shows overlap of the AP Computer Science A course to the CCSS-M. Yet, "[d]espite this overlap, AP Computer Science A is still a computer science

[3] College Board (n.d.).

[4] National Governors Association Center for Best Practices & Council of Chief State School Officers, 2010.

[5] College Board, 2017, p. 17.

[6] Ibid., p. 19.

[7] LeadCS.org, 2015a.

[8] NCTM Emerging Issues Committee, 2015, as cited in LeadCS.org, 2015b.

course and not a mathematics course, and as such does not align to the same extent as AP Calculus and AP Statistics."[9] To help navigate questions about courses that should count toward high school mathematics graduation requirements, NCTM's *Catalyzing Change in High School Mathematics* explains such courses should address mathematics standards (including statistics) and:

- require clarity and precision in reasoning;
- maintain the integrity of the mathematical standards;
- are part of a coherent mathematical learning progression (i.e., they are courses that prepare students to continue their study of mathematics; they are not dead-end courses); and
- approach the mathematics in an instructionally balanced way that includes attention to conceptual understanding, procedural fluency, problem solving, and mathematical reasoning and critical thinking practices.[10]

Discussion

Mathematics educators have long seen the value in utilizing aspects of computer science to support the learning of mathematics. Programming languages such as Logo and its derivative programming environment, *Turtle Math,* have been used with elementary age students to develop conceptual understanding of geometric concepts through programming activity.[11] Developing this type of understanding is about more than just creating figures with a computer. Understanding develops from the act of thinking about creating figures using the available set of commands.[12] These efforts at infusing aspects of computer science into mathematics instruction went far beyond the integration of technology into the classroom as they used computational thinking to develop and solidify mathematical understandings.

Computational thinking, an important theme throughout computer science, is defined in the K–12 Computer Science Framework as "the thought processes involved in expressing solutions as computational steps or algorithms that can be carried out by a computer."[13] The importance of computational thinking extends beyond computer science as *using mathematics and computational thinking* is one of the Science and Engineering Practices in both the National Research Council's K–12 Science Framework and the Next Generation Science Standards.[14] The development of computational thinking in K–12 mathematics is a fruitful pathway to consider as the field moves forward. Among many possibilities, mathematics educators may consider building conceptual understanding through so-called low-floor high-ceiling programming environments.[15] Particular programming languages also allow for effective instruction in areas outside of geometry, such as fractions.[16] Additionally, students and teachers could explore applications of computer

9 LeadCS.org, 2015b.

10 NCTM, 2018, p.82.

11 Clements & Meredith, 1994, Clements & Battista, 1989; Clements & Battista, 1990.

12 Clements & Battista, 1992.

13 K–12 Computer Science Framework, 2016, p.68.

14 National Research Council, 2012, National Research Council, 2013.

15 Grover & Pea, 2013.

16 Harel & Papert, 1990.

science and mathematics such as animation through the use of programming environments and real-world contexts.[17] There is significant potential for computer science and coding to increase student understanding of important mathematics, develop productive practices, and introduce students to computational thinking in a supportive, concept-based environment, all without sacrificing focus on relevant and meaningful mathematical goals and without adding additional, non-mathematics material to the mathematics curriculum.

While computer science is not a subset of mathematics and mathematics is not a subset of computer science,[18] there can, *and should,* be influence in both directions. Developers of mathematics curriculum should look for opportunities to support both the computer science framework and standards, while developers of computer science curriculum should look for opportunities to support strong mathematics standards. Teachers in the two domains should find ways to work together, as students will benefit when this happens. However, we wish to be clear: ***Ultimately, mathematics and computer science are separate domains of learning.*** High school computer science courses focus on teaching computer science principles to students, and thus have computer science-related instructional goals. Mathematics courses have a similar mandate, albeit with a mathematics focus—i.e., there are mathematical instructional standards for these courses.

NCSM is in agreement with NCTM's position on Computer Science and Mathematics Education—that is, "a computer science course should be considered as a substitute for a mathematics course graduation requirement only if the substitution does not interfere with a student's ability to complete core readiness requirements in mathematics."[19] In systems that require three mathematics courses or fewer to graduate (beginning with Algebra I, or equivalent), substituting a computer science course for a mathematics course will likely have a negative impact on a student's long-term college or career goals. No matter the rigor of the proposed computer science course, there is simply no way to ensure that students' mathematical readiness for college or careers is not compromised.

In systems that require four courses of high school mathematics for graduation (beginning with Algebra I, or equivalent), great care should be taken in substituting computer science for a mathematics requirement. The substitution may be "unlikely to adversely affect readiness,"[20] but the reality is that if students wish to attend a four-year postsecondary institution they must be fully prepared to meet the entrance requirements of those institutions. Oftentimes this level of readiness requires advanced study of algebraic concepts and four years of high school mathematics.

Efforts to substitute computer science courses for mathematics courses have the potential to limit access to important mathematics. Decisions about course substitutions should be accompanied by an analysis that describes the impact on access and expectations, and should describe which students may be adversely affected. Decision makers should consider how such substitutions may *limit* further student progress in both mathematics and computer science. Additionally, attention should be given to the reality that students who skip a year or two of mathematics in high school will encounter greater challenge if they later choose to select a college or career path that requires additional mathematics. NCSM believes that *all* students must have access to high-quality mathematics instruction and be held to high expectations for mathematics achievement.

[17] Grover & Pea, 2013.

[18] LeadCS.org, 2015b.

[19] NCTM, 2016, p. 1

[20] Ibid., p.1.

Future Actions

Teacher Actions

- Advocate for the maintenance of mathematics *and* computer science as vital requirements for career and college readiness.
- Be prepared to explain the differences between mathematics and computer science courses.
- Make clear to leaders and other stakeholders the potential for connections between mathematics and computer science, such as through computational thinking, to effectively support students' mathematical learning.
- Learn about and look for opportunities to engage students with accessible programming environments to support student understanding of relevant mathematics content.

Leader Actions

- Advocate for the maintenance of mathematics and computer science as vital requirements for career and college readiness.
- Be prepared to explain the differences between mathematics and computer science courses.
- Provide time for computer science and mathematics teachers to meet collaboratively and think critically about effectively connecting computer science and mathematics.
- Create systems which increase opportunities for students to study computer science without sacrificing mathematics content vital for career and college readiness.
- Advocate at the local, state, provincial, and national levels for effective, thoughtful, and systematic practices in the areas of course substitution for graduation, pathways through high school mathematics and computer science.

Researcher Actions

- Develop a research program that examines how computer science and its subdomains can have a positive influence on mathematics education, particularly in areas beyond the elementary grades.
- Research the effective infusion of coding environments into mathematics curricula.
- Develop and study curricula that use programming environments to help develop mathematical understanding.

50 NCSM CELEBRATING 50 YEARS of LEADERSHIP IN MATHEMATICS EDUCATION

References

Clements, D. H., & Battista, M. T. (1989). Learning of geometric concepts in a Logo environment. *Journal for Research in Mathematics Education, 20*(5), 450–467.

Clements, D. H., & Battista, M. T. (1990). The effects of Logo on children's conceptualizations of angle and polygons. *Journal for Research in Mathematics Education, 21*(5), 356–371.

Clements, D. H., & Battista, M. T. (1992). Geometry and spatial reasoning. In D. A. Grouws (Ed.) *Handbook of Research on Mathematics Teaching and Learning* (pp. 420–464). New York: Macmillan.

Clements, D. H., & Meredith, J. S. (1994). *Turtle math [computer program]*. Montreal, Quebec: Logo Computer Systems Inc. (LCSI).

College Board (2017). *AP computer science principles course and exam description, updated 2017*. New York: Author. Retrieved from: https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf

College Board (n.d.). AP Computer Science Principles: Course Overview. Retrieved from: https://apstudent.collegeboard.org/apcourse/ap-computer-science-principles

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, *42*(1), 38–43.

Harel, I., & Papert, S. (1990). Software design as a learning environment. *Interactive Learning Environments, 1*, 1–32.

LeadCS.org (2015a). Computer Science Terminology. CEMSE, Outlier Research & Evaluation, University of Chicago. Retrieved from http://leadcs.uchicago.edu/question/#communication_cs_terminology

LeadCS.org (2015b). Connecting and Integrating Computer Science With Other Disciplines. CEMSE, Outlier Research & Evaluation, University of Chicago. Retrieved from: http://leadcs.uchicago.edu/question/#courses_connecting_cs_with_other_disciplines

National Council of Teachers of Mathematics (2016). *Computer science and mathematics education: A position of the National Council of Teachers of Mathematics*. Reston, VA: Author. Retrieved from: http://www.nctm.org/uploadedFiles/Standards_and_Positions/Position_Statements/Computer%20 science%20and%20math%20ed%20022416.pdf

National Council of Teachers of Mathematics (2018). *Catalyzing change in high school mathematics: Initiating critical conversations*. Reston, VA: Author.

National Governors Association Center for Best Practices & Council of Chief State School Officers (2010). Common Core State Standards for Mathematics. Washington, D.C.: CCSSO.

National Research Council (2012). A framework for K–12 science education: Practices, crosscutting concepts, and core ideas. National Academies Press.

National Research Council (2013). Next generation science standards: For states, by states.

K–12 Computer Science Framework (2016). Retrieved from http://www.k12cs.org

LEADERSHIP IN MATHEMATICS EDUCATION