Computer Science and Mathematics Graduation Requirements

Overview for the NCTM Board

Developed by the NCTM Emerging Issues Committee[1]
March 2015

Background

In recent years, there has been much discussion about the need for more students, especially in high schools, to take computer science classes. Generally speaking, there are three common arguments for including computer science as part of the high school curriculum:

1. Increasing the focus on computer science at the K-12 level is critical because there are many computer science jobs open for those with the right skills, and the number of these jobs is going to grow.

2. Expanding K-12 computer science education could promote equity goals. Women and non-Asian minorities are underrepresented in computer science fields, making expansion of computer science education an equity issue. Moreover, to advance overall attainment across the nation will likely involve tapping into populations of students that have historically had less access to computer science education [CSE].

3. The skills of computing or computational thinking are now fundamental to all disciplines, and critical for success in today's world—though there is little evidence indicating that studying computer science in high school translates to improved performance in other high school disciplines.

Advocates for more computer science in the high school curriculum argue that efforts underway to expand the core requirements for high school graduation make it difficult to find room in the curriculum for students to enroll in elective courses such as computer science [ROE, p. 13]. For example, the emphasis on college and career readiness that is exemplified in the Common Core State Standards has resulted in increased course requirements in mathematics and other disciplines in some states.

Advocates for more computer science in the high school curriculum also argue that because computer science is not a core graduation requirement, it is relegated to a secondary status. Thus, there is increased advocacy both to require computer science courses and to allow computer science courses to count toward mathematics and science graduation requirements.

[1] The EIC thanks Sarah Wille and Michael Lach, of the University of Chicago Center for Elementary Mathematics and Science Education, for their assistance in identifying background information for this document.

In 2014, the Education Policy Committee of the Association of Computing Machinery (ACM) released a report titled *Rebooting the Pathway to Success: Preparing Students for Computing Workforce Needs in the United States* [RPS]. The report makes 10 basic recommendations as part of a national call to increase attention on computer science education so as to "increase the pipeline of qualified students obtaining computer science and other computing-related degrees, and to prepare them for the 21st century workforce." One of the *Pathways* recommendations is that "rigorous computer science courses should count as a core high school graduation requirement in computer science, mathematics, or science."

In 2013, the computer science advocacy group Code.org produced a two-page document called *Making Computer Science Fundamental to K-12 Education: Eight Policy Ideas* [MCSF] that provides some strong and thoughtful recommendations for school and system leaders to advance computer science. Recommendation 7 calls for schools to "Allow computer science to count for a core mathematics or science graduation requirement." Recommendation 8 calls for institutions of higher education to "Allow computer science to count as a mathematics or science admission requirement."

ACM and the Computer Science Teachers Association (CSTA) recommended in their 2010 report, *Running on Empty: The Failure to Teach K-12 Computer Science in the Digital Age*, that state and local governments should "Count computer science courses toward a student's core graduation requirements either as a computer science credit or as a mathematics or science credit" [ROE].

Code.org now reports that 25 states plus the District of Columbia allow computer science courses to satisfy either a mathematics or science course requirement, and that "thanks to the advocacy efforts by Code.org and sister organization Computing in the Core, the list of states that allow computer science to count towards graduation credit has increased monthly." This is evident from data in earlier reports. In 2013, CSTA indicated the number of states that allowed this was 14 (CSTA & ACM, 2013). In its 2010 *Running on Empty* report, CSTA and ACM reported that nine states allowed computer science credits to count as a mathematics or science graduation requirement.

Should Computer Science Count as Mathematics?

The fundamental question being raised by the Emerging Issues Committee (EIC) is whether computer science courses should satisfy a high school mathematics graduation requirement. The rapidly growing number of states that are allowing computer science courses to satisfy mathematics requirements suggests that NCTM should consider this issue soon.

To frame the issue for the Board, in this report the EIC addresses relevant issues regarding:

- College and career readiness and state graduation requirements;
- The mathematics in computer science courses; and
- (To a lesser degree) Teacher certification.

College and Career Readiness and State Mathematics Graduation Requirements

The fundamental question when considering whether computer science should count toward a mathematics graduation requirement is how such a policy would impact students' college and career readiness in relation to mathematics. We begin with a basic premise: *Supplanting a core mathematics requirement with another course that takes a student off the path of being (mathematically) college and career ready upon high school graduation is untenable*.

There is no federal definition of "college and career ready." However, many definitions exist. For example, there seems to be growing (but not complete) consensus in state documents that this means a student who has graduated high school and "is prepared to" enroll and successfully complete a collegiate or vocational program without the need to take remedial (non-credit bearing) courses [C].

In terms of the design and intention of the Common Core State Standards for Mathematics (CCSS-M), being college and career ready means that, minimally, upon high school graduation students need to have succeeded in meeting the elementary and middle school CCSS-M standards, and additionally to have met the goals set forth in each of the high school content domain standards (that are not marked by a plus [+]), together with increased capacity to employ each of the mathematical practices during their study in high school. If one looks at the non-compact model/example pathways in Appendix A of the CCSS-M document [A] meeting the college- and career-ready high school content standards intends that all students complete three high school mathematics courses, with content that leads through an updated Algebra II/Integrated Mathematics III course. In addition, students would continue to take mathematics courses throughout their high school years to "keep their mathematical understanding and skills fresh for use in training or course work after high school" [A, p. 3]. Advocacy groups Change the Equation and the National School Boards Association's Center for Public Education (CPE) also define college and career readiness in relation to the requirements of the CCSS-M [OS].

In determining whether to supplant a mathematics course requirement with a computer science course, the discussion above brings us to the consideration of the mathematics course requirements in different states. In 2013, Change the Equation and CPE examined state graduation requirements in relation to CCSS-M content requirements. [CE]. While noting that course requirements do not necessarily align with actual course enactments, they considered graduation requirements for three or four years of mathematics, including Algebra II or Integrated Mathematics III, to be fully or partially aligned with the college- and career-readiness expectations of the CCSS-M. Twenty-three states and the District of Columbia had requirements that were designated as fully or partially aligned with the CCSS-M expectations. States that require three years or less of mathematics without a requirement for "substantial content traditionally taught in Algebra I, Geometry, Algebra II, and statistics and probability" were deemed to be *not* aligned with the CCSS-M expectations. Notably, among states that have graduation requirements, five require only two years of mathematics.

It is reasonable to expect, therefore, that any recommendation about whether a computer science course can count toward a mathematics requirement may be different in states that require two, three, or four years of mathematics. The question of what mathematics would be supplanted by substituting a computer science course and how that substitution impacts college and career readiness is important to consider.

A 2013 study by the National Center for Education and the Economy, *What Does It Really Mean to be College and Work Ready? The Mathematics Required of First Year Community College Students*, [NCEE], calls into question a general requirement for a typical Algebra II course. NCEE examined the prerequisite knowledge needed in a wide range of introductory courses in degree programs at the community college level, (including Accounting, Automotive Technology, Biotech/Electrical Technology, Business, Computer Programming, Criminal Justice, Early Childhood Education, Information Technology, Nursing – and the General Track). It was found that "[t]he most advanced mathematics content used in the vast majority of the first-year college programs NCEE analyzed can reasonably be characterized as the mathematics associated with Algebra 1.25, that is some, but not all, of the topics usually associated with Algebra I, plus a few other topics, mostly related to geometry or statistics." [NCEE, p. 2]

While this seems to contradict the CCSS-M's purported foundation based upon college and career readiness, the suggestion is that the content in mathematics course sequences, such as those suggested in Appendix A of the CCSS-M, may not align with *actual* needs for career readiness. This might open the door for adjusting mathematics content in those or other courses, especially a third-year Algebra II course, so as to include other content options such as, "dimensions of mathematics not attended to by the CCSSM—the ability to interpret geometric visualizations, understand schematic diagrams and conduct complex applications of measurement…" [NCEE, p. 35] as well as sophisticated uses of topics first encountered in middle school like ratios and proportional relationships. This does not imply *less* mathematics is needed to be college and career ready.

The Mathematics in Computer Science Courses

In considering whether computer science courses should supplant a mathematics course requirement, the mathematics that appears in computer science courses needs to be considered.

While there are similarities, connections, and intersections, in using any modern definition of computer science, it is widely agreed (especially by computer scientists and mathematicians) that computer science is not a subfield of mathematics (or vice versa). Furthermore, there is not an

agreed upon definition of computer science[2] for computer science education. A perusal of Computer Science course codes used by some schools [JM] or course titles for CS listings [DS] highlights the wide range of course offerings listed by code or named in a computer science category. While these are only names, not curriculum descriptions, many of these courses probably contain very little mathematics. Examples include courses called Keyboarding, Desktop Publishing, and Office Applications.

In 2011, CSTA published its K-12 Computer Science Standards, which "present computer science at the secondary school level in a way that can fulfill a computer science, math, or science graduation credit," [CSTA]. As part of the standards document, CSTA developed some broadly stated K-12 standards and course frameworks that are often mentioned in referencing K-12 computer science education (and suggest a sequence of CS courses at the secondary level) However, there are few actual enactments of those standards in the form of curriculum materials. As such, the best we can do in terms of assessing the mathematical content of computer science courses is to look at those few available models and extrapolate from the CS Standards.

The College Board has developed *AP Computer Science Course A* [AP], which is offered in many schools and has a strong focus on computer programming. We use that course, as well as the *Computer Science Concepts and Practices* (CSCP) course concept described in the CS Standards—which both seem to be high-quality offerings—as a springboard to considering the mathematics content of CS courses. We will then briefly discuss other current CS curricula.

The mathematics prerequisites for *AP Computer Science Course A* are listed as basic algebra, and comfort with functions and the "concepts found in the uses of functional notation" (including the concept of function composition—a plus [+] standard in CCSS-M). The course description includes an additional sentence indicating that this course would build on a solid "foundation of mathematical reasoning." [AP, p. 7] The proposed CSCP course has similar prerequisites, although Algebra I is a "should" not a "must" prerequisite and the reasoning reference relative to this course refers to "computational thinking" in particular. CSCP is conceived as an elective course and is the *second* in a sequence of three possible courses at the high school level [CSTA, p. 20].

One consideration for the NCTM Board is the threshold of mathematics content that needs to be in a course in order to be considered as a mathematics core requirement. **We suggest that *a computer science course fulfilling a mathematics graduation requirement should involve***

---

[2] As an example germane to this discussion, consider the definition of CS offered in the ACM/CSTA publication *Running on Empty* [ROE]—the study of computers and algorithmic processes, including their principles, their hardware and software designs, their applications, and their impact on society. Or, the definition from Boston University's Dept. of CS: Computer Science—the systematic study of the feasibility, structure, expression, and mechanization of the methodical procedures (or algorithms) that underlie the acquisition, representation, processing, storage, communication of, and access to information, whether such information is encoded as bits in a computer memory or transcribed in genes and protein structures in a biological cell.

explicit mathematical learning goals. Generalizing from these two courses—an Advanced Placement course and a course in the middle of a proposed CS course sequence—can provide some insights about whether the mathematics included in computer science courses that might surface as contenders to supplant a math course.

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. Some of this content could be used in sophisticated ways, but these ways are unknown to the authors of this document.

In courses such as *AP Computer Science A* and CSCP mentioned above, this content is used when students develop, analyze or implement a computational algorithm, and many of these algorithms may arise when students encounter real-world problem-solving situations. As such, problem solving involving computational thinking can be related to, but not the same as, one or more of the broader Mathematical Practices stated in CCSS-M. Furthermore, the real-world problem solving may or may not be close to the process of mathematical modeling when it is implemented in problems—and even when it is, it may or may not be articulated as such. Finally, it is a concern as to whether the mathematical concepts that may underlie computer science ideas and constructs are explicitly made or that such ties are one of the goals of these courses.

Two other well-regarded computer science courses—*Exploring Computer Science* (ECS) and a new AP *CS Principles* course—are designed to broaden participation in the study of computer science. Two enactments of the new AP course, one developed at UC Berkeley and one being developed by Code.org, are in the early development and pilot stages. But like the other courses discussed, the mathematical learning in ECS and *CS Principles* is seen as a potential secondary benefit rather than as a primary goal. It appears that the mathematical content addressed in these courses is similar to that described earlier for *AP Computer Science A* and CSCP. As with the courses described above, therefore, these courses may provide opportunities to apply or fill gaps in students' mathematical knowledge but include little new mathematics or emphasis on deepening understanding of students' existing mathematical knowledge. Thus, even if the course is of the caliber of those described above, it appears that the mathematical content from a course that would be skipped by substituting a computer science course for a mathematics requirement most likely would *not* be learned by the student. Facility with Mathematical Practice-like thinking might occur if there is a focus on computational thinking.

Should a computer science course supplant a core math requirement at the high school level?

First, in this complicated landscape the answer is not a simple yes or no. We suggest that the answer is, "It depends." It depends on the nature (design and implementation) of the computer science course. It depends on the content of the course it is replacing. And it depends on the

number of mathematics courses that are required. Regardless of the answer to these questions, **our basic premise is unequivocal: the course should not replace any mathematics course that takes a student off the path of being (mathematically) college and career ready**

Questions to Consider

If the course is typical of the majority of secondary computer science courses that do not teach new mathematics content, we suggest the Board consider the following:

- What is the mathematics content of the computer science course and how is it enacted? The recommendation might differ when considering AP Computer Science A vs. a course on Web design.
- What is the requirement that is being supplanted? Is it a fourth-year course that goes beyond the three years of high school mathematics, such as those suggested in the CCSS-M Appendix A, or can the computer science course substitute for Algebra II, Geometry, or Integrated Mathematics II or III?
- What are the core graduation requirements in mathematics? It makes a difference if the computer science course is taken in a state that requires four years of mathematics (including Algebra II or Integrated Mathematics III) rather than in a state that requires only two years of mathematics or otherwise does not meet the requirements identified above as being aligned with the college- and career-readiness expectations of the CCSS-M. In the case of a four-year requirement, for example, in some cases it may be reasonable to suggest that the fourth year be either a mathematics course or a category of computer science courses. In a state with three years of mathematics required, the Board may recommend other stipulations. In all cases, our basic premise is that students should not be taken off the path of being (mathematically) college and career ready upon high school graduation.
- Should computer science courses be labeled as a mathematics course? We suggest that only courses that are designed explicitly to teach mathematics, with the mathematical goals guiding the course content, should be designated as mathematics courses. Otherwise the course should be labeled as a computer science course.

In considering a computer science course as a replacement for a math course, there may be benefit to creating a set of descriptors or design an evaluation tool for such a course. Some descriptors might be required, some might be listed in either/or categories. For example, one descriptor might be a strong focus on computational thinking. Another might be a focus on some problem solving in real-world settings. It might be advisable to require connections between some of the computer science principles studied and constructs within mathematics that are known to students or studied in the course and that these connections be made explicit. There may be other relevant descriptors for such a course.

In that light, **the NCTM Board may want to consider working with the computer science education community to define criteria for when a computer science course can count towards a mathematics graduation requirement.**

Another option for including computer science in high school could be a course (or courses) that strongly integrates computer science with mathematics topics rather than supplanting an existing math course. (This is similar in spirit to the approach taken by many of the NSF-funded projects in the 1990s that used various contexts to help teach the mathematics). In such cases, the computer science topics would be integrated into mathematics courses, with the mathematics in "first position." There are a number of projects producing some materials along these lines. Code.org, in conjunction with Bootstrap, is developing curriculum materials for middle school and into high school that teach algebraic and geometric concepts and align with CCSS-M [B]. Also, VCTAL is an NSF-funded project that is developing several instructional modules focused on real-world problem solving involving computational thinking that then may be incorporated into high school mathematics courses in grades 9-12 [V]. A third example might be a course (or courses) along the lines of the course relating to Algebra II+ in high school, described by Al Cuoco [BJCM]. These examples, however, address a different question—whether computer science can be used as a context for motivating and learning mathematics within the confines of a mathematics course rather than whether a computer science course should supplant a mathematics graduation requirement.

Issues related to teacher certification

It is commonly understood that if a teacher is to teach mathematics, she or he should know the mathematics being taught. CSTA makes the same argument about computer science. CSTA's 2013 report, *Bugs in the System: Computer Science Teacher Certification in the U.S.,* identifies a range of concerns regarding the certification of computer science teachers, most or all of which would likely be supported by NCTM. See [BITS] pages 9-14 for an introduction to the array of issues within the certification landscape. The EIC considered a related issue: If the computer science course replaces a mathematics graduation requirement, should it be taught by a mathematics-certified teacher?

We propose that the question should be less about the requirement being supplanted than by the content of the course. If the course is primarily a computer science course, with some connections to mathematical content, as described above, it should be considered a computer-science course and taught by a teacher who is certified to teach such a course. If, however, the course is considered a mathematics course with computer science content used in support of the mathematics learning, it should be taught by a mathematics-certified teacher.

References

[AP] *Computer Science: A Course Description*, College Board, 2014, http://apcentral.collegeboard.com/apc/public/courses/teachers_corner/4483.html

[ROE] *Running on Empty: The Failure to Teach K-12 Computer Science in the Digital Age*, The Association of Computing Machinery, The Computer Science Teachers Association, 2010, http://runningonempty.acm.org/fullreport2.pdf

[CSTA] *CSTA Computer Science Standards*, the Computer Science Teachers Association and the Association for Computing Machinery, (Revised) 2011, http://csta.acm.org/Curriculum/sub/K12Standards.html

[A] Common Core State Standards for Mathematics: Appendix A http://www.corestandards.org/assets/CCSSI_Mathematics_Appendix_A.pdf

[JM] Received from Jennifer Michalek (See below)

[DS] *School Leaders Mostly Mystified by Computer Science Education*, Dian Schaffhauser, THE Journal, February 2, 2015

[NCEE] *What Does It Really Mean to be College and Work Ready? The Mathematics Required of First Year Community College Students*, National Center for Education and the Economy, 2013, http://www.ncee.org/college-and-work-ready/

[C] Conley, D.T. (2012). A complete definition of college and career readiness. Retrieved from Educational Policy Improvement Center (EPIC) http://www.epiconline.org/publications/documents/College%20and%20Career%20Readiness%20Definition.pdf

[B] *CS in Math: Bootstrap*, code.org, https://code.org/curriculum/msm

[BJCM] *The Beauty and Joy of Computational Mathematics: The Case of Advanced High School Algebra*, Al Cuoco, (Attachment) EIC computer sci_context.1.docx

[MCSC] *Make Computer Science in K-12 Count*, Code.org, http://code.org/files/convince_your_school_or_state.pdf

[MCSF] *Making Computer Science Fundamental to K-12 Education: Eight Policy Ideas*, Code.org, http://code.org/files/Making_CS_Fundamental.pdf

[V] VCTAL, DIMACS, Rutgers University, http://dimacs.rutgers.edu/VCTAL/

[RPS] *Rebooting the Pathway to Success: Preparing Students for Computing Workforce Needs in the United States*. Kaczmarczyk, Lisa, & Dopplick, Renee. (2014). New York, NY: Association For Computing Machinery.

[CCSSA]Hart, Beth, Elaine Carman, Danielle Luisier, and Natasha Vasavada. (2011). Common Core State Standards Alignment: Advanced Placement. New York, NY: The College Board.

[CE] http://changetheequation.org/sites/default/files/MathGraduationRequirements_table.pdf

[SSE] *Stuck in the Shallow End: Education, Race, and Computing*. Cambridge, Margolis, Jane, Rachel Estrella, Joanna Goode, Jennifer Holme, and Kim Nao. (2010). MA: MIT Press.

[OS] *Out of Sync: Many Common Core States Have yet to Define a Common Core-worthy Diploma*, The Change Equation, 2013, http://changetheequation.org/out-sync

[BITS] *Bugs in the System: Computer Science Teacher Certification in the U.S*, CSTA, 2013,

http://csta.acm.org/ComputerScienceTeacherCertification/sub/CSTA_BugsInTheSystem.pdf
JS

[Insert horizontal Excel spreadsheet here]