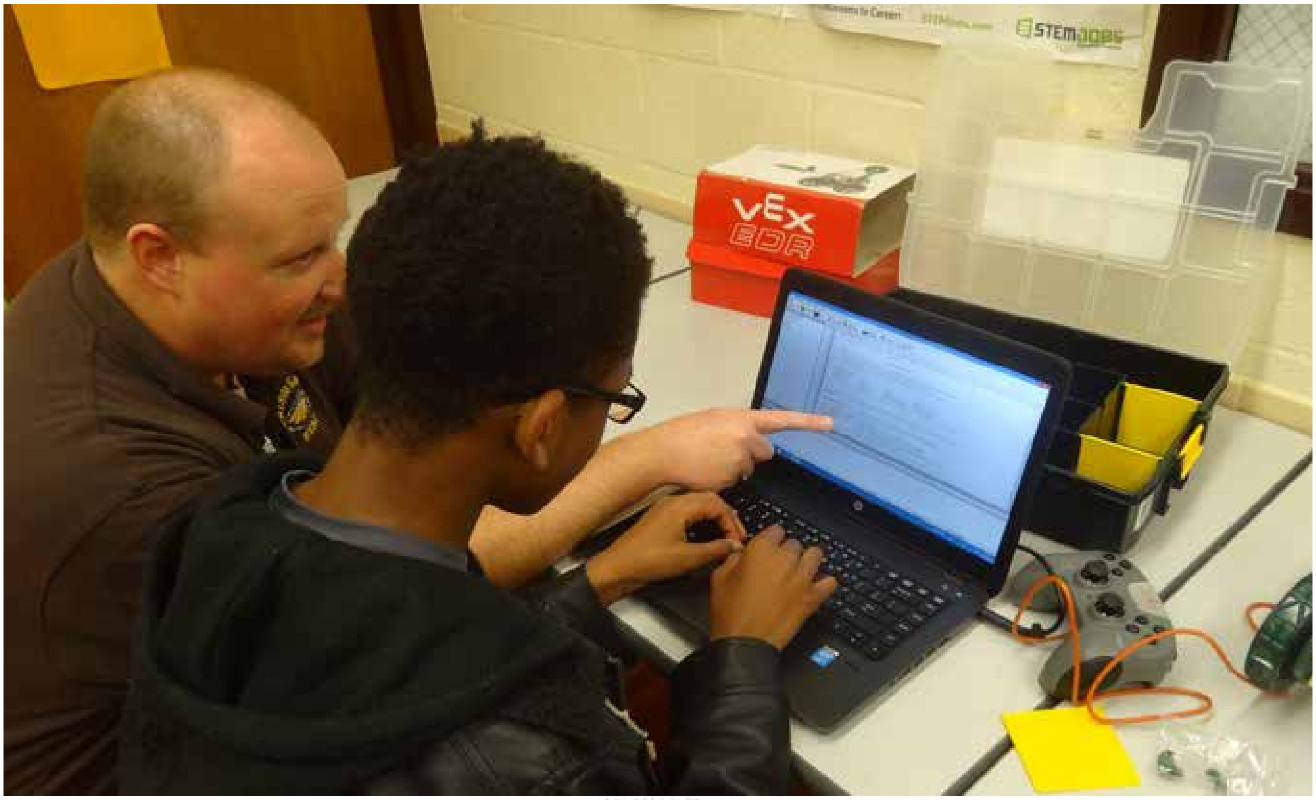
feature article



is computer science compatible with technological literacy?

There should be a place to include computational thinking knowledge and skills in technology and engineering education without wholesale substitution of our content.

by
Chris Buckler,
Kevin
Koperski, and
Thomas R.
Loveland, DTE

Computer Science and Computational Thinking

Although technology education evolved over time, and pressure increased to infuse more engineering principles and increase links to STEM (science technology, engineering, and mathematics) initiatives, there has never been an official alignment between technology and engineering education and computer science. There is movement at the federal level that is attempting to make the two content areas closer: the first was in 2015, when the U.S. Congress passed the STEM Education Act of 2015, which officially made computer science a part of STEM. The second was in January 2016, when President Obama announced his Computer Science for All initiative, setting a goal that every student who wants to learn computer science should be able to do so (Guzdial & Morrison, 2016). This initiative has set many states in motion to include computer science courses in their school systems as an elective.

Some states are in the process of identifying their own computer science standards because "having standards makes it easier to define classes, to create teacher certifications, and to grow teacher professional development programs" (Guzdial, 2016, p. 25). Along with trying to develop standards, there are many schools and districts using different versions of computer science, even though they may not be teaching the same content; titles of courses are labeled differently, and there is no consistency. This is the reason why The College Board worked with Code.org and Project Lead the Way and its AP Computer Science Principles courses to be able to offer consistent content, a framework, and benchmark goals for the students (Guzdial & Thompson, 2015).

According to The College Board (2016), through the computer principles courses, students will need to learn to be creative in their processing of computational artifacts and in using the computer software. There is no set coding program that the course requires; organizations and schools can select which options best suit their needs. In addition to developing code, The College Board (2016) states that students, "will also develop effective communication and collaboration skills, working individually and collaboratively to solve problems, and discussing and writing about the importance of these problems and the impacts to their community, society, and the world" (p. 4).

AP Computer Science Principles (The College Board, 2016) delineates seven big ideas related to computational thinking: Creativity, Abstraction, Data and Information, Algorithms, Programming, The Internet, and Global Impact. Creativity (Idea 1) promotes the notion that computing is a creative activity. Computers can be used as a tool to create artifacts (programs, audio, video, presentation, etc.) to help solve problems. Abstraction (Idea 2) is the concept of reducing information and detail to facilitate focus on relevant topics. Students will use abstraction techniques to reduce the complexity of a system, operation, or a task down to graphical, textual, and tabular formats through programming language. Data and Information (Idea 3) facilitate the creation of knowledge. Students will use computer science techniques to learn how data can be made into useful information. Algorithms (Idea 4) are used to develop and express solutions to problems. Algorithms are sequences of instructions for a wide array of processes that are accessed and controlled by computers. Programming (Idea 5) enables problem solving, human expression, and creation of knowledge. Programming language is used to develop software, video, audio, and other computational artifacts; it is the very essence of creation using computers. The internet (Idea 6) has become a baseline for modern computing communication. This section discusses how the internet functions, the characteristics of the system it runs on, and addressing issues such as cybersecurity. The last idea is Global Impact. Computing has played a major role in how society functions today; communication, creation, innovation, and discovery are being done through computers (The College Board, 2016).

Over half the country's schools allow a computer science course to qualify as some type of graduation credit (Guzdial, 2016). As part of this national initiative, the Maryland State Department of Education made a policy shift in 2015 to incorporate computer science standards into its technology education framework and allow computer science courses to count as technology education credit. Maryland is one of the few states that require a technology education credit for graduation from high school, and this policy is resulting in some districts shifting away from their technology and engineering education programs and replacing them with computer science programs.

Technology and Engineering Education in Maryland

Technology and engineering education has been an ever-changing entity in the world of education. High school technology education has been modified for decades to offer courses that use the engineering design process—to get students to think, reason, problem-solve, and create using problem-based learning. The push for computer science in technology and engineering education is becoming more prevalent as our reliance on technology increases even further. "Students need to know how to use technology, and they need engaged computer science learning opportunities to build creative thinking, logical reasoning and problem solving-skills that involve computing" (Page & Flapan, 2015, p.34).

Technology and engineering education has a long and illustrious history in Maryland. The most well-known educator in Maryland technology education was the late Dr. Donald Maley of the University of Maryland at College Park. His Maryland Plan for Industrial Arts found widespread influence not just in Maryland, but across the entire discipline. The Maryland Plan sought to transform the role of the industrial arts teacher from "a dispenser of facts" to "a facilitator—one who inspires, encourages, and evaluates" (Maley, 1969, pp. 5-6). Throughout the Maryland Plan, one will find "hands-on, minds-on" learning strategies that engage students in more meaningful ways than passive or rote learning. The Maryland Plan, among others, is credited by Wicklein (2006) as a foundational document for the modern version of technology education.

Perhaps the first codified implementation of modern-day technology education in Maryland occurred on August 2, 1993 with regulations mandating that all local school systems offer a technology education program in Grades 9-12. This mandate preceded the launch of ITEEA's Technology for All Americans Project, which led to the 2000 release of *Standards for Technological Literacy* (ITEA/ITEEA, 2000/2002/2007). The Maryland regulation was updated in 2005 to better align with *Standards for Technological Literacy*. On the same date, COMAR 13A.04.01.02 was adopted, requiring all school systems to certify that their technology education programs align to the content standards by September 1, 2007 and every five-year period thereafter. The Code of Maryland Regulations (COMAR), which is "the official compilation of all regulations issued by agencies of the State of Maryland" (University of Maryland, 2015), governs K-12 public education in the state. These regulations were later modified on January 14, 2010 to require all local school systems to submit technology education documents that align to the COMAR regulation and to the Maryland Voluntary State Curriculum for Technology Education (Division of State Documents, n.d.).

The groundwork for the policy shift toward inclusion of computer science curricula in the Maryland technology education graduation requirement began as early as 2010 within Montgomery County Public Schools (MCPS). A memo written by then-MCPS superintendent Joshua Starr cited concerns from students who argued that the 2005 revisions to COMAR were "too narrow," and that the new requirements "may have a negative effect of limiting students' pursuit of computer science study" (Starr, 2012). This view is corroborated by other Maryland computer science advocates such as Purtilo (2012), who called Maryland's 2005 technology education credit system the "ultimate policy deterrent to [the expansion of] computer science," and by other technology education supervisors in Maryland, who echoed Starr's desire for a broader array of course options, which include computer science (Gensemer, 2014). One concern of supervisors was the small pool of certified technology education teachers, making it difficult to fill open teaching positions.

The culmination of these concerns was a memorandum (2015) published by then-chief academic officer Jack Smith announcing the expansion of Maryland technology education requirements to include computer science, and the revision of the Maryland Technology Education Standards to include a standard on computational literacy. In addition to three technology and engineering education courses (ITEEA *Foundations of Technology*, PLTW *Introduction to Engineering Design*, PLTW *Principles of Engineering*), three new computer science courses, *Exploring Computer Science*, *Foundations of Computer Science*, and *Advanced Placement (AP) Computer Science Principles*, were allowed to count toward the Maryland technology education requirement. Any districts desiring to include additional courses beyond the six cited in the memorandum could complete a process sanctioned by Maryland State Department of Education that includes completing a curriculum alignment rubric, among other requirements.

COMAR and the Maryland Technology Education Standards

Educational programs at the K-12 level in the United States are often regulated by state-level regulations or laws. A section of the Maryland COMAR specifies the content that must be taught as part of a technology education program in Maryland high schools for Grades 9-12. To satisfy this regulation, high schools must offer a technology education program, and students must

Table 1. The Maryland Technology Education Standards – Standard 5. Essential Skills and Knowledge at the 9-12 Level

Objective #	Essential Skill and Knowledge
5.CTCSA.01	Decompose a complex problem or system into parts.
5.CTCSA.02	Use a programming language to develop solutions to problems and/or accomplish tasks.
5.CTCSA.03	Design, use, and evaluate computational abstractions that model the state and behavior of real-world problems and physical systems.
5.CTCSA.04	Automate solutions through algorithmic thinking.
5.CTCSA.05	Apply strategies for identifying and solving routine hardware and software problems.
5.CTCSA.06	Use a variety of productivity technology tools to collaborate with others, manage projects, collect and analyze data, share information, and/or publish findings.
5.CTCSA.07	Apply responsible legal and ethical behaviors in the use of technology systems and software.

Note: Coding of objectives done by UMES graduate students to help in processing the raw data.

Mr. Kevin Koperski (L) works with Malique Belgrave-Johnson (R) to program different movements using a Lynxmotion robot.

be offered instruction in the following areas as part of this program:

- The Nature of Technology
- Impacts of Technology
- Engineering Design and Development
- Core Technologies
- The Designed World

Central to the rationale behind the state regulations was the notion that Maryland high school graduates should be technologically literate. Technological literacy is defined in *Standards for Technological Literacy: Content for the Study of Technology* (ITEA/ITEEA, 2000/2002/2007) as "the ability to use, manage, assess, and understand technology" (p. 242). Garmire and Pearson (2006) expand on this notion of technological literacy, explaining that technologically literate people should have "a basic knowledge about technology," "[an ability to] employ an approach to solving problems that rely on aspects of a design process," and the ability "to think critically about technological issues and act accordingly" (p. 21).

To provide a basis for curricula that meet the requirements in COMAR, the Maryland State Department of Education (MSDE, 2005) released *Voluntary State Curriculum* (*VSC*) for technology education. *VSC* is a blueprint that contains standards, indicators, and objectives that should be included in courses students would take. *VSC* provides the foundation for several technology education courses, explaining specific concepts to be included in each course. The main course linked to the high school technology education program required by COMAR, as specified in *VSC*, is entitled *Foundations of Technology*, a course published by ITEEA's STEM Center for Teaching and Learning as part of its Engineering byDesign™ initiative. Links to ITEEA's *Standards for Technological Literacy* (*STL*) can be found throughout *VSC*. Thus, a strong case can be made that courses satisfying the COMAR requirement should directly address the *VSC* and *STL* (published by ITEEA).

Based on the memorandum (Smith, 2015), the Maryland Technology Education Standards (MSDE, 2016) were also revised to accommodate the policy changes to include computer science concepts. The original Standard 4, Core Technologies, and Standard 5, Designed World, were collapsed together, and a new Standard 5 was released titled Computational Thinking and CS Applications. Standard 5 of the Maryland Technology Education Standards—Essential Skills and Knowledge—at the 9-12 level is in Table 1.

Maryland Study on Computer Science

In response to the policy shift of 2015-2016, graduate students in the Career and Technology Education program at University of Maryland Eastern Shore initiated a research study to review the correlation of the new 2016 Maryland Technology Education Standards (Grades 9-12) to the *Standards for Technological Literacy* 9-12 Benchmarks (ITEA/ITEEA, 2000/2002/2007) and then compare the 2016 Maryland Technology Education Standards to course objectives in three computer literacy or coding-based courses (Buckler, 2017, Koperski, 2017). The three equivalent 11th grade courses are Computer Science Principles (Project Lead the Way), Computer Science Principles (CODE.org), and VEX Educational Robotics (EDR). The objectives used for comparison were the College Board's *AP Computer Science Principles*, Standards Learning Objectives used in Project Lead the Way, Unit Objectives used by Code.Org, and the Unit Learning Objectives in VEX EDR.

Briefly, the first results showed a strong correlation between ITEEA's *Standards for Technological Literacy* and the 2016 Maryland Technology Education Standards. When focused at the individual standard level, though, Standards (4) Core Technologies and Designed World and (5) Computational Thinking and CS Applications had lower mean averages and were not statistically significant. The second set of results showed that none of the three computer-based courses were significantly correlated to the Maryland Technology Education Standards. The highest mean match for all three courses was in Standard 5, Computational Thinking, and CS Applications. The results indicate that the computer science courses by themselves do not help students meet the Maryland Technology Education Standards, with the exception of the new Standard 5 (Buckler, 2017, Koperski, 2017).

The findings are an indicator that computer science courses are not in complete alignment with technology and engineering education, so action should begin at the state level to ensure that the courses are equivalent. At the present time, a student could take Code.org's Computer Science Principles in a Maryland high school and earn his or her required technology education credit without being taught any of the steps in the engineering design process or learning how to properly evaluate a product based on criteria and constraints. These are key elements that have made up technology and engineering education classes, and computer science just does not address them. Another finding was that poorly written course objectives and benchmarks make it difficult to know exactly what is being taught and how it matches to state objectives.

These results should be a conversation starter for education leaders in informing state, district, and local educators what computer science is, what elements make up computer science, and why it differs from technology and engineering education. There should be a way to reach an understanding between technology and engineering education and computer science leaders so that content areas are clearly defined and both programs are effectively distributed across the state.

Love and Strimel (2016) state that "there are successful curricular resources that have utilized CS as a tool to teach multiple components of the designed world portion of the *STL* and CS concepts" (p. 85). Cybersecurity, computer numerical control, and game art design are just a few of the promising courses with substantial computational thinking and doing. Computational thinking or literacy objectives could be incorporated into revised benchmarks of *STL* Standard 17, Information and Communication Technologies. There should be a place to include computational thinking knowledge and skills in technology and engineering education without wholesale substitution of our content. According to Dr. Phil Reed from Old Dominion University (2017), "I am one of those [who] believe our discipline is technological education (content) through design-based learning (pedagogy), period. We have a lot of interdisciplinary strengths/connections like all academic disciplines. We can and should be pre-engineering just as we can and should be pre-vocational. We also offer a contextual element to literacy and numeracy education not found in most other disciplines. However, our discipline at the very root is technological education."

Mr. Kevin Koperski (R) shows his students, Eugene Naguit (L) and Malique Belgrave-Johnson (M), in his Computer Integrated Manufacturing class how robot programming is used between inputs and outputs.

Conclusion

The December/January 2017 issue of *Technology and Engineering Teacher* presented a special themed issue, "Who Are We?" Inside the journal were three perspectives: one group advocated for a return to industrial arts, another advocated for continuing to focus on the concept of technological literacy (essentially the current course), and another group advocated for a complete restructuring of technology education as engineering education. Against this backdrop, it is unsurprising that in some states, the ambiguity of technology and engineering education—its ever-changing nature and its persistent struggle to gain acceptance—provides space for decision-makers to change the definition of technology and engineering education and replace T & E courses with computer science courses.

Technology and engineering education and computer science courses have been positioned in Maryland in such a way that they are considered equal in meeting the goal of teaching technological literacy, although the courses outline different objectives, knowledge, concepts, and tools. The misalignment of computer science courses to the standards developed by technology and engineering education leaders results in a misinterpretation of what each of the courses is about. While technology and engineering and computer science are both important content for students to understand in this day and age, a structure should be in place that allows the two courses to function independently of each other.

There can be a place in technology and engineering to include computational thinking. The starting point now might be to revisit *Standards for Technological Literacy* and include computational thinking benchmarks at all levels, particularly in *STL* 17, Information and Communication Technologies, and to ensure that all course and program objectives are clearly written and in alignment. There are technology courses like game art and computer numerical control currently being offered in states that incorporate computational thinking, and there are new courses in cybersecurity on the horizon. These courses are different than computer science courses. Treating technology and engineering education and computer science courses as the same content sends the wrong message to educators across the state and country.

References

Buckler, C. (2017). *An investigation into the relationship between the 2016 revisions to the Maryland technology education standards and the standards for technological literacy, Code.org computer science principles, Project Lead the Way computer science principles, and the VEX educational robotics curriculum.* Unpublished manuscript, Career and Technology Education Program, University of Maryland Eastern Shore, Baltimore, MD.

Division of State Documents (n.d.). Requirements for technology education instructional programs. Retrieved from www.dsd.state.md.us/comar/comarhtml/13a/13a.04.01.01.htm

Garmire, E. & Pearson, G. (Eds.). (2006). *Tech tally: Approaches to assessing technological literacy.* Washington, DC: The National Academies Press.

Gensemer, A. (personal communication, April 2, 2014).

Guzdial, M. (2016). Bringing computer science to U.S. schools, state by state. *Communications of the ACM, 59*(5), 24-25.

Guzdial, M. & Morrison, B. (2016). Growing computer science education into a STEM education discipline. *Communications of the ACM, 59*(11), 31-33.

Guzdial, M., & Thompson, A. (2015). Plain talk on computing education. *Communications of the ACM, 58*(8), 10-11.

International Technology Education Association (ITEA/ITEEA). (2000/2002/2007). *Standards for technological literacy: Content for the study of technology.* Reston, VA: Author.

Koperski, K. (2017). *An alignment comparison between the Maryland technology education standards and three computer science-based courses.* Unpublished manuscript, Career and Technology Education Program, University of Maryland Eastern Shore, Baltimore, MD

Love, T. & Strimel, G. (2016). Computer science and technology and engineering education: A content analysis of standards and curricular resources. *The Journal of Technology Studies, 42*(2), 76-86.

Maley, D. (1969). *The Maryland plan for industrial arts at the junior high school and the behavioral task analysis approach.* Retrieved from http://files.eric.ed.gov/fulltext/ED034852.pdf

Maryland State Department of Education. (2005). *Maryland technology education state curriculum.* Baltimore, MD: Author.

Maryland State Department of Education (2016). *Maryland technology education standards. Baltimore,* MD: Author.

Page, G. & Flapan, J. (2015). Why computer science matters. *Leadership, 45*(1). 34-37.

Purtilo, J. (2012). Maryland's tech education effort does not compute. Retrieved from www.baltimoresun.com/news/opinion/oped/bs-ed-computer-science-20120717-story.html

Reed, P. (2017, July 8). Re: *NSTA Engineering Edition Journal* [Electronic mailing list message]. Retrieved from CTETE@LIST.APPSTATE.EDU

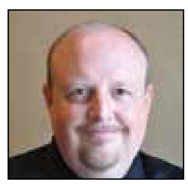
Smith, J. (2015). *Expanding technology education choices for students.* Retrieved from http://archives.marylandpublicschools.org/MSDE/divisions/dccr/docs/Computer-ScienceTechEdCredit.pdf

Starr, J. (2012). *Maryland state department of education technology credit.* Retrieved from www.montgomeryschoolsmd.org/boe/meetings/agenda/2011-12/2012-0521/5.0%20MSDE%20Technology%20Credit.pdf

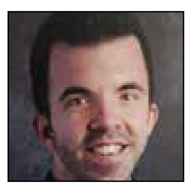
The College Board. (2016). *AP computer science principles: Course and exam descriptions.* Princeton, NJ: Author.

University of Maryland (2015). *Maryland law resources.* Retrieved from http://lib.guides.umd.edu/c.php?g=326466&p=2197150

Wicklein, R. (2006). Five good reasons for engineering as the focus for technology education. *The Technology Teacher, 65*(7), 25-29.



Chris Buckler *is the chair of the Technology Education Department at Owings Mills High School in Baltimore County. He is a recent graduate of the Career and Technology Education M.Ed. program at the University of Maryland Eastern Shore. He can be reached at sbuckler@bcps.org.*



Kevin Koperski *teaches Project Lead the Way at Meade High School in Anne Arundel County, Maryland. He is a recent graduate of the Career and Technology Education M.Ed. program at the University of Maryland Eastern Shore. He can be reached at kkoperski@aacps.org.*

Thomas R. Loveland, Ph.D., DTE *is a professor and director of the M.Ed. program in Career and Technology Education at the University of Maryland Eastern Shore in Baltimore. He can be reached at tloveland@umes.edu.*

This is a refereed article.