technology and engineering TEACHER
December/January 2018

SPECIAL INSERT: SAFETY DESIGN CHALLENGE

SPECIAL ISSUE: COMPUTATIONAL LITERACY

www.iteea.org

Join us in Atlanta in April of 2018, where there are SO many exciting activities planned, we created:

The ACTION ANNEX!

ITEEA
Technology and Engineering bring STEM to Life!
International Technology and Engineering Educators Association
iteea.org

Competitions!* *STEM Showcase!* *Ride-on car builds!* *Mini-Theatre Presentations!

It's all here and all free for ITEEA conference registrants!
Here's some of what you'll find on Thursday, April 12th and Friday, April 13th in ITEEA's Atlanta Action Annex!

THE ITEEA STEM SHOWCASE

Highlighting Best Practices Through Integrative STEM Education!

The ITEEA STEM Showcase is a unique and exciting event that features ideas, techniques, or best practices related to learning activities, marketing materials, career guidance, facility design, program design, assessment methods, equity, or classroom and laboratory management techniques. Showcasers illustrate a single element of technology or engineering teaching and learning that they feel they have exemplified. Attendees are invited to join ITEEA in the exhibit area for our Celebration Reception immediately following the STEM Showcase.

GO BABY GO RIDE-ON CAR BUILD

Go Baby Go! provides mobility (and play!) to very young children in the form of adapted toy ride-on cars. Over the course of the conference, faculty and students from Central Connecticut State University will be adapting several motorized ride-on cars near the Exhibit Floor. Detailed information about the Go Baby Go! program and ways to get involved will be available, and all conference attendees are encouraged to stop by for as little or as long as they'd like to see the program in action. On Friday, April 13th, the cars will be donated to children with special needs from the Atlanta area, who'll "test drive" their new rides before taking them home.
Learn more at www.tedmed.com/talks/show?id=292991.

TEECA STUDENT COMPETITIONS

The Technology and Engineering Education Collegiate Association is well represented at the ITEEA Annual Conference, with over 20 universities participating annually in a series of exciting competitions that require fast-paced ingenuity to solve problems and create solutions pertaining to robotics, manufacturing, and more. This year, teams competing in the Transportation Challenge will work to model, design, and fabricate a new frame for an existing quadcopter that will allow it to pick up packages of varying shapes and sizes and deliver them to predetermined locations. You won't want to miss the finals!

ITEEA MINI THEATRES

In Atlanta, ITEEA introduces Mini Theaters, which provide a forum for action-oriented presentations on topics such as STEM Wars, BattleBots, Student Competitions, and AMPing up Your STEM Instruction!

Make plans to spend time at the ACTION ANNEX in Atlanta!

Preregistration and Housing are Now Open!

For the latest conference information, go to
www.iteea.org/ITEEA_Conference_2018.aspx

ITEEA
ATLANTA
APRIL 12-14, 2018

ITEEA Children's Council's Innovative Grand Design Challenge!

The winning Challenge earns a one-year I-STEM Education Group Membership as well as one free hotel night in Atlanta, AND a spot in ITEEA's STEM Showcase!

ITEEA's Children's Council is sponsoring the Global Design Challenge for Elementary STEM to provide students with a chance to solve a real problem, and show the world that everyone can help find solutions to these global challenges.

Elementary STEM students will work in small design teams to solve the Challenge. The team with the most elegant solution to the GDC will be provided an opportunity to present at the STEM Showcase in Atlanta, GA, April 12-14, 2018 and one night's complimentary lodging. This winning solution will also be featured in the May 2018 issue of *Children's Technology and Engineering* and the team will earn an Elementary I-STEM Education Group Membership for their entire school!

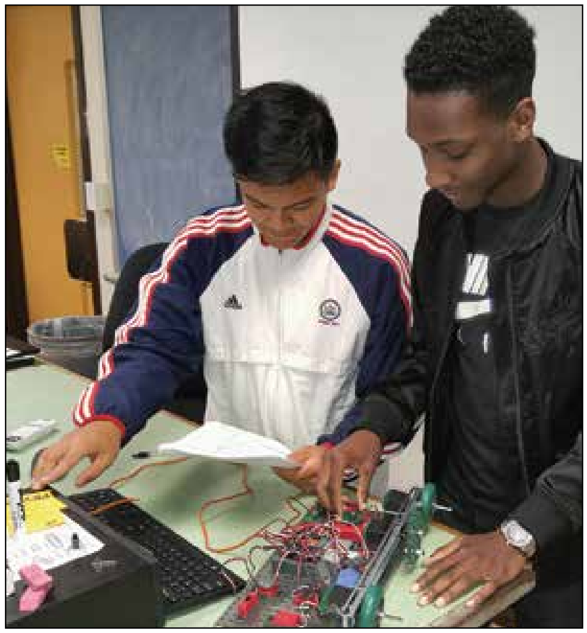
The Global Design Challenge: can you work as a member of a small design team to develop a better product or tool that can be used to give small children doses of liquid medicine? Learn full details about the Challenge at www.iteea.org/News/282/ESDesignChallenge.aspx.

Questions can be directed to Michael Daugherty, mkd03@uark.edu, Virginia Jones, vrjones@rappahannock.edu, or Thomas Roberts, otrober@bgsu.edu.

Submission Deadline: December 31, 2017
For the latest information, go to
www.iteea.org/News/282/ESDesignChallenge.aspx

contents

technology and engineering teacher VOLUME 77 ISSUE 4



features

departments

On the cover: *Photo courtesy of Kevin Koperski.*

ITEEA's 80th Annual Conference

Building Bridges Within the STEM Community and Beyond

Register today! Atlanta 2018

Join your professional colleagues April 12-14, 2018 at the 80th Annual ITEEA Conference in Atlanta, Georgia.

www.iteea.org/Registration_2018.aspx

ITEEA
ATLANTA
APRIL 12-14, 2018

Conference Preregistration Deadline is March 20, 2018!

https://www.iteea.org/ITEEA_Conference_2018.aspx

March 20, 2018 is the deadline to preregister for ITEEA's 80th Annual Conference in Atlanta, GA on April 12-14, 2018. After March 20, full conference rates will apply. Don't be late, or you'll miss the advantages.

- Save nearly 20% on your registration.
- Preregistered attendees will receive access to all conference programming through ITEEA's mobile app. Download the ITEEA mobile app to access events, speaker information, exhibitors, meeting rooms, maps, and much more!
- Have your packet ready for you when you arrive. No waiting!
- Be eligible to win a $100 Amazon gift card for preregistered attendees.
- Secure your housing to receive ITEEA's limited discounted room-block rate at the beautiful Westin Peachtree.

NEW! Computational Thinking: How it's Defined - How it's Practiced

https://www.iteea.org/ComputationalThinking.aspx

This new resource provides a launching point upon which teachers can build in their technology and engineering classrooms. The intent is to identify promising computational thinking practices as part of instructional practices and demonstrate to the entire STEM community the value of quality technology and engineering. Best practice examples of computational thinking embedded within iSTEM are being developed and will be posted as they become available. If you have a best practice to share for this resource, please send it to Tom Loveland at tloveland@umes.edu.

April 12-14, 2018 80TH ANNUAL ITEEA CONFERENCE
ATLANTA, GA
www.iteea.org/ITEEA_Conference_2018.aspx

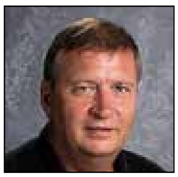
Preregistration Deadline: March 20, 2018
www.iteea.org/Registration_2018.aspx

Housing Deadline: March 14, 2018
www.starwoodmeeting.com/Book/ITEEA2018Conference

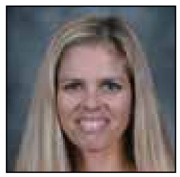
ITEEA ATLANTA APRIL 12-14, 2018

ITEEA Board of Directors Election Results

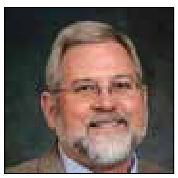
ITEEA's professional and life members have completed a balloting process to elect a new President-Elect and Directors for Regions II and IV. Joining the ITEEA Board of Directors at the ITEEA Atlanta conference in April 2018 are:



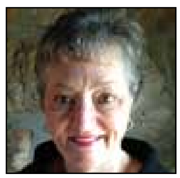
President-Elect
Michael A. Sandell, DTE
Michael is a Technology and Engineering Educator at Chisago Lakes High School in Lindstrom, MN.



Region II Director
Abbi Richcreek
Abbi is an Engineering/Technology Educator at Edgewood Middle School in Warsaw, IN.



Region IV Director
Gary Stewardson
Gary is an Associate Professor at Utah State University in Logan, UT.



ITEEA Children's Council
Also joining the ITEEA Board of Directors is **Charlotte Holter**. Charlotte is a Challenge Teacher (Gifted Education) at John Wayland Elementary and Linville-Edom Elementary in Bridgewater, VA.

Sincere thanks are extended to the new board members for taking on this leadership role and to the other candidates for bringing such a wealth of experience and talent to the balloting process. By being a part of the ballot, each of the candidates has demonstrated leadership in the field.

calendar

December 4-5, 2017
NICE K-12 Cybersecurity Education Conference
Omni Nashville Hotel
Nashville, TN
www.K12cybersecurityconference.org/

February 1, 2018
Entry deadline for the Engineering for Your Community Writing Contest by Engineer Girl
www.engineergirl.org/2018Contest.aspx

February 8-9, 2018
Virginia Children's Engineering Convention
The Hotel Roanoke
Roanoke, Virginia
www.cpe.vt.edu/vcec/

February 18-24, 2018
2018 Engineers Week
Engineers: Inspiring Wonder
www.discovere.org/our-programs/engineers-week

technology and engineering TEACHER

EDITORIAL POLICY
As the only national and international association dedicated solely to the development and improvement of technology and engineering education, ITEEA seeks to provide an open forum for the free exchange of relevant ideas relating to technology and engineering education.

Materials appearing in the journal, including advertising, are expressions of the authors and do not necessarily reflect the official policy or the opinion of the association, its officers, or the ITEEA Headquarters staff.

REFEREE POLICY
All professional articles in *Technology and Engineering Teacher* are refereed, with the exception of selected association activities and reports, and invited articles. Refereed articles are reviewed and approved by the Editorial Board before publication in *Technology and Engineering Teacher*. Articles with bylines will be identified as either refereed or invited unless written by ITEEA officers on association activities or policies.

TO SUBMIT ARTICLES
All articles should be sent directly to the Editor-in-Chief, International Technology and Engineering Educators Association.

Please submit articles and photographs via email to kdelapaz@iteea.org. Maximum length for manuscripts is eight pages. Manuscripts should be prepared following the style specified in the *Publications Manual of the American Psychological Association, Sixth Edition*.

Editorial guidelines and review policies are available at www.iteea.org/Publications/submissionguidelines.htm. Contents copyright © 2017 by the International Technology and Engineering Educators Association, Inc., 703-860-2100.
ITEEA is an affiliate of the American Association for the Advancement of Science.

TECHNOLOGY AND ENGINEERING TEACHER,
ISSN: 2158-0502, is published eight times a year (September through June, with combined December/January and May/June issues) by the International Technology and Engineering Educators Association, 1914 Association Drive, Suite 201, Reston, VA 20191. Subscriptions are included in member dues. U.S. Library and nonmember subscriptions are $90; $110 outside the U.S. Single copies are $10 for members; $11 for nonmembers, plus shipping and handling.

Technology and Engineering Teacher is listed in the Educational Index and the Current Index to Journal in Education. Volumes are available on Microfiche from University Microfilm, P.O. Box 1346, Ann Arbor, MI 48106.

ADVERTISING SALES
ITEEA Publications Department
703-860-2100
membership.sales@iteea.org

SUBSCRIPTION CLAIMS
All subscription claims must be made within 60 days of the first day of the month appearing on the cover of the journal. For combined issues, claims will be honored within 60 days from the first day of the last month on the cover.

Because of repeated delivery problems outside the continental United States, journals will be shipped only at the customer's risk. ITEEA will ship the subscription copy but assumes no responsibility thereafter.

CHANGE OF ADDRESS
Go to the ITEEA website – www.iteea.org. Log in and edit your profile. It's that simple.

POSTMASTER
Send address change to: Technology and Engineering Teacher, Address Change, ITEEA, 1914 Association Drive, Suite 201, Reston, VA 20191-1539. Periodicals postage paid at Herndon, VA and additional mailing offices.

Email: kdelapaz@iteea.org
Website: www.iteea.org

editorial

The "Who Are We?" special issue of *Technology and Engineering Teacher* (December/January 2017) generated a considerable discussion in listservs and forums as well as during a panel presentation at ITEEA's 2017 conference. These discussions and conversations are valuable to opening a healthy discourse within the field as different viewpoints and ideas are exchanged. Therefore, we set out to focus on additional topics that could lend themselves to more spirited discussion, resulting in this special issue of *TET*, which addresses calls for more computational thinking and coding skills in technology and engineering technology classrooms.

Why do we anticipate strong interest and opinions of the ITEEA members? Our field has long been confused with educational technologies. National surveys consistently show that Americans think first of computers when asked to define technology. The development of technological literacy standards demonstrated that computers are but one tool in the use of technologies, rather than the sole definition. Computer programming was once more closely associated with a career and technical education path than with the technological literacy of open-ended engineering design and problem solving. With the greater capabilities and infusion of computer thinking in today's work world, perhaps it is time for educational leaders to begin rethinking long-held ideas about computational thinking. In this special issue, we present three articles on computational thinking for the purpose of generating discussion, not to advocate for a specific agenda.

The first article discusses how computational thinking is a key 21st century skill that should be taught in technology and engineering classrooms. The focus would be on integrating conceptual understanding of computational thinking, rather than a skills-based focus on coding in order to expand the role that technology and engineering play in education. The second article reviews a decision made in Maryland to allow substitution of computer science courses for technology education required credit in high schools. While the article highlights the mismatch between the CS courses and the state definition of technological literacy, the authors nevertheless conclude that computational thinking can be adapted into technology and engineering education where it fits rather than as a wholesale substitution. The third article examines how computer programming can be infused in elementary mathematics classrooms through rich coding applications, effective training of elementary teachers, and using the applications to bridge learning across all content areas.

We hope you find this special issue informative and insightful. As is always the case, we welcome any constructive feedback, which can be emailed to kdelapaz@iteea.org.

computational thinking

by
Thomas Loveland, DTE
and Kathleen B. de la Paz

feature article

integrating computational thinking into technology and engineering education

There are several compelling reasons why CT should be integrated into T&E programs.

by Michael Hacker, DTE

Introduction

Computational Thinking (CT) is being promoted as "a fundamental skill used by everyone in the world by the middle of the 21st Century" (Wing, 2006). CT has been effectively integrated into history, ELA, mathematics, art, and science courses (Settle, et al., 2012). However, there has been no analogous effort to integrate CT into Technology and Engineering (T&E) education despite the vast opportunities it provides for engaging learners in CT practices in the context of authentic technological challenges.

Interest in computational thinking is not new. In the 1950s, it was referred to as "algorithmic thinking" (Denning, 2009). It can also be traced to Papert's interest in children working with computers to develop procedural thinking skills (Papert, 1980). A U.S. workforce well versed in CT was advocated by a presidential advisory committee over a decade ago (PITAC 2005).

Many definitions of CT have been proposed (NAS, 2010). The International Society for Technology in Education and the Computer Science Teachers Association (CSTA) have operationally defined computational thinking as:

> A problem-solving process that includes: formulating problems to enable us to use a computer to solve them; logically organizing and analyzing data; representing data through abstractions such as models and simulations; automating solutions through algorithmic thinking; identifying, analyzing, and implementing solutions to achieve the most efficient and effective combination of steps and resources; and generalizing this problem-solving process to a wide variety of problems (Barr, Harrison, & Conery, 2011, p. 21).

CSTA suggests that students should apply CT strategies and tools in virtual and real-world contexts to be better able to conceptualize, analyze, and solve complex problems (CSTA, 2011). Weintrop et al. (2016) remind us that:

> From a pedagogical perspective, providing meaningful contexts within which CT can be applied differs markedly from teaching CT as part of a stand-alone course in which the assignments students are given tend to be divorced from real-world problems. The sense of authenticity and real-world applicability is important to motivate diverse and meaningful participation in computational activities (p. 128).

Internationally, efforts to include CS in K-12 education are being made in Australia, China, Israel, Singapore, and South Korea (Wing, 2016). The UK Department for Education has provided statuory guidance for CS in the national curriculum. The purpose is to "implement high-quality computing education that equips pupils to use computational thinking and creativity to understand and change the world" (Gov.UK, 2013, p. 1). The curriculum focuses on helping 5- to 16-year-olds:

- Understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms, and data representation.
- Analyze problems in computational terms and have repeated practical experience writing computer programs in order to solve such problems.
- Evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems.
- Act as responsible, competent, confident, and creative users of information and communication technology.

In the U.S., teachers with varied backgrounds teach CS at the K-12 level, and many states do not require computer science certification (Teaching-Certification.com, 2011-2016). Guzdial (2012) has written that, "in most states, CS is classified in the business department, as a vocational education subject." Love and Strimel (2016) identified relationships between ITEEA's *Standards for Technological Literacy* (*STL*) and the K-12 CS Framework. Loveland (2012) discussed how teaching G&M code aligns with *STL* and the NCTM standards. *Next Generation Science Standards* lists "using mathematics and computational thinking" as one of eight "Science and Engineering Practices" (NGSS Lead States, 2013). However, no school discipline has yet to take on CT as a central focus. Since the school day is a zero-sum game, adding stand-alone new courses is a challenge. Integrating CS/CT into existing coursework might be considered more feasible. T&E can take the lead in addressing what is a growing need.

Rationale for Integrating CT into T&E Programs

There are compelling reasons why CT should be integrated into T&E programs. These relate to aligning T&E curriculum and instruction with societal and workforce needs; broadening participation in CT; the feasibility of implementation within T&E programs; and staunching the decline of T&E teachers. The need is real—T&E can help to fill that need, but transformational changes in professional mission, curriculum, and professional preparation are required.

Informed Design Pedagogy

The instructional model for integrating CT into T&E draws upon the informed design pedagogy that has been developed and validated through several large-scale NSF-funded projects (Hofstra, 2004, 2008). Using informed design (Burghardt & Hacker, 2004) students complete a series of just-in-time tasks called knowledge and skill builders (KSBs) that build their knowledge and skill base before they begin designing.

These pre-design KSBs help students gain the CS and CT competence needed to approach designing from a more informed perspective (rather than merely through trial-and-error). The subsequent design challenges call upon students to apply their new knowledge and skills to the modeling of prototypical solutions.

Aligning T&E Curriculum and Instruction with Societal and Workforce Needs

The public strongly supports the need for students to assimilate digital literacy. Silicon Valley executives are funding school-based CS programs (Singer, 2017), but since schools are not moving quickly enough into this space, coding bootcamps are proliferating to bridge the gap. T&E can play a role in helping students learn to become computational thinkers and thus become more highly regarded as part of the educational mainstream. This can be done without compromising the discipline's core mission of teaching students about the human-made world by integrating CS & CT principles, practices, and vocabulary with core T&E concepts—design, systems thinking, modeling, resources, and human values (Rossouw, Hacker, & de Vries, 2011).

Broadening Participation in CT

Integrating CT within project-based T&E contexts has the potential to significantly broaden participation for a large cohort of students (and their teachers) who might not be specifically interested in taking stand-alone CS courses but are interested in designing solutions to technological and engineering problems. Object-oriented programming environments like Scratch and Snap! have been used successfully to engage students (including those from underrepresented groups) in CS (Maloney, Peppler, Kafai, Resnick, & Rusk, 2008).

Figure 1. Supplies for use in physical-world contexts (e.g., robotics and computer control).

With the active advocacy of ITEEA, the 30,000 T&E teachers in the U.S. (Barbato, 2017) can serve as a significant new professional constituency that can support students to learn CS and CT skills.

Feasibility of Implementation

In a survey conducted by Hofstra University and ITEEA in December 2016, data was collected from a sample of T&E educators (n=202) about their interest in adding a CS/CT component to their programs. On a seven-point Likert scale, T&E teachers were highly supportive of adding a CS/CT component to their curriculum (median = 6.3); 76% of HS teachers responding would devote 18 or more weeks to such a program; and notably, 61% would attend a multi-week intensive summer PD program to learn how to teach CS/CT principles.

Design pedagogy is at the core of T&E instruction. Students frame the challenge, clarify criteria and constraints, engage in related research, iteratively generate ideas, make tradeoffs in choosing optimal solutions, develop and test prototypes, iteratively improve designs, and reflect upon and share their thinking with others.

Design thinking is also at the heart of CS and CT in that problems are framed; research is synthesized and informs the design process; problems are addressed through logical and systematic approaches; prototypes are tested for usability with target users; and designs are validated and iteratively improved through feedback.

Facilitating understanding of CS and CT requires conceptual understanding over and above coding skills. Preparing students to become computational thinkers requires a focus on the "big ideas" of computing: creativity, abstraction, programming, algorithms, data/information, the internet, and global impacts of computing (Snyder, Astrachan, Briggs, & Cuny, 2010).

T&E educators have a great deal of autonomy in making curricular choices, as they are normally not constrained by high-stakes testing. It is feasible to teach CT and computer science skills by incorporating real-world computing problems into T&E design challenges.

Stanching the Decline of T&E Teachers

The number of universities granting T&E undergraduate degrees in the U.S. has plummeted from 81 in 1988 (Moye, 2017) to 29 in 2016 (Rogers, 2017) (a 64% decrease); and the number of T&E BS/BA degrees awarded in the U.S. has fallen from 815 in 1995-96 to 206 in 2015-16 (Moye, 2017), a startling drop of 75%. T&E faces an existential challenge. Addressing CT not only will align curriculum and instruction with societal and workforce needs, but has the potential to expand the breadth of our teaching cohort, an issue critical to the survival of T&E education. Young people interested in CS, programming, and data science could serve as a new T&E teaching constituency. This new cohort could add immeasurably to the origination of design problems based on actionable insights from data and the subsequent data-driven analysis and optimization of solutions.

Curriculum and Professional Development

As with the introduction of any new educational program, exemplary curriculum must be provided (with guidance for students and teachers), and associated professional development (PD) should be offered.

Newly Developed or Adapted Curriculum

New curricula can be developed, but to do so requires funding and time for materials development, classroom testing, evaluation, and revision. Alternatively, existing exemplary curricula can be adapted for use in T&E programs.

An example. The NSF-funded *Beauty and Joy of Computing* (*BJC*), an introductory CS curriculum developed at UC Berkeley, is recognized for its appeal to a wide range of students. It uses an easy-to-learn, object-oriented language to teach key CS and CT principles (MSPnet, 2016). *BJC* has been extensively tested by students and teachers, including many in high-minority districts (Price, Albert, Catete, & Barnes, 2015). T&E curricular adaptations would apply *BJC* CS and CT concepts and skills to the solution of design problems in contexts that resonate well with the T&E community.

A sound pedagogical approach would guide students to revisit the same CS and CT concepts in both *physical-world* contexts (e.g., robotics and computer control) and *virtual-world* contexts (e.g., game design). Since most students are familiar with robotics through toys, movies, and industry-based robotic systems and have played electronic games, these contexts are particularly promising for connecting student experiences to computing, technology, and engineering. Curricular development and/or adaptation will most likely require collaboration between T&E and CS educators. Computational thinking can be taught using systems with which T&E teachers are comfortable and familiar, like robotics and game design. See Figures 1 and 2.

Figure 2. Gaming professional development workshop.

The curriculum model shown in Table 1 (page 12) is an illustrative example of how a one-semester course might be implemented to integrate CS and CT concepts and skills within T&E contexts. In this model *BJC* is used as an example of a curriculum to be adapted.

This approach is not intended to teach students to become programmers in languages like Python or JavaScript (this can come later); rather, it serves as an introduction to computer science principles where students will use a block-based, drag-and-drop programming language (Snap!, based on Scratch) to learn and apply key CS and CT ideas.

Professional Development

Inservice T&E teachers will need to learn how to integrate CT into their practice. Thus, development and conduct of PD programs to support implementation is essential. As noted earlier, when surveyed, T&E teachers expressed eagerness to attend intensive PD programs focused on CS and CT. Preservice teacher educators can advocate for programmatic reform, but that will require courage in confronting the realization that, in some cases, our own backgrounds may be insufficient to provide the instruction necessary. Engaging colleagues who have CS expertise could lead to mutually beneficial collaborations.

Research-Based Professional Development

In planning PD programs generally (and especially in areas of endeavor outside teachers' comfort zones), PD initiatives informed by research will have the highest likelihood of success. According to Darling-Hammond and McLaughlin (1995), teachers need a rationale for adopting new curricula. Traditional notions of inservice education need to be replaced by opportunities for "knowledge sharing." Teachers need to learn collaboratively, discuss what they know and want to learn, and engage in planning and evaluating (Darling-Hammond & McLaughlin, 1995). Loucks-Horsley and colleagues (2010) further suggest that programs be linked to school-wide efforts, that teachers help each other and choose their own goals and activities, that ongoing support be provided, and that the focus be on practices that result in improved student learning (Loucks-Horsley, Stiles, Mundry, Love, & Hewson, 2010). The PD plan shown in Table 2 (page 13) illustrates a sample agenda for the design of a two-week PD workshop based on the curriculum model shown in Table 1. The PD plan introduces teachers to the major CS and CT concepts and skills within T&E robotics and game design contexts.

Potential Research Opportunities

Integrating CS and CT into T&E programs offers a rich environment for scholarly research. Possible research questions might include:

RQ1. How should T&E courses be designed to help students learn core CS and CT ideas and capabilities?

RQ2. How can we help T&E educators become competent and comfortable with enacting CS and CT-related projects in their T&E courses and facilitating learning of CS and CT content and capabilities?

RQ3. In the context of T&E education that integrates CT, what does it take to get students to value CT knowledge and capabilities, have interest in continuing to engage in CT, and see themselves as people who engage in CS and CT?

A comparative case study approach might be used to answer these research questions. This methodology would compare and contrast data relative to teacher engagement and student learning in order to extract generalizable lessons learned. Data would help us understand how teachers gain confidence, competence,

Table 1. Robotics and Game Design Adaptations Using Existing *BJC* Curriculum as an Example: Sample KSBs and Design Challenges Within a One-Semester Technology and Engineering Course

This particular example of a T&E curricular adaptation uses informed design methodology (page 9) as the pedagogical backbone.

***BJC* CS and CT Concepts and Skills**	**Robotics/Computer Control**	**Game Design Context**
Building a simple app. Draw, move, and turn sprites.	***KSBs:*** Create a new program to turn LEDs on and off. ***Challenge:*** Design and program a traffic control system for an emergency medical services (EMS) station on a busy highway.	***KSBs:*** Controlling sprites; keyboard input, mouse input. ***Challenge:*** Program keyboard controls to move a small animal to safety.
Building Your Own Blocks (BYOB); using loops.	***KSBs:*** Create custom blocks that use a loop to gradually fade an LED and to change colors of a tri-color LED; move servos and motors; use loops to blink lights and move servos between two positions; use nested loops to create complex combinations of lights and motion. ***Challenge:*** Continue EMS station work.	***KSBs:*** Sprite cloning as object creation; create blocks to draw simple shapes; use loops to make complex patterns. ***Challenge:*** Control a white blood cell that eats replicating bacteria cells, thus protecting the human body from infectious bacteria.
Building grids for games; students use mathematics expressions to draw grids.	*Game design only*	***KSBs:*** Use list and matrix design to simulate probable spread of a forest fire. ***Challenge:*** Create a forest fire and hero to locate and put out the fire.
Conditional blocks; if-else and if statements and predicates (such as < or =) to control a program's behavior.	***KSBs:*** Write conditionals to control a robot using sensors; use sensor inputs to control the output of motors. ***Challenge:*** Design and program a vehicle to move a sample away from a dangerous biological environment.	***KSBs:*** Artificial Intelligence (AI) in games. Watson (IBM's supercomputer) plays Tic-Tac-Toe. ***Challenge:*** Explore programming routines for Watson so it never loses a Tic-Tac-Toe game to a human player.
Script variables; tools and techniques; Boolean operators (and/or/not). Using local variables to control systems.	***KSBs:*** Create a block to find and return the threshold for a sensor; use sensors in compound Boolean statements. ***Challenge:*** Design a physical whack-a-mole game that has a mole appearing randomly using servos and sensors.	***KSBs:*** Script variables, using programming tools. ***Challenge:*** Create an on-screen version of the whack-a-mole game that has a mole appearing randomly.
Using abstraction to write clear, debuggable programs.	***KSBs:*** Abstraction, reducing complexity, increasing efficiency. ***Challenge:*** Sensor-controlled design project (e.g., a vehicle that follows a white-line guide track to enter a radioactive environment).	***KSBs:*** Abstraction, reducing complexity, increasing efficiency. ***Challenge:*** Design a number guessing game that uses abstractions to write clear, debuggable, improvable programs.
Introduction to lists to store data; programs that access and manipulate list contents.	***KSBs:*** Use lists to store sensor data. ***Challenge:*** Design and model an environment to monitor and adjust temperature and light to protect museum artwork.	***KSBs:*** Using lists to store sequence data. ***Challenge:*** Design a Simon game where a list can store a sequence of lights that the user must repeat.
Nesting lists.	***KSBs:*** List matrices used to record ordered pairs to represent sensor measurements at various times. ***Challenge:*** Continue with above.	***KSBs:*** List matrices, placing lists within lists. ***Challenge:*** Continue Simon game development.
Combining list operations; higher-order list-processing functions.	***KSBs:*** Use the map function to scale data before graphing. Use combine to find the mean of data. ***Challenge:*** Continue with above.	***KSBs:*** Combining lists, linked to writing a script for the "add item" button. ***Challenge:*** Continue with above.
Algorithms and data; graphing; timers; reporters.	***KSBs:*** Acting on input data algorithmically. Controlling multiple outputs. ***Challenge:*** Continue with above.	***KSBs:*** Timers, reporters, modeling a graphing app. ***Challenge:*** Design a simulation relating population size to the rate of disease spread.

and understanding and what instructional practices might be implemented as teachers hone their CT skills.

Summary

Integrating CS principles and CT within T&E can expand the role the discipline plays in all students' fundamental education, can broaden participation in computing education, and can increase T&E's status within the educational system.

Presently, no discipline has taken upon itself the responsibility of being the primary instructional vehicle to teach CT in the nation's schools. T&E can take great advantage of this opportunity—without compromising the discipline's core mission of teaching students about the human-made world—by integrating CS & CT principles, practices, and vocabulary with core T&E concepts. It is feasible to teach CT and computer science skills by incorporating real-world computing problems into T&E design challenges.

Table 2. Sample Two-Week Workshop Schedule

Week 1	Monday	Tuesday	Wednesday	Thursday	Friday
AM	Introduction to integrating CS and CT within T&E. First look at Snap! Developing a phone app.	Build your own block! Blocks and scripts. Intro to control technology. Use blocks to control an LED.	Intro to lists and matrix design tools. Intro to abstraction. Complete white-blood-cell challenge.	More complex lists (e.g., nested) and programs. Use conditionals to control motor and servo outputs.	More complex algorithms. AI and its role in game design. Complete Tic-Tac-Toe game challenge.
PM	Experiment with basic commands. Write a program to move a sprite. On-screen drawing; make sprite follow the mouse.	Use scripts to create a light sequence. Complete the EMS station challenge. Discuss societal aspects, e.g., games & violence; discuss *Blown to Bits* book.	Game design using conditionals and script variables. Design a forest fire game: be a hero and put out a fire! Discuss privacy issues.	Design, make, and program a simple vehicle that reverses when it touches an object. Discuss copyrights and how they work in the computer world.	Use script variables. Linking game screen activity to HB inputs. Complete on-screen "Where in the World" challenge (to determine the best places to live in the future) using four digital inputs.
Week 2	**Monday**	**Tuesday**	**Wednesday**	**Thursday**	**Friday**
AM	Graphing in the real world: using analog inputs to display data on screen. Data tables and storing information.	Timers and timing. Design an "against the clock" number guessing game and improve it to store results of multiple tries.	Using the design journal as a scaffold, design, make, and program a four-mole Whack-a-Mole on-screen game.	Using the design journal, design, make, and program a vehicle that follows a white line guide track to enter a radioactive environment.	Managing the integration of CS and CT in the T&E classroom.
PM	Design and make a monitoring system for grow room temperature. Discuss encryption of data.	Linking digital and analog inputs to outputs. Simulate a one-axis prosthetic wrist. Discuss the workplace impact of robotics.	Extend the onscreen game to the real world using servos and switches. Present your work to group.	Improve the design to grasp the radioactive flask and remove it.	Present your work to group. Group discussion; implementation planning; evaluation, and feedback.

Doing so will require changes in curriculum and in the way teachers at preservice and inservice levels are prepared. This is well within the capability of those in our profession who are willing to be courageous enough to learn the necessary skills to lead what could be a transformative reform effort, well aligned with the transition from technology education to technology and engineering education.

The integration of CS and CT within T&E provides a rich area of inquiry for researchers to investigate how educators within and beyond T&E might optimize curriculum and pedagogy focused on broadening CS and CT participation.

References

Barbato, S. (2017, January 3). Personal email correspondence.

Barr, D., Harrision, J., & Conery, L. (2011). *Computational thinking: A digital age skill for everyone.* International Society for Technology in Education (ISTE), p. 21. Retrieved from www.iste.org/docs/learning-and-leading-docs/march-2011-computational-thinking-ll386.pdf

Burghardt, M. D. & Hacker, M. (2004). Informed design: A contemporary approach to design pedagogy as the core process in technology. *The Technology Teacher, 64*(1), 6-8.

Computer Science Teachers Association (CSTA) Standards Task Force. (2011). In *CSTA K-12 computer science standards*, p. 9. Retrieved from http://c.ymcdn.com/sites/www.csteachers.org/resource/resmgr/Docs/Standards/CSTA_K-12_CSS.pdf

Darling-Hammond, L. & McLaughlin, M. (1995). Policies that support professional development in an era of reform. *Phi Delta Kappan, 76*, 597-604.

Denning, P. J. (2009). The profession of IT: Beyond computational thinking. *Communications of the ACM, 52*(6), 28-30. doi: 10.1145/1516046.1516054.

Gov.UK. (2013, September 11). *National curriculum in England: computing programmes of study,* p. 1. London, England: Department for Education.

Guzdial, M. (2012, October 22). *Why isn't there more computer science in U.S. high schools?* Blog@CACM. Retrieved from https://cacm.acm.org/blogs/blog-cacm/156531-why-isnt-there-more-computer-science-in-u-s-high-schools/fulltext

Hofstra University. (2004). *The New York State curriculum for advanced technological education* (*NYSCATE*). Retrieved from www.hofstra.edu/academics/colleges/seas/ctl/nyscate/index.html

Hofstra University. (2008). The Mathematics, Science, and Technology Partnership Project (MSTP). Retrieved from www.hofstra.edu/academics/colleges/seas/ctl/mstp/index.html

Loveland, T. (2012). Understanding and writing G & M code for CNC machines. *Technology and Engineering Teacher, 71*(4): 24-29.

Loucks-Horsley, S., Stiles, K., Mundry, S., Love, N., & Hewson, P. (2010). *Designing professional development for teachers of science and mathematics* (3rd ed.). Thousand Oaks, CA: Corwin Publishing.

Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). Programming by choice: Urban youth learning programming with Scratch. *ACM SIGCSE Bulletin, 40*(1): 367-371.

MSPnet. (2016, January 27). *The beauty and joy of computing website*. Retrieved from http://hub.mspnet.org/index.cfm/29055

National Academy of Sciences. (2010). *Report of a workshop on the scope and nature of computational thinking.* Washington, DC: National Academies Press.

NGSS Lead States. (2013). *Next generation science standards: For states, by states.* Appendix F–Science and Engineering Practices in the *NGSS*. Washington, DC: The National Academies Press. Retrieved from www.nextgenscience.org/sites/default/files/Appendix%20F%20%20Science%20and%20Engineering%20Practices%20in%20the%20NGSS%20-%20FINAL%20060513.pdf

National Science Teachers Association (NSTA). (n.d.). *Science and engineering practices. Using mathematics and computational thinking.* Retrieved from http://ngss.nsta.org/Practices.aspx?id=5

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas.* New York: Basic Books.

President's Information Technology Advisory Committee (PITAC). (2005). *Report to the President. Computational science: Ensuring America's competitiveness.* Arlington, VA: National Coordination Office for Information Technology Research and Development.

Price, T., Albert, J., Catete, V., & Barnes, T. (2015). *BJC* in action: Comparison of student perceptions of a computer science principles course. *Research in Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*. Retrieved from http://ieeexplore.ieee.org/document/7296506/?reload=true

Rossouw, A., Hacker, M., & de Vries, M. J. (2011). Concepts and contexts in engineering and technology education: An international and interdisciplinary Delphi study. *International Journal of Technology and Design Education, 21*, 409. doi:10.1007/s10798-010-9129-1.

Settle, A., Franke, B., Hansen, R., Spaltro, F., Jurisson, C., Rennert-May, C., & Wildeman, B. (2012). Infusing computational thinking into the middle- and high-school curriculum. In: *Proceedings of the 17th ACM Conference on Innovation and Technology in Computer Science Education.* Association for Computing Machinery, New York, pp 22-27.

Singer, N. (2017, June 6). The Silicon Valley billionaires remaking America's schools. *The New York Times.* Retrieved from https://mobile.nytimes.com/2017/06/06/technology/tech-billionaires-education-zuckerberg-facebook-hastings.html

Snyder, L., Astrachan, O., Briggs, A., & Cuny, J. (2010). *An open letter to the computer science community: AP computer science principles.* Retrieved from https://csprinciples.cs.washington.edu/thecase.html

Teaching-Certification.com. (2011-2016). *Computer science teacher certification.* Retrieved from www.teaching-certification.com/computer-science-teacher-certification.html

Weintrop, D., Beheshti, El., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*, 125-147. DOI 1.1007/s10956-015-9581-5.

Wing, J. M. (2006, March). Viewpoint. Computational thinking. *Communications of the ACM, 49*(3), 33-35.

Wing, J. M. (2009). Computational thinking. J*ournal of Computing Sciences in Colleges 24*(6), 6-7.

Wing, J. M. (2016, March 24). Computational thinking, 10 years later. *Phys.org.* Retrieved from https://phys.org/news/2016-03-years.html

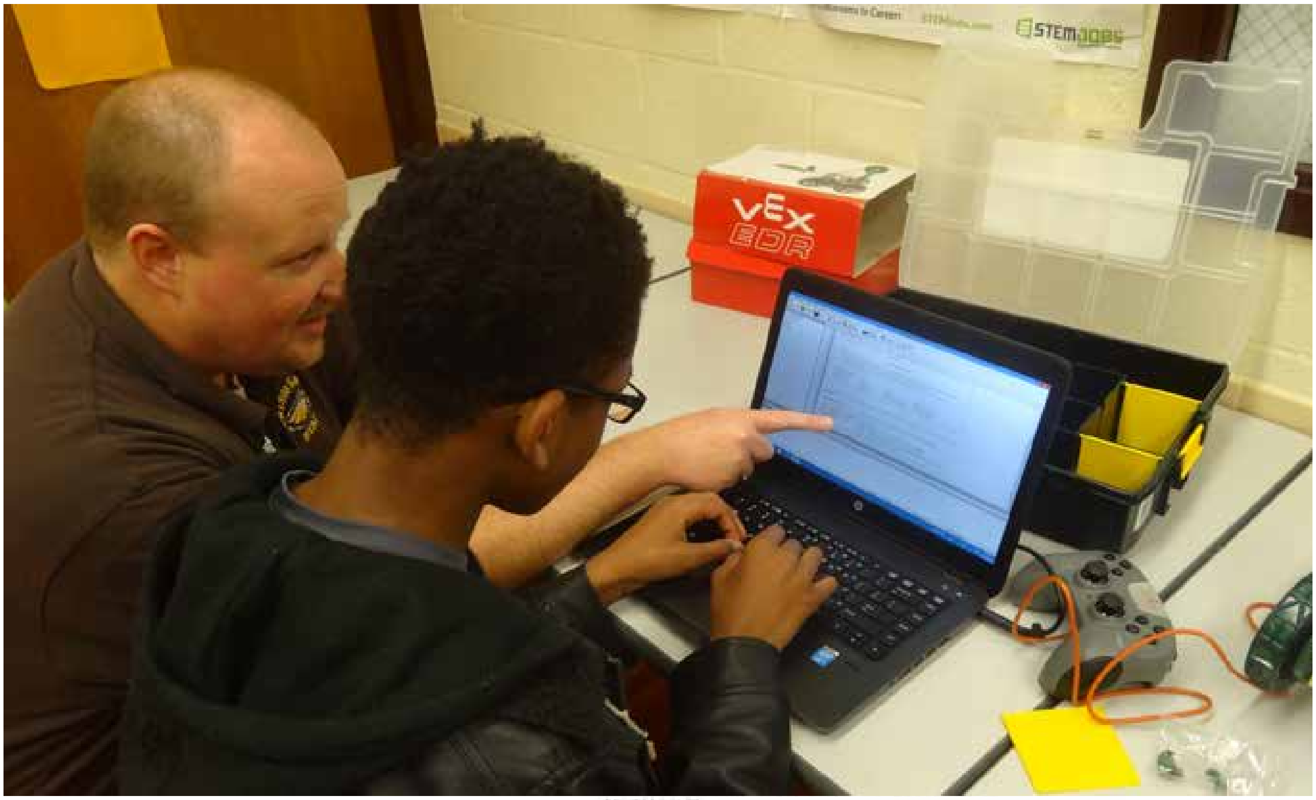
Acknowledgements

The author would like to acknowledge the contributions made by several colleagues to his understanding of the potential of computational thinking to support T&E instruction. Dr. Tiffany Barnes (North Carolina State University) and Tony Gordon (Hofstra University) helped the author to better understand how CT might be integrated within robotics and gaming contexts; and Dr. Janet Kolodner (The Concord Consortium) framed the research questions suggested herein as potential research opportunities.

Michael Hacker, Ph.D., DTE, *is Co-Director of the Hofstra University Center for STEM Research. He served as the EfA Project Principal Investigator. He can be reached at Michael.Hacker@hofstra.edu.*

This is a refereed article.

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/ComputerScienceTechEdCredit.pdf

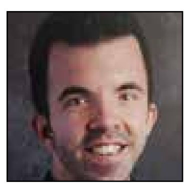
Starr, J. (2012). *Maryland state department of education technology credit.* Retrieved from www.montgomeryschoolsmd.org/boe/meetings/agenda/2011-12/2012-0521/5.0%20MSDE%20Technology%20Credit.pdf

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.

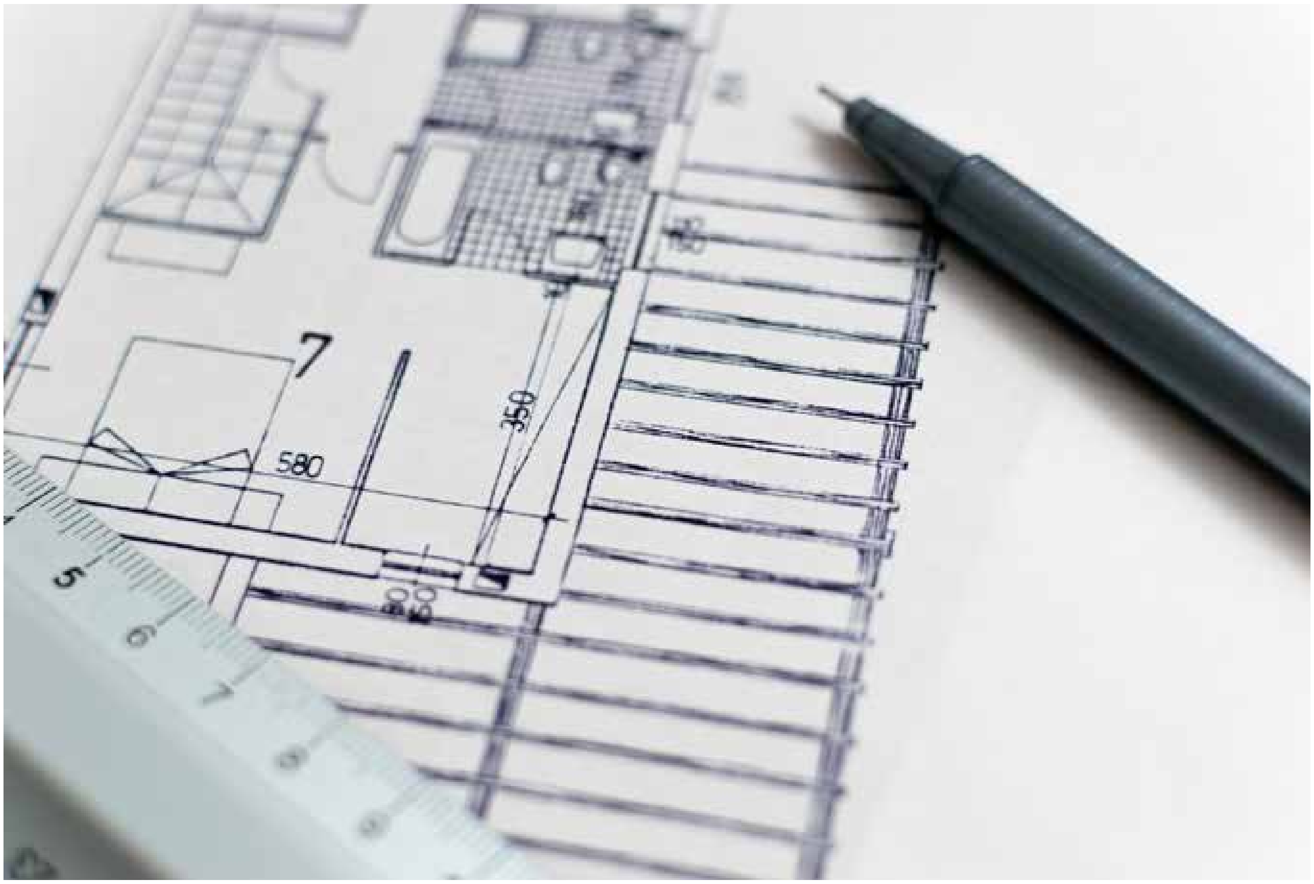
Chris Buckler *is the chair of the Technology Education Department at Owings Mills High School in Baltimore County. He is a recent graduate of the Career and Technology Education M.Ed. program at the University of Maryland Eastern Shore. He can be reached at sbuckler@bcps.org.*



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.



safety design challenge: exciting students using STEM concepts

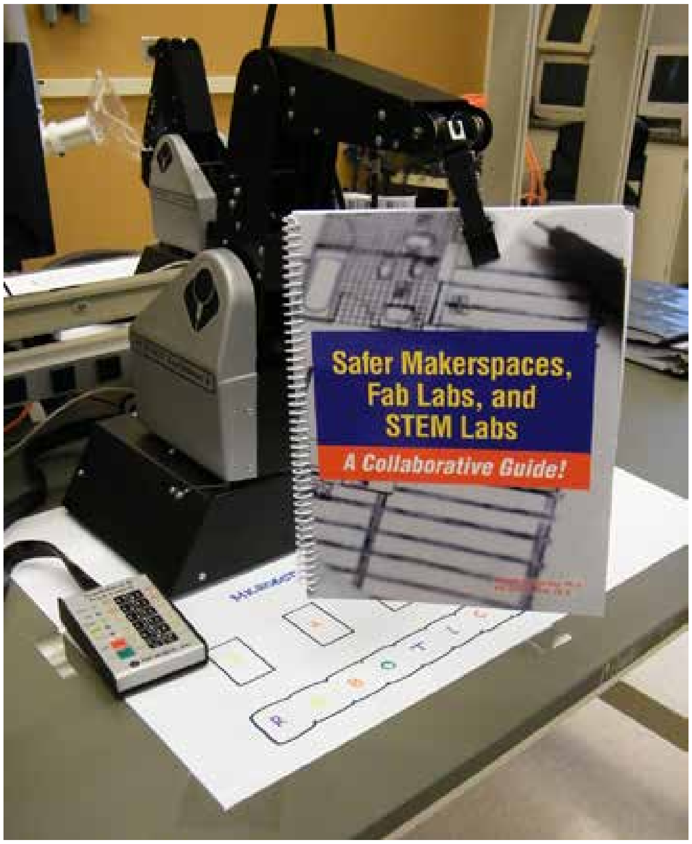
by Tyler S. Love and Ken R. Roy

Context

Safety is an integral part of the world in which we live. There are many safety features designed and built into our homes, schools, and transportation systems to name a few examples. A design challenge is an excellent way to help students identify the safety controls around them in a makerspace, Fab Lab, or STEM lab. Furthermore, it can be used to teach myriad standards-based concepts such as: federal and state building codes and standards (e.g., OSHA, NFPA, ICC), design and construction processes (*Standards for Technological Literacy* [*STL*]: 9, 20), budget management (*STL* 3), and communication technologies (e.g., AutoCAD, Revit) (*STL* 17). Resources such as *Safer Makerspaces, Fab Labs, and STEM Labs: A Collaborative Guide!* (Roy & Love, 2017) can help students gain a better understanding of safety regulations and controls that must be considered in the design of structures.

The Challenge

Students will be assigned a role (described in the extend phase) and work with the other stakeholders in their group to renovate or design a new makerspace, Fab Lab, or STEM lab for their school/community within the budget, size, and other constraints



specified by the instructor. Students must address all safety concerns in their design, and will be expected to communicate their final results through a design portfolio and presentation. The portfolio should document the design process and include floorplans created from software specified by the instructor.

Recommended Resources

Materials:

- Internet access
- Book by Roy & Love (2017) (see reference list, page 24)
- Article by West & Motz (2017) (see reference list, page 24)
- State safety guidelines for school facilities and/or STEM labs (check with your local and state supervisor. Some state STEM lab guidelines can be found at www.nsta.org/safety/)
- Computer software to draw floorplans (Google SketchUp, Microsoft Publisher, AutoCAD, Revit, or other software packages. An alternative to computer software is graph paper, an architect's scale, and colored pencils.)
- Modeling materials (poster board, etc.) to create a physical 3D model if desired.

Time:

- 12-14 class periods (based on 45-minute classes. May vary based on a number of factors.)

Implementing the Design Challenge

The following phases describe how instructors could use the 5E model to implement this design challenge.

Engage (1 class)

To challenge and excite students, there are a number of strategies that can be used. The instructor could divide students into groups and present each with an index card that describes different laboratory accident scenarios. Students could be asked to analyze what they would do when the accident occurs, which safety issues are involved, and which safety controls or design aspects of the building would help to limit the students' injuries. Another method is to show a video of a person who survived an accident as a result of safety features in a building. A guest speaker—such as your school system's facilities manager, safety officer, an architect, or fire marshal—could share his or her experiences with the safety requirements in structures. These examples help to generate dialog among the class so they see how safety regulations and controls protect them as occupants. Additionally, they assist students in thinking about safety features they may have overlooked.

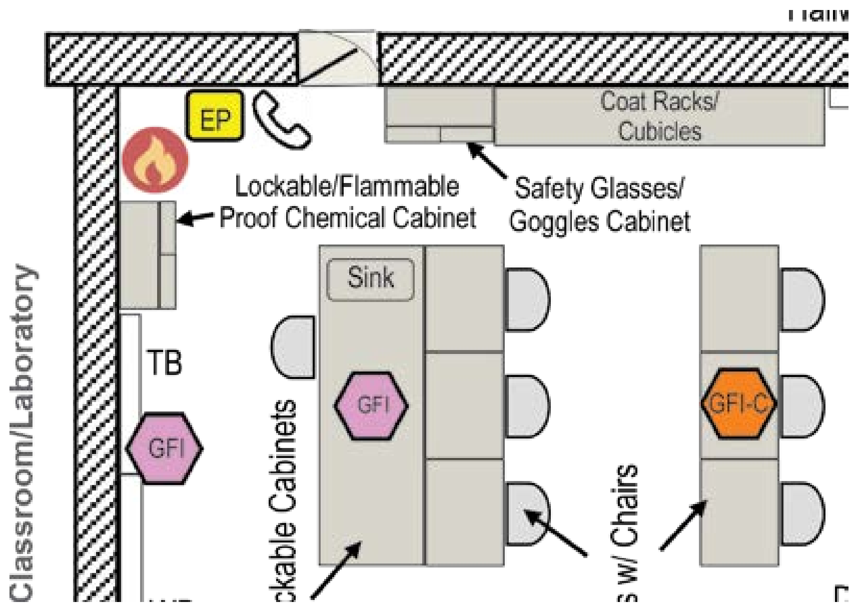
Explore (1 class)

Now that students have a greater appreciation for the safety controls in a building, the next step is to allow them to explore the design and placement of various safety items. During the Explore phase, students should be asked to conduct a laboratory inspection report individually or in pairs. An example of an inspection checklist can be found on pages 100-107 of the Pennsylvania Department of Education's Safety Guidelines document (www.teeap.org/Publications/safety/PDE_safety_guide.pdf). Students should take note of the design, placement, accessibility, quantity, color, symbol/signage, and other details associated with safety items such as engineering controls (ventilation, eye wash, machine guards, emergency power switch, etc.). Allowing students to visit a few types of labs or makerspaces is also a good experience for them to compare and contrast safety designs. This phase could conclude with a discussion about what the students saw and what they would change to make the area safer.

Explain (3 classes)

Building on the discussion of what students noticed during their inspection reports, the Explain phase allows the instructor to delve into more detail about designing safer structures. Were there critical safety controls that students had questions about or did not notice during their inspection? Students should be introduced to the building codes, federal standards, and state regulations related to makerspace and STEM lab safety in their state. Chapters 2 and 5 from *Safer Makerspaces, Fab Labs, and STEM Labs: A Collaboratie Guide!* (Roy & Love, 2017) provide clear explanations and examples of these items. Discussing the instructional needs or goals of the facility is also important when considering what safety features are required.

Students should be introduced to the phases of the design and building process. Which major stakeholders should be involved when designing a facility (in this case a makerspace or STEM lab)? What is the primary focus and role of each stakeholder? Lastly, students should have an understanding of how to read floorplans and the software (or architect's scale) they will use to create their designs. This could take more than the allotted class time pending students' prior experiences with the software (or scale), and how much detail the instructor wishes to provide. This could also lead to further discussion about the properties of various building materials.



Section of a lab design using computer software. From *Safer Makerspaces, Fab Labs, and STEM Labs: A Collaborative Guide!* (Roy & Love, 2017).

Extend (5-7 classes)

In the Extend phase students will be tasked with researching and designing their own makerspace or STEM lab. First, students should be split into groups and assigned roles as stakeholders that are integral to the planning and building process. Index cards explaining their stakeholder's key concerns and role can be handed to each student. For example, a student assigned the role of a school administrator may be concerned about the cost and the maximum number of students that can occupy the facility, while an architect may be worried about the aesthetics of the design and building codes. Different roles for this activity could include: engineers, architects, contractors, teachers, parents, administrators/supervisors, school facilities directors, school safety officer, local industry/community partners, nearby college faculty members, the fire marshal, local building official/code administrator, etc. Students should not share their assigned stakeholder's concerns with other students.

Next, students should work together in their groups to research various makerspace and lab designs, state safety regulations, and national safety standards. While groups design their makerspace or STEM lab, students should advocate for their stakeholder's concerns. The instructor can impose additional criteria such as a budget for their makerspace or lab, designing it to scale, and meeting certain green building characteristics. If the student in the contractor role wishes to make changes to the design, he or she should be required to get approval for the change order from the students serving as the architect and school representatives. During this design challenge, the instructor assumes the role of school board president and reserves the right to modify the budget or other criteria at any point. If the instructor wishes to extend this design challenge further, he/she can task students with constructing physical 3D models of their designs.

Evaluate (1-2 classes)

Students should be evaluated on their design portfolio and class presentation. Evaluation criteria could be based on the design process and how well students were able to articulate their ideas. They should provide a detailed explanation describing how their research findings influenced the safety features depicted in their design. The extent to which they addressed all safety regulations and standards should be a key criterion. Furthermore, the instructor could invite guests who represent the role of school board members. These guests could ask questions, complete evaluations, and select a final design that they would approve for building.

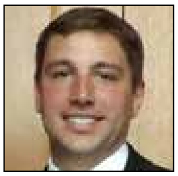
Conclusion

The design challenge presented in this article provides a very cost-effective way for students to learn about safety, the design-and-build process, green building concepts, budget management, working with others, and additional standards-based topics. Assigning students roles as various stakeholders in the design-and-build process helps them develop the skills needed to work with others who may have varying interests. This design challenge can be made as simple or complex as desired to integrate various STEM concepts.

References

Roy, K. R. & Love, T. S. (2017). *Safer makerspaces, fab labs, and STEM labs: A collaborative guide!* Vernon, CT: National Safety Consultants, LLC.

West, S. S. & Motz, L. L. (2017). Safer STEM and CTE classroom/laboratory facility design: General guidelines. *Technology and Engineering Teacher, 76*(5), 20-22.



Tyler S. Love, Ph.D., *is an Associate Professor and Coordinator of Technology and Engineering Education at the University of Maryland Eastern Shore. He currently serves as Editor of Safety Spotlight. Dr. Love can be reached at tslove@umes.edu.*

Ken R. Roy, Ph.D., *is the chief science safety compliance adviser for the National Science Teachers Association (NSTA), safety compliance officer for the National Science Education Leadership Association (NSELA), and a member of the International Council of Associations for Science Education Safety Committee. He also serves as Director of Environmental Health & Chemical Safety for the Glastonbury Public Schools (CT). Dr. Roy can be reached at safesci@sbcglobal.net. Follow Dr. Roy on Twitter @drroysafersci.*

Have questions or a safety issue that you would like to see addressed in a future Safety Spotlight article? Please send them to Dr. Tyler Love at tslove@umes.edu.

feature article

recommendations to support computational thinking in the elementary classroom

Computational thinking is an important and necessary way of thinking for computer programmers and other professionals in STEM.

by
Anne Estapa,
Amy Hutchison,
and
Larysa Nadolny

Mobile applications intended to provide exposure to the concepts of computer programming and coding, referred to as coding apps, are becoming increasingly recognized as useful tools for classroom instruction (Hutchison, Nadolny, & Estapa). For example, the ScratchJr app provides opportunities for users to create stories, games, and animations through visual coding and, as a result, experience what it is like to be a computer programmer. These programming apps can be used to expose students not only to computer programming or coding, but they also teach mathematics concepts and the broader skills associated with computational thinking by asking students to engage in tasks that require them to do things such as group variables, apply conditional logic, develop algorithmic functions, calculate angles within geometric shapes, and more. Computational thinking is described as a problem-solving process and can be defined as follows:

> Formulating problems in a way that enables us to use a computer and other tools to help solve them; logically organizing and analyzing data; representing data through abstractions such as models and simulations; automating solutions through algorithmic thinking (a series of ordered steps); identifying, analyzing, and implement-

Figure 1. Example of snap coding from ScratchJr.

ing possible solutions with the goal of achieving the most efficient and effective combinations of steps and resources; and generalizing and transferring this problem-solving process to a wide variety of problems (Society of Technology in Education [ISTE] and the Computer Science Teachers Association. [CSTA]) (Israel, Pearson, Tapia, Wherfel, & Reese, 2015, p. 263).

Computational thinking is an important and necessary way of thinking for computer programmers and other professionals in science, technology, engineering, and mathematics (STEM). Research on emerging practices around computational thinking that is developed through coding initiatives in schools reports that elementary children typically learn how to operate technologies rather than learn how to develop new technologies (Israel, et al., 2015). As a result, students in elementary schools experience only the receiving end of technology (Burke & Kafai, 2014). This lack of production potentially limits the effectiveness of technology integration since early experiences with computational thinking as a means of problem solving in abstract ways has the potential to improve attitudes, engage students, and enhance programing skills (Israel, et al., 2015). Thus, it is important to provide students with early exposure to computational thinking. Yet, with so many apps and so little guidance, it can be difficult to know how to integrate these apps into classroom instruction. Therefore, the purpose of this article is to provide recommendations for teachers, drawn from research, on how to select apps and begin practices that support computational thinking.

Recommendation #1: *Select Computationally Rich Coding Apps*

To ensure that an app is appropriate for all learners within the classroom, the idea of "low ceiling, high ceiling" should be a guiding principle. Grover and Pea (2013) explain that computational thinking tools (coding apps) for elementary students should be easy for beginners to start an activity and create programs or codes (low ceiling). However, the tool should also be powerful and extensive enough to satisfy the attention and learning of more experienced or advanced programmers (high ceiling). Apps with this principle in mind often follow a use-modify-create progression to allow a learner to experience each stage to support learning and engagement. Therefore, in reviewing apps for implementation within the classroom, the authors recommend that teachers select apps that allow students to increase their engagement and production with the app as their skill increases.

Grover and Pea (2013) highlight the following apps as examples that allow early experiences to focus on designing and creating: Scratch, Alice, Kodu, and Greenfoot. Many of the apps provided use a visual programming language, which allows programmers to snap visual programming codes together to control actors on the screen. This format supports computational thinking and provides students with the opportunity to create their own digital media products. Yet, it is simple enough that beginning users can be successful with the apps. The authors highlight this process in Figure 1, with an example from ScratchJr. In this example, the student selects a series of commands and places them in a logical sequence to make the animals move around the barn. Further, this app allows for the addition of a recorded speech response (represented by the microphone) that plays as the movement on the screen occurs. This example shows how simple it is to navigate a coding app such as ScratchJr and apply computational thinking skills (low ceiling). Yet, the app also provides opportunities for students to develop and apply more complex computational thinking by creating original characters, developing and connecting multiple scenes, changing colors and words, etc. (high ceiling).

Recommendation #2: *Become a Learner*

For teachers to effectively integrate coding apps into mathematics instruction it may be helpful to first engage with these apps as a learner. Some teachers may believe that coding is too difficult to learn or too far outside the realm of their expertise. However, coding apps and many coding initiatives are designed for beginners and require no previous coding experience. Many apps are designed with a game-like format or simple tutorials that teach the user what he or she needs to know to engage in the activities presented within the app. By engaging with coding apps as a learner, teachers can gain experience with the apps while also determining the specific concepts that can be taught through the app.

There are many popular apps and websites with which users can try to gain a better understanding of the function and purposes of coding apps. The authors recommend that teachers get started with Scratch or ScratchJr, depending on their own skill level and the grade level they teach. Scratch and ScratchJr (scratch.mit.edu) are both free and allow users to create animations, art, games, stories, or more. Scratch is targeted at ages 8-16 and allows users to program their own content, but also has an online community in which teachers can engage to get resources and ideas for integrating Scratch into their classrooms.

ScratchJr is targeted at younger students, ages 5-7, and is a great tool for those who are inexperienced with coding apps. Both apps teach computational thinking, as they require students to apply conditional logic and solve problems to get the outcome they want—creation of a game, image, animation, etc. Similarly, many coding apps require the application of mathematics skills such as group variables, applying conditional logic, developing algorithmic functions, and calculating angles within geometric shapes. Teachers can consider how these skills can be taught through the apps as they explore them for themselves.

Recommendation #3: *Use Apps for Active Learning*

Student content creation within a coding app can meet the needs of learners in several ways. The 2016 National Education Technology Plan (U.S. Department of Education, 2016) asks all educators to consider equity in the use of classroom technology, particularly considering differences in passive or active learning through technology. Differences in the way technology is used in the classroom for either more active creation with digital content and tools or more passive consumption of information from digital devices has been termed the "digital use divide" (U.S. Department of Education, 2016). By engaging all students in the active, creative use of coding apps, the teacher is helping to bridge the digital use divide. In addition to classroom activities, the authors recommend providing students, parents, and/or guardians with additional online resources to encourage engagement with groups underrepresented in the STEM fields. For example, Black Girls Code (www.blackgirlscode.com/) was created by Kimberly Bryant in partnership with major corporations in the fields of technology and finance. Students can attend workshops, join afterschool communities, and participate in hackathons across the nation. Girls who Code (https://girlswhocode.com/) hosts

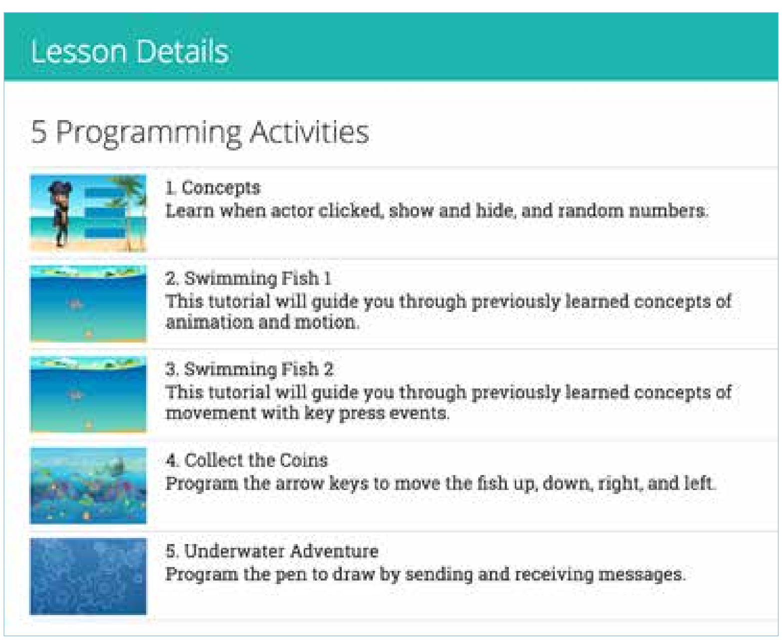


Figure 2. Sample lesson activities from Tynker.com.

Example of a coding app.

summer camps and afterschool clubs. If online resources do not meet the needs of students, teachers can consider starting a coding club using the free resources at Code Academy (www.codecademy.com/schools/curriculum/resources) or the ready-made lesson plans for afterschool clubs using the Tynker app (see Figure 2).

Recommendation #4: *Bridge Learning Across the Disciplines*

Recently, several researchers have illustrated how concepts of computational thinking can be aligned with other content areas to provide authentic learning experiences (e.g., Jona, et al., 2014; Sengupta, et al., 2013; Weintrop, et al., 2014). Advocates of computational thinking contend that computational thinking is at the core of all STEM disciplines (Henderson, Cortina, Hazzan, & Wing, 2007) and has the potential to bridge learning within and across discipline areas. Importantly, coding apps can be used to help students begin thinking like scientists, mathematicians, or engineers. For example, coding apps can be used to develop what Lucas and Hanson (2014) refer to as Engineering Habits of Mind (EHOM), which include: (1) systems thinking, (2) adapting, (3) problem-finding, (4) creative problem solving, (5) visualizing, and (6) improving. For instance, as part of a science lesson, teachers could ask students to create an animated demonstration of the life cycle of a butterfly using a coding app or explore the topic of adaptations (Figure 3). As part of that process, teachers could also teach and integrate engineering habits of mind such as creative problem-solving

Butterfly Coding Challenge

National Science Education Standards: K-4 The Characteristics of Organisms: Each plant or animal has different structures that serve different functions in growth, survival, and reproduction.

Now that you are familiar with how some butterflies use camouflage or a disguise to hide themselves from predators, it is time to help your own butterflies survive!

1. Choose two butterflies from a botanical garden website, such as http://rgbutterflyapp.com/ or www.missouribotanicalgarden.org/
2. Download images of your selected butterflies to your iPad.
3. Follow the same steps above to find and download a picture of a predator of butterflies.
4. Create a background in your coding app that will help hide those butterflies.
5. Using the sequence and looping tools in your coding app (control and motion in Scratch), move the butterflies and the predator, showing how a butterfly can survive by using its adaptations.

Extension

Turn your story into a survival game! Use controls and variables to allow the player to earn points when the predator touches the butterfly. For example, when the space bar is clicked, the wasp will move four steps in a random direction until it touches the butterfly.

Figure 3. Buterfly Coding Challenge

(EHOM 4) by having students generate coding and design solutions together and then by adapting (EHOM 2) their code and design to improve (EHOM 6) their demonstration.

Further, engaging with coding apps can also help students develop digital literacy skills and exposure to disciplinary vocabulary by introducing them to specialized language and opportunites to create and produce new information in digital contexts (Hutchison, Nadolny, & Estapa, 2016). Through the use of coding apps students can learn coding skills ranging from basic to complex, can learn how to devise and communicate effective messages through a combination of images, text, and color. Further, students will gain experience that will support their development towards proficiency with the International Society for Technology in Education's *ISTE Standards for Students* (2016), such as becoming computational thinkers and creative communicators.

To maximize learning when implementing coding apps into the classroom, teachers should begin by connecting the mathematical content learning within the app to one other discipline, building connections one content area at time. This will ensure that efforts are purposeful and that students will be shown the connection among the STEM disciplines. For example, when working on an app focused on computational thinking goals, through problem solving and representing data using graphs and/or tables (Mathematics) students could also engage in conversations around patterns in coding (Technology), create stories to provide context for what is happening on the screen (Literacy), or recreate a code using classroom materials to design and redesign paths given specific criteria (Engineering). In this way, the learning experience connects student understanding within and across STEM disciplines as recommended within *Next Generation Science Standards* (*NGSS*). In Table 1, the authors highlight how a computational thinking coding experience might align with *NGSS* (NGSS Lead States, 2013).

Through the integration process, the lesson or activity implemented supports student learning within and across STEM content areas.

Conclusion

The authors support claims that early access to and experiences with computational thinking will strengthen elementary students' computational thinking abilities while enhancing their understanding of mathematics and the connection of mathematics to other disciplines. In defining computational thinking as a way for

Table 1. *NGSS* K-2 Engineering Design Standards

Performance Expectation	Ask questions, make observations, and gather information about a situation people want to change to define a simple problem that can be solved through the development of a new or improved object or tool.
Science and Engineering Practices	Ask questions based on observations to find more information about the natural and/or designed world(s). (K-2-ETS1-1)
Disciplinary Core Idea	A situation that people want to change or create can be approached as a problem to be solved through engineering. (K-2-ETS1-1) Before beginning to design a solution, it is important to clearly understand the problem. (K-2-ETS1-1)

students to not only use computers to solve problems but also as a means to create and represent model solution strategies, student learning reaches beyond programming. As teachers explore options and purposefully integrate apps into their classroom following the recommendations in this article, students will be provided with the opportunities and tools they need to learn. The interest generated from such experiences has the potential to prime students for success within the classroom and in future computational-thinking-based opportunities.

References

Burke, Q. & Kafai, Y. B. (2014). Decade of game making for learning: From tools to communities. *Handbook of Digital Games,* pp. 689-709.

Code.org. (2014). *2014 annual report.* Retrieved from http://code.org/about/2014

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

Henderson, P. B., Cortina, T. J., & Wing, J. M. (2007, March). Computational thinking. In ACM *SIGCSE Bulletin, 39*(1), 195-196.

Hutchison, A., Nadolny, L., & Estapa, A. (2015). Using coding apps to support literacy instruction and develop coding literacy. *The Reading Teacher*, 1-11.

International Society for Technology in Education. (2016). *ISTE standards for students.* Retrieved from www.iste.org/standards/standards/for-students-2016

Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education, 82*, 263-279.

Jona, K., Wilensky, U., Trouille, L., Horn, M., Orton, K., Weintrop, D., et al. (2014). *Embedding computational thinking in science, technology, engineering, and math* (CT-STEM). Presented at the Future Directions in Computer Science Education Summit Meeting, Orlando, FL.

Lee, Y. J. (2011). Empowering teachers to create educational software: A constructivist approach utilizing Etoys, pair programming and cognitive apprenticeship. *Computers & Education, 56*(2), 527e538. Retrieved from http://dx.doi.org/10.1016/j.compedu.2010.09.018

Lucas, B. & Hanson, J. (2014). Thinking like an engineer: Using engineering habits of mind and signature pedagogies to redesign engineering education. *International Journal of Engineering Pedagogy, 6*(2), 4-13.

Lye, S. Y. & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12?. *Computers in Human Behavior, 41*, 51-61.

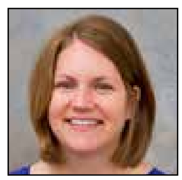
NGSS Lead States. (2013). *Next generation science standards: For states, by states.* Washington, DC: National Academies Press.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies, 18*, 351e380. Retrieved from http://dx.doi.org/10.1007/s10639-012-9240-

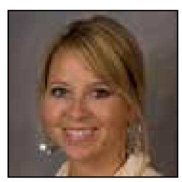
U.S. Department of Education, Office of Educational Technology. (2016). *Future ready learning: Reimagining the role of technology in education.* Washington, DC. Retrieved from http://tech.ed.gov/files/2015/12/NETP16.pdf

Weintrop, D., Beheshti, E., Horn, M. S., Orton, K., Jona, K., Trouille, L., et al. (2014). *Defining computational thinking for science, technology, engineering, and math.* Poster presented at the Annual Meeting of the American Educational Research Association (AERA 2014), Philadelphia, PA.

Wing, J. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-36.



Anne Estapa *is an assistant professor in the School of Education at Iowa State University. Her research focuses on the facilitation of teacher noticing and learning through teaching practices, technologies, and professional development opportunities. She can be reached at aestapa@iastate.edu.*



Amy Hutchison *is an associate professor at George Mason University. Through her scholarship she examines how digital technology can be used equitably to support diverse learners and ways of supporting the development of STEM literacy among underrepresented students.*

Larysa Nadolny *is an assistant professor in the School of Education at Iowa State University. Her research interests include examining factors in student motivation and achievement within digital environments.*

This is a refereed article.

Ad Index

premiere PD

designing standards-based STEM

When standards are appropriately taught, students will be better prepared for further education or to take on the challenges of the modern workplace.

by
Aaron Clark, DTE,
Bill DeLuca,
Jeremy Ernst,
Daniel Kelly, and
Jerry Ridgeway

With regard to STEM programs, there are seven national curriculum standards projects: *STL* (*Standards for Technological Literacy*); *NGSS* (*Next Generation Science Standards*); *NSES* (*National Science Education Standards*); *CCSS-M* (*Common Core State Standards - Mathematics*); and *PSSM* (*Principles and Standards for School Mathematics*). In addition to focusing on mathematics, *CCSS-ELA* also focuses on English Language Arts. All seven curriculum standards were developed by teachers, teacher educators, and business and industry stakeholders. The development of *STL* and *NGSS* also included engineers and scientists. Each of these curriculum standards shares a common purpose in attempting to accomplish the following:

- Improve the quality of teaching and programs in each of the respective STEM or language arts areas.
- Prepare all students to be fully participating 21st century citizens.
- Define literacy of each of the respective areas (what it means to be technologically literate, mathematically literate, and scientifically literate).
- Identify the most important concepts and skills within each of the respective areas.
- Identify a sequence of instruction.

- Approximate the appropriate age at which students should be introduced to subject-area concepts and skills as well as identify the depth and complexity of those concepts at various grade levels.
- Develop a common vision and "language" within the area of study.
- Develop common assessment tools for evaluating student achievement and program effectiveness.

The content that follows provides a closer look at the curriculum content and professional teaching standards.

Technology Education Standards: *Standards for Technological Literacy*

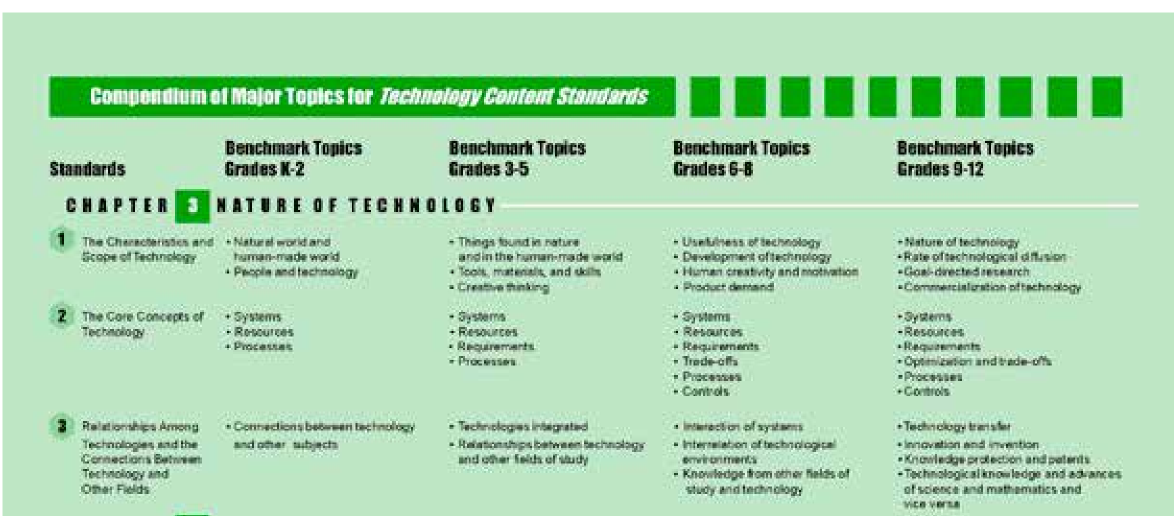
First published in 2000, by what is now the International Technology and Engineering Educators Association (ITEEA), *Standards for Technological Literacy: Content for the Study of Technology* (commonly called *STL*, available here: www.iteea.org/39197.aspx) became the curriculum standards for Technology Education. The work was funded by the National Science Foundation (NSF) and the National Aeronautics and Space Administration (NASA). These K-12 standards identify "what students should know and be able to do in order to achieve a high level of technological literacy" (*STL*, p. 12). There are 20 standards of two types. Some are primarily cognitive or conceptual, while others focus on process or performance. The 20 standards are clustered in five areas (see *STL*, p. 15): The Nature of Technology; Technology and Society; Design; Abilities for a Technological World; and the Designed World. In the Designed World, the traditional systems of communication, construction, and manufacturing, transportation, and power were often found in industrial arts programs. Two relatively recent additions to the traditional curriculum areas are agricultural and related biotechnologies and medical technologies. The standards are comprised of benchmarks that offer greater specificity as to what students should know and do. The benchmarks help the educator determine what concepts should be presented to the class, as well as determining the material's degree of complexity. ITEEA's STEM Center for Teaching and Learning™ has developed curricula that reflect the standards and mapped benchmarks to the courses it has developed (STEM Center Responsibility Matrix 071709.pdf). The *STL* standards were preceded and influenced by science standards.

Science Education Standards

Next Generation Science Standards (*NGSS*), released in 2013, addresses not only science standards but the role of engineering in society, as well as the ways in which science and engineering mirror each other. These standards, developed under the direction of the National Research Council (NRC), the National Science Teachers Association (NSTA), the American Association for the Advancement of Science (AAAS), and Achieve, can be found online at www.nextgenscience.org/get-to-know. *NGSS* outlines performance expectations that push student learning toward deeper understanding of core ideas, demonstration of scientific and engineering practices, and the integration of crosscutting concepts. These standards take a more comprehensive look at integrating science content with engineering practices, defining a new vision for science instruction that incorporates knowledge and skills needed for the 21st century, and elevating engineering design to the same status as scientific inquiry for teaching K-12 science.

Mathematics Education Standards

Led by the National Council of Teachers of Mathematics, *Principles and Standards for School Mathematics* (*PSSM*) was first copyrighted in 2000 and offered its fifth printing in 2008, www.nctm.org/.

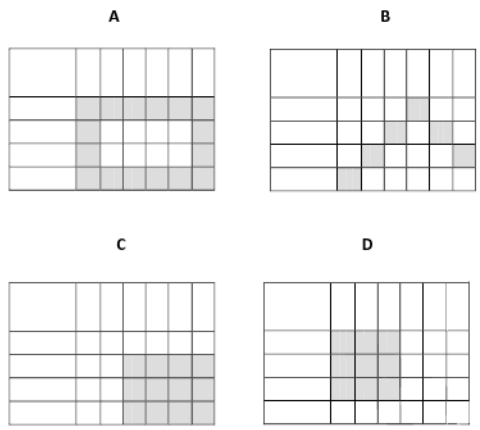


Compendium of Major Topics for *Technology Content Standards*

CHAPTER 3 NATURE OF TECHNOLOGY

Standards	Benchmark Topics Grades K-2	Benchmark Topics Grades 3-5	Benchmark Topics Grades 6-8	Benchmark Topics Grades 9-12
1 The Characteristics and Scope of Technology	• Natural world and human-made world • People and technology	• Things found in nature and in the human-made world • Tools, materials, and skills • Creative thinking	• Usefulness of technology • Development of technology • Human creativity and motivation • Product demand	• Nature of technology • Rate of technological diffusion • Goal-directed research • Commercialization of technology
2 The Core Concepts of Technology	• Systems • Resources • Processes	• Systems • Resources • Requirements • Processes	• Systems • Resources • Requirements • Trade-offs • Processes • Controls	• Systems • Resources • Requirements • Optimization and trade-offs • Processes • Controls
3 Relationships Among Technologies and the Connections Between Technology and Other Fields	• Connections between technology and other subjects	• Technologies integrated • Relationships between technology and other fields of study	• Interaction of systems • Interrelation of technological environments • Knowledge from other fields of study and technology	• Technology transfer • Innovation and invention • Knowledge protection and patents • Technological knowledge and advances of science and mathematics and vice versa

Summary of technological literacy content standards presented in Chapter 1 of ITEEA's *Standards for Technological Literacy: Content for the Study of Technology Executive Summary.*



Quiz Chart

Like the other curriculum standards projects, it identifies what it considers to be the most important concepts, when they should be introduced, and their level of complexity. There are ten standards: five content standards and five process standards. The content standards "explicitly describe the content that students should learn" as the following: number and operation, algebra, geometry, measurement, data analysis, and probability. Highlighting ways of acquiring and using content knowledge are the five process standards: problem solving, reasoning and proof, communications, connections, and representation. The math standards differ from the other content standards in that they provide the teacher with concrete examples of how to present and teach mathematical concepts and skills. Most, if not all, technology teachers appreciate the importance of mathematical concepts and skills for understanding and applying these skills in their areas. A deeper personal understanding of the standards outside of your core area improves your lessons, pedagogy, and activities, resulting in better student learning outcomes in all STEM areas.

Work began to develop *Common Core State Standards in Mathematics* (*CCSS-M*) in 2009, with the final *Common Core State Standards* (*CCSS*) released in June 2010. As noted above, *CCSS* focuses on mathematics and English language arts. (See www.corestandards.org/.) This effort was led by the Council of Chief State School Officers (CCSSO) and the National Governors Association Center for Best Practices (NGA Center). Of particular interest to STEM teachers are English language arts standards specifically written for teachers of science and technical areas.

State, Local, and National Technology Education Curriculum

It would be a serious omission to neglect mentioning state and local standards and curriculum that many teachers are expected to follow. Some states and local school systems have adopted curriculum content standards outright or modified them. Other states and school systems have developed their own standards. Regardless of where the curriculum content standards come from in the United States, the vast majority of local school systems develop their own curriculum. As schools create curricula in technology education, there is one well-known curricula of national importance: Engineering byDesign™. Approximately 20 states belong to ITEEA's Consortium of States, which develops the Engineering byDesign™ materials. This Consortium has published nine courses for Grades 7-12 that directly reflect *STL* (see www.iteea.org/EbD/ebd.htm). Teachers utilizing commercial or locally-mandated courses may not have as much freedom to change what happens in the classroom, while those using courses developed independently have the complete freedom to make changes.

Designing and Assessing Curriculum (Content, Activities, Assessments)

As discussed above, when teachers design or assess curriculum, consideration should be given to its content. Instructors attempt to answer the questions of what concepts, principles, and skills are most important, when these concepts should be introduced, and at what depth they should be taught. There are also universal principles and criteria that may help to design and assess instructional units, lessons, and assessments. They help the teacher determine what content is the least and most important, what to avoid, and what to stress. These principles will also help in determining content sequencing and depth of coverage, and the type of activities in which the students should be engaged. A powerful tool for assessing one's lessons, instructional units, assessment instruments, and courses is the Revised Bloom's Taxonomy (RBT) Chart (www.celt.iastate.edu/teaching/effective-teaching-practices/revised-blooms-taxonomy). To develop a deep understanding of the RBT requires time and commitment. Nevertheless, the RBT Chart can be used without a great deal of training to get a quick and rough sense of the types of information and activities students are engaged in day by day. A Knowledge Dimension column is on the left (factual, conceptual knowledge, etc.), and a process dimension row on top (remember, understand, apply, etc.).

In the Procedures in the Classroom section of this learning object teachers are asked to "map" the activities in an instructional unit (a unit covering several days to a couple of weeks) to see where the activities are located. If most activities in the chosen unit fit in the A1 cell (remember factual knowledge) and few other cells, there may be a serious problem. Generally, instructional activities should demonstrate a wide variety of knowledge and cognitive processes. It is important to point out that an introductory course usually spends more time in the cells found in the upper left-hand quadrant of the chart, largely because the introduc-

Revised Bloom's Taxonomy Worksheet Example

Knowledge Dimension	1. Remember	2. Understand	3. Apply	4. Analyze	5. Evaluate	6. Create
a. Factual Knowledge						
b. Conceptual Knowledge						
c. Procedural Knowledge						
d. Metacognitive Knowledge						

Instructors can create a working spreadsheet based on the example above.

tory phase of most new learning requires the acquisition of new terms, concepts, and skills (procedural understanding). Naturally, activities in an advanced course should reflect higher-order thinking and creating than are found in an introductory course. Nevertheless, it is important to remember that all courses should have wide coverage of the knowledge and cognitive processes reflected by the RBT Chart. In other words, even in introductory courses, students should be engaged in higher-order thinking and creating, and in upper level courses students should still be learning new concepts, principles, and skills. One can also use the RBT Chart for critiquing assessments such as quizzes and tests. For example, if while using the chart to evaluate a unit assessment it's determined that most questions fall in the A1 cell, it may mean that the unit dealt primarily with lower-order understanding.

The Columns of the Cognitive Process Include:

1. Remembering
2. Understanding
3. Applying
4. Analyzing
5. Evaluating
6. Creating

The Rows of the Cognitive Process also Include:

A. Factual Knowledge
B. Conceptual Knowledge
C. Procedural Knowledge
D. Metacognitive Knowledge

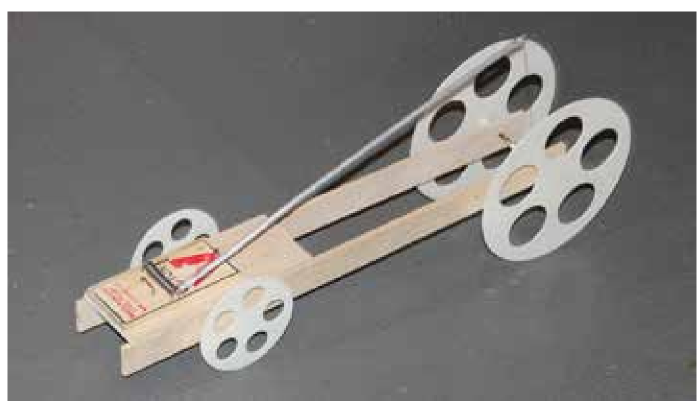
Curriculum Integration and Alignment

One of the most powerful arguments for curriculum integration and alignment is pragmatic in nature—it follows from necessity. In all STEM areas instructors must use concepts, skills, and principles of language, mathematics, science, technology, and art to teach STEM content. As an example, the development and use of models is an identified core practice within science, technology, and mathematics: both virtual and physical models. Making models or prototypes is central to the *STL* standards. Developing and using models is one of the eight Science and Engineering Practices (SEP) identified within *NGSS*. One of the eight *CCSS-M* practices focuses on modeling with mathematics. STEM teachers and students must use models to understand, analyze, evaluate, and create the things we do. Four principle types of models are used: conceptual, graphical, mathematical, and working.

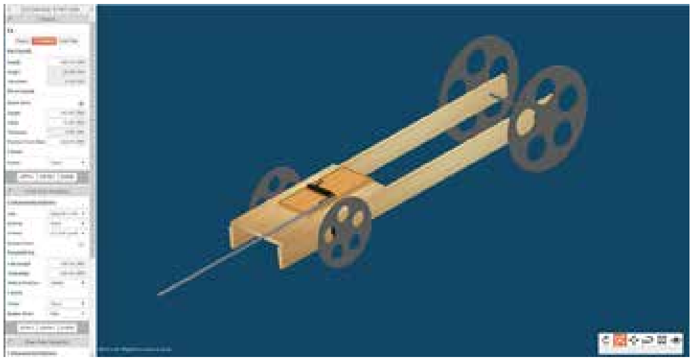
Conceptual models (written and spoken words organized into thoughts, such as poetic imagery) are utilized when talking about ideas and concepts, defining problems, writing technical papers, and working together in teams. Many of these conceptual ideas come from the sciences and the arts. In order to understand simple electricity and design a circuit, it is important to have at least a simple conceptual framework of atomic theory.

Graphical models are used to develop technical drawings, apply dimensions to a concept, and develop charts and graphs. This requires the application of art, geometry, and mathematics. We use **mathematical models** when we measure things, do geometrical constructions, solve for unknowns ($I = E/R$), or analyze data. Clearly, this type of planning and teaching integrates science, mathematics and technology concepts and skills. We are also using models to depict ideas, test, analyze, and solve problems. Building a **working model** involves the use of concepts (problem identification, definition, research, assessment criteria), and application of mathematics in the design and execution of the model. In model building, science, and often art, are also involved in the understanding of materials and their behavior, and in understanding and depicting aesthetics.

Physical model



Graphical model



Additionally, curriculum integration must occur when encountering students who are weak in a core area. Sometimes it is necessary to teach basic core subject concepts that some students lack before teaching more advanced concepts in technology education classes. For example, when designing physical models, students typically need to be able to manipulate (add, subtract, etc.) fractions, and this skill may be one they lack when entering technology education classes. As technology education teachers, it is important not to lose sight of the fact that advanced concepts are often being taught. Instructors must do what is necessary to accomplish this and not settle for teaching less just because some students require additional help.

Beyond understanding national, state, and local mathematics and scientific curriculum standards, perhaps the most effective way to design curriculum and instructional activities that are well integrated into a program is to begin a dialog with other core educators. Without a basic understanding of core area curriculum standards, this dialog would not be possible. A relatively easy thing to do is list the math and science concepts your students will need to understand in a given unit or activity and then meet with the math and science teachers to determine how they teach those concepts, considering how to integrate that instruction. Keep in mind that integration of core concepts enforces, supports, and often provides missing meaning to what students are learning in other academic areas. When standards are appropriately taught, students will be better prepared for further education or to take on the challenges of the modern workplace.

Summary

This topic provides a deeper understanding of the most important STEM and core curriculum content projects. This improved understanding can allow teachers to develop more powerful curriculum frameworks, instructional materials, and assessments. Finally, it must be said that the best curriculum design is of little worth without a competent teacher. It is our hope that teachers continue striving to become even more effective by understanding and applying the dispositions, concepts, and skills argued for by the many standards' frameworks.

Procedures in Classroom

Below is a step-by-step process developed to assist teachers in the development of a standards-based curriculum framework. Please note that the development of working models will be used for illustration purposes throughout this section.

1. Identify a specific concept, principle, or skill (we will use the term "concept" to refer to principles and skills throughout the remainder of this section) to be developed into a standards-based curriculum framework. The framework will serve as an outline or blueprint for an instructional unit to be taught. The unit should be in-depth, lasting at least several days, but not longer than a couple of weeks.
2. Next, determine if the concept has been identified as essential, or core, in the content standards being used in the program. For example, a middle school technology teacher within a school system that has adopted *Standards for Technological Literacy* may choose to develop a unit introducing students to working models and modeling. A simple search of the *STL* standards document will reveal that the understanding of models and their development is indeed considered important. Modeling tools and processes is considered one of the core concepts of technology (p. 33) and should formally be introduced in the middle grades (p. 120). The importance and limitations of working models are noted on page 41. Also, *STL* Standard 11, Apply Design Processes, Benchmark J, expects students to make mathematical, graphical, and working models (p. 121). It is clear that working models and modeling are considered an important part of the *STL* standards, but do they meet the criteria established by *NGSS*?
3. Determine if the selected concept meets the above criteria. Continuing with the example, modeling *STL* and *NGSS* standards: models are useful and often necessary in understanding fundamental principles, timeless in their applications, applicable to many situations. Obviously, instructors will want to incorporate models/modeling into the instruction.
4. Next is the question of integration. Examine the core subject standards to determine how a concept is related and if there is a strong argument for authentic integration, versus contrived integration for integration sake only. If a strong relationship exists between the concept and the core standards, plan activities that incorporate the standards to help students learn the concept. Continuing with the example of models and modeling, working (laboratory experiments) models are commonly used in the sciences. It is known that both science and math concepts and principles must be applied in order to design, build, understand, and create working models (i.e., prototypes) in labs.

   At this point the instructor should have identified a valuable standards-based concept that is both worthwhile and engages students in activities of value, and hopefully, demanding of curriculum integration. Now to fill out the rest of the unit's curriculum framework.

5. Use the RBT chart to help you determine what supporting concepts belong in the framework. Virtually all understanding will include remembering factual knowledge. Returning to the concept of working models, students can be expected to know what is meant (definition) by the words "working" and "model." Help them to understand the concepts of working models by explaining why they are important, and perhaps compare working models to conceptual, mathematical, and graphical models. Continuing with the RBT chart, students can certainly be expected to apply working modeling skills. If time permits, ask students to use working models to analyze and critique information, such as when they use an airfoil they have created to better understand lift. It may also be necessary to include other concepts. Often, working models are built to something other than full scale. Of course measurement must be addressed regardless of scale as well as tool use and safety issues. All of these are essential, core concepts and skills that are required in many fields.

A word of caution must be offered here. Most would agree that not all standards are equal. Using the criteria for determining valuable concepts will help to make this determination. To use one illustration, compare the idea of working modeling to *STL* Standard 18, Benchmark I: "Processes, such as receiving, holding, storing, loading, moving, unloading, delivering, evaluation, marketing, managing, communicating, and using conventions are necessary for the entire transportation system to operate efficiently." Is this type of understanding of equal value to that of modeling? There is a finite amount of time to teach students; therefore, how we spend our time and theirs is extremely important and demands our thoughtful consideration.

Appendix

Useful Links

AAAS Science for All Americans: www.project2061.org/publications/sfaa/online/sfaatoc.htm

AAAS Benchmarks for Science Literacy: www.project2061.org/publications/bsl/online/index.php

Core Curriculum Content Standards: www.corestandards.org/

ITEEA's Engineering by Design™: www.iteea.org/EbD/ebd.htm

Next Generation Science Standards: www.nextgenscience.org/

NCTM Mathematics Standards: www.nctm.org/

Revised Bloom's Taxonomy: http://thesecondprinciple.com/teaching-essentials/beyond-bloom-cognitive-taxonomy-revised/

Links to PDFs

ITEEA's Compendium of Major Topics for Technology Content Standards, accessed through *Standards for Technological Literacy: Content for the Study of Technology: Executive Summary*. pp. 5-7. Retrieved from www.iteea.org/TAA/PDFs/Execsum.pdf

ITEEA's *Standards for Technological Literacy: Content for the Study of Technology*: www.iteea.org/TAA/PDFs/xstnd.pdf

ITEEA's STEM Center Responsibility Matrix: www.iteea.org/EbD/Resources/EbDresources.htm):STEM Center Responsibility Matrix 071709.pdf

Aaron Clark, DTE *is a Professor at NC State University.*

Bill DeLuca *is Professor Emeritus at NC State University.*

Jeremy Ernst *is an Associate Professor at Virginia Tech.*

Daniel Kelly *is a Teaching Assistant Professor at NC State University.*

Jerry Ridgeway *is a Technology Education Teacher at Turner Ashby High School.*

resources in technology and engineering

improving regional manufacturing ecosystems:

developing authentic industry-driven design projects

Implementation of an industry-driven design project can expose students to their local manufacturing ecosystem, creating a natural connection between local industries and potential future employees.

by William H. Walls and Greg J. Strimel

Introduction

Workforce development has become a key issue for manufacturers across the United States. According to Deloitte's 2015 Skills Gap Report, manufacturers are not maintaining the workers they need because of factors including, but not limited to, lack of interest in manufacturing careers, shortage of qualified new talent, and retirements in their aging workforce. Additionally, evidence has indicated that students leaving high school lack the employability and technical skills needed to be effective contributors to the manufacturing workforce and regional economic ecosystem (Adecco, 2014). To address these concerns, the authors, with support from the Indiana Next Generation Manufacturing Competitiveness Center, launched the *Improving Regional Manufacturing Ecosystems* (IRME) project, which seeks to advance the next generation manufacturing workforce through high school engineering/technology (ET) programs. The general focus of the project is to build relationships between industry, education, and the surrounding communities to better meet the needs of the local manufacturing ecosystem by gathering/analyzing data to identify regional workforce issues and develop industry-driven design projects that strive to cultivate the technical and employability skills for the next generation workforce. As a result, this article aims to share the IRME project framework as a resource for ET teachers to engage with regional industries and provide an accurate depiction of manufacturing in classrooms, spread awareness of career opportunities in advanced manufacturing, and help cultivate student skills that translate to the school's local manufacturing ecosystem.

Therefore, the authors will provide a rationale for connecting with local manufacturing industries, an overview of the IRME process to identify the needs of local industry and develop authentic, industry-driven design projects, and an example design project and lesson developed following the IRME process. As Strimel (2014) states, authenticity is important for developing any type of transdisciplinary STEM lesson or design project, as it provides students the opportunity to learn from performing tasks that have purpose, are genuine, and have meaning to them and their community.

Why Connecting with Local Industry Can Be Important

A major goal of education is often thought of as preparing young people to be productive members of society. This goal can include developing a high-performance economy by cultivating student skills needed to succeed in the high-performance workplace (The Secretary's Commission on Achieving Necessary Skills, 1991). However, recent skill gap concerns point toward the manufacturing sector, one of the largest drivers of our economy, as not being satisfied by schools, as few students are interested or qualified to meet the projected workforce needs (Adecco, 2014; Deloitte, 2015). According to a 2015 Adecco survey of 500 U.S. business leaders, 92% of those surveyed believe American workers are not as skilled as they need to be to support manufacturing operations. Soft skills, such as communication, creativity, critical thinking, and collaboration are identified as the largest area of concern (44%), followed by technical skills (22%), leadership skills (14%), and software aptitude (12%). Additionally, business leaders indicated one of their most pressing concerns is the manufacturing industry, where unqualified prospective employees increase training needs and raise company costs. Fifty-nine percent of those leaders thought the education system was responsible for the highlighted skills gap.

In addition to the perceived widening skills gap, another national concern is that students are facing increasing university tuitions, higher student loan debt, more competition as they enter the workforce, and lower earnings than previous generations. However, a 2017 report by the Georgetown University Center on Education and the Workforce has revealed there are still 30 million “good” jobs that do not require a Bachelor’s degree and pay an average of $55,000 per year. While the shift away from a manufacturing-based economy has left many blue-collar workers behind, manufacturing industries still hold the majority of “good” jobs that pay well and do not require a Bachelor’s degree (Carnevale et al., 2017). Historically, high school graduates could leave school and obtain a “good” paying job in a local industry. But as the Center on Education and the Workforce (2017) report indicates, “good” jobs that pay well in manufacturing will increasingly require more postsecondary education and training than in the past in an effort to meet competitive requirements and to fully exploit advancing technologies. Therefore, the number of “good” jobs has shifted from only requiring high school diplomas to desiring Associates' degrees, which still generally require a lower monetary investment than a Bachelor’s degree. Nevertheless, as the authors engaged with local companies, they found employment opportunities in manufacturing for students leaving high school that also provide benefits such as tuition reimbursement toward training and postsecondary education. This highlighted a potential career trajectory for students that reduces the debt often tied to pursuing postsecondary education.

Regardless of one’s perspective of education and workforce development, the path to “good” employment and the needs of industries are continuously changing, while public perceptions/awareness of careers can be difficult to transform. Therefore, it becomes important to match education with the demands of new “good” job pathways by connecting with local employers. Through the IRME project, it became apparent that associates within workforce development departments wanted to focus on growing talent, rather than buying it. This concept means that associates believe the company’s best interest is to invest in young, local, prospective employees, rather than relying on paying to bring in external talent. This strategy relies on building a direct workforce pipeline to minimize costs by reducing training time. For example, manufacturing companies across all 92 counties in Indiana have increased the priority for K-12 and postsecondary educators to raise the level of academic performance required for advanced manufacturing careers that are demanding enhanced technology and technical skills. Therefore, developing local connections between ET programs and industry can allow students to see the direct impact the manufacturing ecosystem has on their lives and can allow teachers to work with manufacturers to identify and address regional workforce needs.

Developing Authentic Regional, Industry-Driven Design Projects

The IRME project was initially developed to provide preservice ET teachers with an experience working with local industries prior to their student teaching experiences. The preservice teachers are provided a procedure to conduct research with a local profile company to identify a workforce need/issue and develop an industry-driven, education-related solution for addressing the identified need/issue. Thus, the outcome of the project is to provide a series of educational solutions for improving regional manufacturing ecosystems and provide preservice teachers knowledge of manufacturing industries to inform their future instruction as ET teachers. While the process for developing these educational solutions was created for preservice teachers, the authors believe the procedure can be valuable for inservice teachers as well. Therefore, the IRME process for developing authentic, industry-driven design projects is provided in Table 1, and a sample educational initiative developed through this procedure is discussed in the subsequent section.

An Example Regional, Industry-Driven Design Project

Following the process provided in this article, a preservice teacher immersed himself within a local profile manufacturing company to collect data and identify a specific workforce need. The individual first made contact with the company's training and development department and conducted initial interviews with associates to identify issues and trends regarding their workforce. Through these discussions, it was identified that a shortage of qualified associates in the area of welding could potentially impact their ability to continue to produce quality products at an increasing rate in the future. This demonstrated a practical, immediate, and local workforce challenge. A challenge that seemingly aligns with the national concerns, often highlighted by organizations such as the Mike Rowe Works Foundation (2017), of a widening skills gap, unfilled technical manufacturing jobs, and the remaining belief that a four-year degree is the best path for all people. However, to better understand the identified workforce challenge and frame the problem in a solvable way, the preservice teacher conducted further interviews with welding associates and participated in the company's welding training opportunities.

Through these investigations, the preservice teacher was able to determine the specific technical competencies required of employees, equipment parameters, proper techniques, and safe practices to integrate into a classroom design project. For example, there are five specific welds that a prospective employee must perform to a designated standard before being hired. Additionally, the prospective employees must complete these welds in four required positions found in an authentic manufacturing setting. Figures 1 through 3 illustrate the welds that prospective employees must complete.

While these skills themselves can be important, it also became evident that these specific welding tasks provide numerous connections to engineering and science concepts and practices

Table 1. Procedure for Developing Authentic Regional, Industry-Driven Design Projects

1.	***Locate a profile company.*** Identify a profile company within the local manufacturing ecosystem that is willing to engage with schools and provide training opportunities. The company will represent the customer throughout this development process, meaning the educational solution should be created in tandem with the company to address one of their identified needs (assuming that need is also beneficial to students).
2.	***Contact relevant stakeholders.*** Before reaching out to company stakeholders, it is important to develop an understanding of the workforce and educational challenges in the region. Armed with this knowledge, one can then contact those vested in addressing the regional challenges and request a meeting with the company's workforce development and training department. The purpose of the meeting is to conduct initial interviews with associates to identify issues and trends regarding their workforce and training.
3.	***Conduct use-inspired research.*** Carry out interviews with the profile company's associates to identify issues and trends regarding their workforce and training. Take advantage of the company's existing training programs to gather pertinent information related to workforce development to identify opportunities for improvement, as well as better understand the profile company. Explore the company's current workforce development solutions and observe their pedagogical strategies to document the desired outcomes of the various training programs. Lastly, question employees directly about workforce issues, as they are closest to the various problems and can provide valuable feedback and solution ideas. Be sure to record project ideas during all phases of research and collect any training documentation, as they will be helpful later when deciding on a direction.
4.	***Use findings to frame the problem.*** Problem framing involves developing criteria for success and identifying educational constraints. This process requires general knowledge about the skills gap and the local community's educational system, as well as more specific knowledge on the profile company's problem and the equipment needed to solve it. The problem should be phased in a succinct problem statement detailing the who, what, where, and when associated with the identified problem. The vision for short-term and long-term impact/success should also be detailed.
5.	***Develop an educational solution.*** Using the findings and training materials gathered during research, align potential solution outcomes with the appropriate STEM concepts and standards. Analyze and evaluate the potential design project ideas generated during the data-gathering process, share the ideas with the relevant stakeholders to gather feedback from the customer, and work with the profile company to select a solution idea that addresses the framed problem. Once a solution is chosen, identify what success looks like for each stakeholder and check that the design project aligns with the objectives of the ET program and the profile company's identified need. This should include both short-term and long-term goals focusing on immediate, tangible deliverables for industry, as well as building on a foundation of knowledge for education and industry alike. One way to clearly define success for students is to create a design brief that includes an authentic client, problem statement, and the criteria for success. Lastly, determine what resources (i.e., funds or donated equipment/materials) can be provided to the school from the profile company to support the implementation of the developed solution.
6.	***Implement design project and collect data.*** Data gathered during the implementation of the developed educational solution can be used to check for success as well as provide rationale to justify the continuation of the design project and requests for additional resources from the profile company.
7.	***Reflect with industry stakeholders.*** Did the project meet the criteria for success? Why or why not? Gather feedback from the profile company and reflect on the strong points of the project, as well as the challenges.
8.	***Reiterate.*** Use the feedback to refine the project for future iterations and request additional support from the profile company.

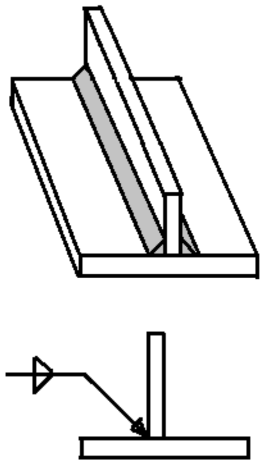


Figure 1. A fillet weld on a T-Joint must be completed by prospective employees at three positions; overhead, horizontal, and vertical downward. *Image credit: Creative Commons, https://commons.wikimedia.org/wiki/File:Fillet_Weld_Notation.png*

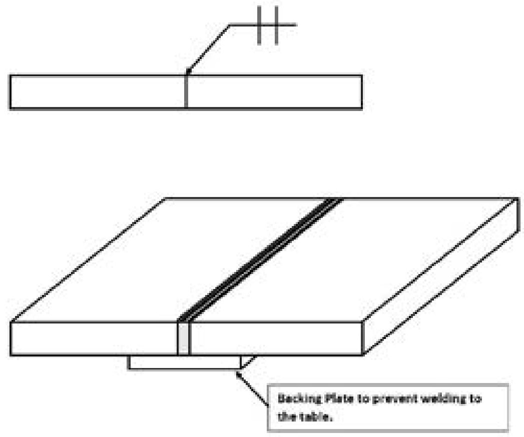


Figure 2. Flat butt joint is a gap weld that fuses two horizontal pieces with a predetermined gap. *Image credit: Author.*

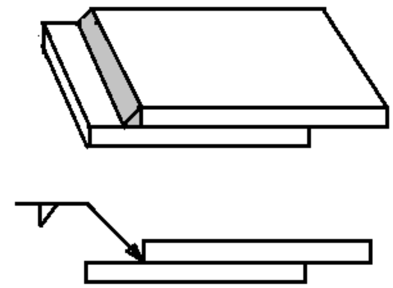


Figure 3. Lap joint welds bond two pieces of metal in which the edges or ends are overlapped. *Image credit: Creative Commons, https://commons.wikimedia.org/wiki/File:Fillet_Weld_Notation.png*

(e.g., material properties, quality control, stress/strain, fatigue, changes in states, electricity, chemical reactions, centroids, etc.). Understanding these connections enabled the development of a class design project that integrates necessary ET course content with authentic technical skills related directly to a local industry. In doing so, students who are unsure of a future career direction can be introduced to potential local manufacturing career options and provided the skills necessary for such jobs without much training. This career awareness can offer a pathway for students to obtain gainful employment after high school and acquire tuition reimbursement benefits to pursue higher education if they decide to do so—thus reducing and possibly eliminating massive amounts of college debt. However, if a student does plan to pursue a degree in engineering or engineering technology, the project still provides them with technical skills that will help to inform their work in the future. For example, the profile company also provides welding training to all new engineers and technologists to ensure they can make proper decisions when designing new products, systems, and processes in their organization.

After the preservice teacher completed the industry research, the scope of the classroom project was defined. This scope consisted of two overarching industry-related outcomes beyond the typical classroom learning objectives: (1) cultivate student employability and technical skills to prepare them for the regional manufacturing ecosystem and (2) improve student awareness/perceptions of career options in local manufacturing industries. Table 2 provides the design project lesson overview, rationale for the overall scope of the project, the specific objectives, and the resources required.

Using the collected information, a design project was developed and framed within the engineering design-based lesson plan format presented by Grubbs and Strimel (2015) (Table 3). The lesson first engages students by touring local manufacturing companies to introduce them to the field of advanced manufacturing, local job opportunities, and potential career trajectories after high school or college. From there, students will explore welding, focusing on common practices (using gingerbread and icing to simulate different weld types or virtual welding training equipment) and understanding the need for welders within the manufacturing ecosystem. Once students understand what welders do, they will learn how welding is used throughout the manufacturing process, as well as the scientific principles involved with welding. The students will then receive the developed design brief (provided in Table 4) that outlines the profile company's need. Student-led design teams will work to complete the challenge, and will be evaluated at completion by an industry stakeholder. The evaluation will be based on meeting established criteria while staying within constraints, technical ability, and demonstrated soft skills like creativity and the ability to communicate an idea.

Conclusion

The procedure and example design project presented in this article demonstrate the way in which industry and ET educators can work together to cultivate local talent, establish a sustainable pipeline of qualified manufacturing employees, and address the widening skills gap in the U.S. While focused on meeting specific industry needs, this experience will provide students with an authentic learning experience that can help foster both the techni-

Table 2. Project Overview

Lesson Purpose	Cultivating high school students' understanding of the impact of their local manufacturing ecosystem, awareness of regional career options in manufacturing, and employability and technical skills to better prepare them for local manufacturing careers.
STEM Standards	*Standards for Technological Literacy*: 3. Students will develop an understanding of the relationships among technologies and the connections between technology and other fields of study. ◦ Benchmark 3F: Knowledge gained from other fields of study has a direct effect on the development of technological products and systems. 12. Students will develop the abilities to use and maintain technological products and systems. ◦ Benchmark 12L: Document processes and procedures and communicate them to different audiences using appropriate oral and written techniques. 19. Students will develop an understanding of and be able to select and use manufacturing technologies. ◦ Benchmark 19Q: Chemical technologies provide a means for humans to alter or modify materials and to produce chemical products.
Global or Local Issue	Addressing the looming manufacturing skills gap. Because of the abundance of cheap labor found around the world, industries are looking for ways to stay competitive in a dynamic economic ecosystem. As the nation shifts toward a service economy, there is a rapidly developing need for hard-working, skilled manufacturing employees. In order to limit outsourcing and downsizing, manufacturing industries need to be competitive in the global market. The skills gap can be seen on a national level as well as in individual communities. Like the profile company, other local manufacturers are not able to replace the employees that will leave, creating a shortage in labor, which affects product quality and efficiency.
Project Objectives	At the conclusion of this lesson students will be able to: ▪ Demonstrate the technical ability to complete the specific types of welds found in an industry setting. ▪ Explain the scientific and engineering principles of welding. ▪ Describe the impact of regional manufacturing ecosystems on their local economy. ▪ Identify manufacturing career opportunities within their community.
Driving Question	What are the opportunities in and skills necessary for advanced manufacturing in my local community?
Materials	▪ Welding Exploration ◦ Gingerbread – Icing – Icing dispensers (Practice Types of Welds) Or ◦ Virtual Welding Trainers (Example: Lincoln Electric VRTEX® Engage™ Welding Education Solution found at www.lincolnelectric.com/en-gb/equipment/training-equipment/vrtex360/Pages/vrtex-engage.aspx ▪ Weld Training ◦ Lincoln Electric MIG Welding Power Source/Wire Feeder/Whip ◦ Wire: ER70S-X .045" diameter ◦ Carbon Steel test plates: 12"x¼" ◦ Wire clippers ◦ Personal Protective Equipment

cal and soft skills that will transfer to any potential manufacturing job. Moreover, the implementation of an industry-driven design project can expose students to their local manufacturing ecosystem, creating a natural connection between local industries and potential future employees.

References

Adecco (2014). *Mind the skills gap.* Retrieved from http://pages.adeccousa.com/rs/adeccousa/ images/2014-mind-the-skills-gap.pdf

Carnevale, A. P., Strohl, J., Ban, C., & Ridley, N. (2017). *Good jobs that pay without a BA.* Washington, DC: Georgetown University Center on Education and the Workforce.

Deloitte. (2015). *The skills gap in U.S. manufacturing 2015 and beyond.* Washington, DC: Deloitte Development LLC. Retrieved from www.themanufacturinginstitute.org/ ~/media/827DBC76533942679A15EF7067A704CD.ashx

Grubbs, M. & Strimel, G. (2015). Engineering design: The great integrator. *Journal of STEM Teacher Education, 50*(1), 77-90.

Mike Rowe Works Foundation. (2017). *Profoundly disconnected.* Retrieved from http://profoundlydisconnected.com/

Strimel, G. (2014). Authentic education by providing a situation for student-selected problem-based learning. *Technology and Engineering Teacher, 73*(7), 8-18.

The Secretary's Commission on Achieving Necessary Skills. (1991). *What work requires of schools: A SCANS report for America 2000.* Washington, DC: U.S. Department of Labor.

Table 3. Project Lesson Plan

Engage: ***Industry visit and introduction to the local manufacturing ecosystems.*** Students will learn about the career opportunities in manufacturing, while interacting directly with the local manufacturing ecosystem. College may not be the most prudent next step for all students, so the engagement can provide awareness of potential benefits such as pay, healthcare, 401k, and tuition reimbursement associated with manufacturing careers.
Explore: ***Focus on a featured profession in manufacturing (welding).*** Students will explore the profile company's five different required welds by using gingerbread cookies and icing to practice each weld. If the resources are available, virtual welders can also be used to engage students without the safety concerns or need for real welding equipment.
Explain: ***Science of Welding.*** The teacher will cover the history, applications, and practice of welding. This includes safety, common practices, and quality assurance strategies. The science of welding offers a practical demonstration of concepts found in chemistry and physics. In chemistry, welding illustrates concepts regarding material properties, fusion, and the changing state of matter. For physics, welding offers a scientific inquiry opportunity to cover difficult mathematical concepts through stress testing, as well as the electrical principles that allow welding to work. Examples of driving questions for inquiry: ▪ What is the purpose of the shielding gas? ◦ Shielding gas prevents the metal from oxidizing. ▪ Why do different materials require different settings? ◦ Heat transfer, conductivity, and malleability dictate the settings. ▪ How does Ohm's Law apply to welding? ◦ MIG welding uses an electrical arc to superheat the metal. This is why it necessary to ground the work in order to weld the metal. Ohm's Law can be used to illustrate the effects of changing the amperage and feed rate. ▪ How is welding different than adhesive? ◦ Fusion melts the material, changing the structure on a micro-level and creating a stronger bond than the original 2 base materials.
Engineer: Teacher will introduce industry-driven kinetic art design project. (See Design Brief in Table 4)
Evaluate: A master weld technician will evaluate all of the student designs as each team presents kinetic art pieces in a gallery walk. Students will answer questions and defend their design, giving the teacher an opportunity to evaluate the soft skills used during the project.

Table 4. Design Brief

Design Brief: Kinetic Art Commission

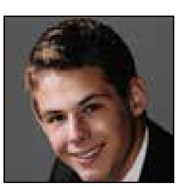
Objective:

The [profile company] wants to make an investment in high school education and has commissioned this class for a project. Identifying a shortage in welders, it has created a design challenge that incorporates the same skills it looks for in potential associates. Students must work in teams to design and create an original piece of kinetic art. Kinetic art utilizes outside forces to create movement. The design will incorporate the welds that associates must demonstrate to be hired as welders. The company plans to select one team's design and use it for public relations and community engagement. Designs must meet the criteria for success found below, and will be evaluated on weld quality, creativity, functionality, and aesthetic appeal.

Criteria for Success:

- Observable movement
- Power derived from an outside force (wind, water, sun)
- Incorporates Carbon Steel, Stainless Steel, and Aluminum
- Aligns to detailed design plans developed using the appropriate industry-standard conventions
- Incorporated the following welds:
  - Horizontal Fillet
  - Lap Joint
  - Butt Joint
  - Overhead Fillet
  - Vertical Downward Fillet

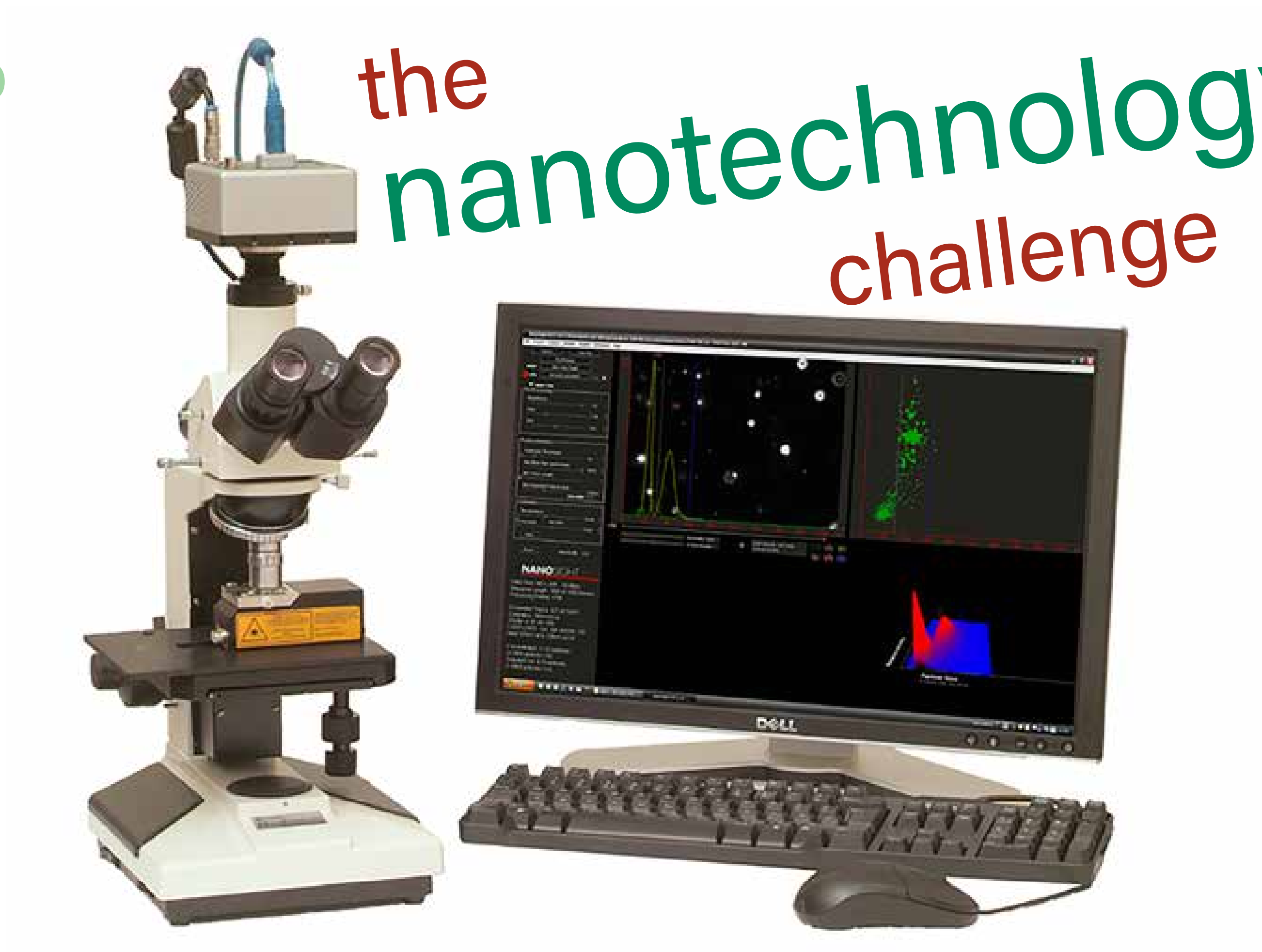
Image credit: Wikimedia Commons, https://commons.wikimedia.org/wiki/File:Eos_xk_(3),_1965.JPG



William H. Walls *is an undergraduate researcher in the Engineering/Technology Teacher Education program in the Purdue Polytechnic Institute at Purdue University. He can be reached at wwalls@purdue.edu.*

Greg J. Strimel, Ph.D., *is an assistant professor of Engineering/Technology Teacher Education in the Purdue Polytechnic Institute at Purdue University. He is also a project coordinator for the Advancing Excellence in P-12 Engineering Education project. He can be reached at gstrimel@purdue.edu.*

the nanotechnology challenge



Introduction

It seems there are powerful advances in this exciting field almost every day, so why not challenge your students to think seriously about applying nanotechnology? Here we shall discuss the design considerations surrounding a nanotechnology medical application where small machines are used to clean the arteries of the human body, removing plaque buildup and preventing dangerous and even fatal blood clots in the heart and brain. Below are some of the many concerns design engineers would have to take into account as well as how it all relates to STEM thinking. Get your students ready to think about the human body and how nanotechnology could be applied to its care and rejuvenation.

The Challenge

Certainly some research by the students in this technology is recommended. However, for the purpose of this activity, we shall assume the technology to inject or introduce nanotechnology machines or robots into the bloodstream of human beings already exists. In this exercise, we are going to explore the process and ramifications of this technology, something that intimately reflects upon its success.

How might patients who could make use of this technology be evaluated and selected as candidates for such treatments? The protocols for using technology and introducing chemical treatments into the body are often as important as the drugs and medicines themselves. The rate of a drug dosage and its impact on the body is intensively evaluated. How might this be for nanotechnology; how many nanobots at a time should be introduced; how might they be retrieved later; and could they become a problem if allowed to navigate inside existing arterial plaques? These are not small concerns. If nanobots can dissolve or operate successfully on the plaques, where do the dissolved or processed materials go?

Nanobots might incorporate fixed or modifiable programming to guide their behavior. They might be programmed to act individually or in groups—swarms as the technical literature states. As nanobots are active in the human body, are they steerable or capable of being communicated with? If a patient has an electronic package (pacemaker) to stimulate heart rhythm already installed in his body, would communicating with the nanobots interfere with the installed pacemaker—in a similar way that pacemaker wearers must exercise caution around microwave ovens?

by
Harry T.
Roman

As the patient is undergoing nanobot therapy, there could be a variety of possible scenarios:

1. Should they be monitored online in real time in a special room with technicians/doctors immediately available?
2. Can they resume normal activities and come back days later for follow-up evaluation?
3. Should they rest comfortably in a hospital setting for several hours/days so monitoring can be done periodically?

Challenge your students to think about this, the risks involved, and the liabilities a hospital/doctors would assume during this form of treatment.

An important concern when introducing foreign materials into the body is the body's reaction to them chemically as well as by the natural immune response to their presence. There is a rich literature about how artificial implants have been attacked by the body and badly deteriorated, or how patients have been sickened by the presence of certain materials. Things as relatively simple as breast implants have triggered serious concerns and reactions. Ask your students to explore this area and consider how they would select potential patients for nanobot treatment.

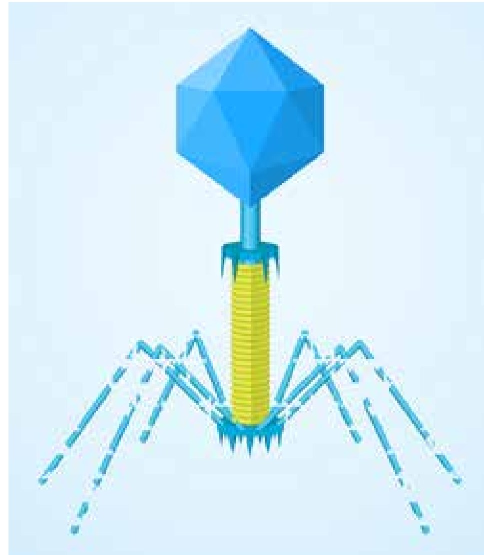
Nanotechnology might use an interesting array of molecular-size materials that could interact with blood at the molecular level, potentially interfering with oxygen transport in the blood cells or causing unusual chemical conditions within the blood stream. Has this been studied before; is there literature available concerning this; how is it different than injecting other chemical compounds into the human body? Gold injections were used for many years to treat joint problems, and gold is a metal. Are nanobots composed of metal likely to be different? How about nanobots composed of nonmetallic materials like silicon or plastics?

The nanobots we are conceptually discussing will obviously need a power source to keep them going. It could be a tiny battery-like power source, or maybe it feeds off the blood supply in which it is "swimming." How does this impact the patient? What would be the lifetime of the energy source and hence the nanobot? Waste by-products of the tiny power sources—where does it go?

Everything in the blood is ultimately filtered through the kidneys, with their delicate structures. Can potential irregularly shaped nanobots introduce major concerns here? What happens if several nanobots pile up, collide, or bunch together—could this cause a clot? Perhaps nanobots should be restricted to certain parts of the bloodstream, and how might that be done?

Seeking Outside Assistance

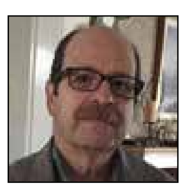
This is a complex challenge, and students might benefit from having medical professionals provide some input and guidance to help get their arms around such an application. This is very similar to project managers in business who use technical consultants when designing new technologies and evaluating the ramifications of possible applications. Here is a list of some medical sources of possible guidance:



- The nurse in your school may be able to suggest some possible contacts.
- Check with your librarian/media expert for access to medical information and specialized technical information.
- Colleges/universities that have biomedical or robotic courses of study may provide students/professors who could speak to your class(es).
- Hospitals that offer arterial plaque treatments could be a direct source of information, and a speaker might be available or even a field trip to their facilities might stimulate meaningful discussions.
- Check patent resources on the internet to assess what may already be under consideration in the nanotechnology field.

Stretch student thinking through the brainstorming of other ideas for nanobot medical applications. This is an application area that is advancing rapidly. Could they be injected into joints to clean up injured areas? How about injected into tumors to "chew" them away, or administer cancer-fighting drugs to destroy them? Might they be used in deep wounds to apply antibiotics and promote healing, or could they slowly dissolve into wounds and be part of the new "mesh of flesh" that eventually closes over and heals the wound?

Ramp up your students' generation of new ideas. See how it helps them identify and assess nanobot arterial plaque cleaners. Some students might find it stimulating to write a short story/sci-fi story about nanobots in the human body. Get those ideas and concerns with the technology out in the open for discussion!



Harry T. Roman *is a retired engineer/inventor and author of technology education/STEM books, math card games, and teacher resource materials. He can be reached at htroman49@aol.com.*