csta Voice

Computational Thinking

The Voice of K–12 Computer Science Education and its Educators

Volume 12, Issue 1 **March 2016**

Inside This Issue

FEATURES

CSTA Gets a Makeover
It's Time to Vote
Make Waves with CSTA in San Diego
Administrator Impact Award – Act Now!
Reclaiming Roots of CT
CT Driving Computing Curriculum in England

COLUMNS

CSTA Congratulates Members
Chapter Success
Classroom Strategies
CT in Action
CT Resources
Meet the Candidates

INFO BRIEFS

Keep Up with CSTA!
Staff
Contribute to the Voice
Contact Info
Meet the Authors
Mark Your Calendar

IN THE NEXT
VOICE
Professional Development

CSTA Gets a Makeover

Lissa Clayborn

Dr. Mark R. Nelson, CSTA's Executive Director, describes exciting changes in computer science (CS) education and for CSTA in a recent blog post. See 2016: The Year of CS Education (*blog.csta.acm.org/2016/01/30/2016-the-year-of-cs-education*).

The CSTA staff and Board have been busy working on many of these changes to provide greater benefits for you, our members. One such initiative is a new website. Not only is the look more up-to-date, mobile-friendly, and easier to navigate, it will also include access to new member-only features. The website is integrated with new association management software that will enable many more ways for members to communicate and participate in the organization. It will be easy for members to volunteer in activities of interest to them and access tools that support the chapters' advocacy and outreach efforts.

The alpha version of the new website, *www.csteachers.org*, is now live. We encourage you to explore the new look and navigation. Over the next six months, we will be updating and adding to the site, including rebranding the site and the rest of the organization.

New features, including member login, will be integrated as soon as they are available. In the meantime, we will keep both sites running so that you can still have access to the currently available member resources. Check back often to watch the progress. We all look forward to sharing this exciting time and transition with you!

It's Time to Vote

In this issue, we are pleased to announce the ten candidates for the five open 2016–2018 CSTA Board positions. Read their personal statements on page 10.

The CSTA Board of Directors consists of eleven voting representatives, elected by the more than 22,000 CSTA members worldwide. In addition to working with the Executive Director in setting the organization's direction, Board members carry out many of CSTA's operational tasks through committees and task forces.

As in past years, the election will take place online using the ElectionBuddy voting system. On February 22, all current CSTA members should have received an email from ElectionBuddy with a personalized link to the ballot. If you have not received your email, first check your spam filter, then contact *customerservice@csteachers.org*. Voting ends March 22.

This is an exciting year for computer science (CS) education. President Obama announced his CSForAll initiative, and CSTA is in the forefront of the effort to support K–12 teachers and bring CS to classrooms across the country (and around the world). We want your voice to be heard. Vote today!

Dave Reed
Chair, CSTA Board of Directors

Keep Up with CSTA!

csta.acm.org

Twitter
(@CSTeachersA
and #csta16)

Facebook
(Computer Science
Teachers Association)

LinkedIn
(search for computer science
teachers association)

Visit CSTA Online

Your CSTA website is changing!

A new...
Look,
Feel,
&
Member experience

www.csteachers.org

Make Waves with CSTA in San Diego

Conference Chairs: J. Philip East and Stephanie Hoeppner

Come make waves with CSTA during the 2016 CSTA Annual Conference in sunny San Diego, CA, July 10–12. We'll celebrate 15 years of CSTA in the biggest conference ever—three full days and eight strands of the high-quality PD you've come to expect from CSTA.

The planning committee selected and recruited presenters to offer a wide variety of topics for the diverse needs of our membership. There will be something for every teacher and administrator interested in K–12 CS education.

The 2016 program will include some exciting new elements while continuing the best from past conferences.

We've expanded the workshops. There will be three session with more choices during each slot, including the best from our sponsors. The workshops are first-come, first-served. Space is limited. Register early.

Among the many session topics you'll find:

- Advanced Placement Computer Science
- Computational thinking
- K–8 topics
- Programming
- Robotics
- STEM

We've added new elements too.

- A brand new strand for principals and administrators.
- Expanded exhibitors' hall. More vendors and demonstrations will be ready to show you the latest technologies and classroom tools.
- Birds of a feather sessions. You told us what you wanted and we listened.
- Lots of time for networking. Catch up with old friends and make new friends–always a highlight of the conference.
- Social Events. You are sure to enjoy the fun events we've planned.

Innovative, inspiring, and collegial features, keynote speakers, hands-on workshops, one-hour presentations, mini-sessions, door prizes, and lots of fun await you this summer in San Diego.

Make plans now for July 10–12, 2016. Registration is open, but early-bird registration ends on April 1 (save $50).

Complete details, registration link, and housing information are available at the conference site (*cstaconference.org*).

Administrator Impact Award — Act Now!

CSTA is currently accepting nominations for the Administrator Impact Award. Teachers, community members, and even students can nominate an administrator in their school or district who has shown dedication and commitment to improving CS education.

Nominations close on March 15, 2016

www.csteachers.org/adminimpact

CSTA Voice ISSN: 1555-2128

CSTA Voice is a publication of the Computer Science Teachers Association.

CSTA Voice is a quarterly publication for members of the Computer Science Teachers Association. It provides analysis and commentary on issues relating to K–12 computer science education, resources for educators, and information for members. The publication supports CSTA's mission to promote the teaching of computer science and other computing disciplines.

Change of Address and Membership Questions: Contact Member Services via email at customerservice@csteachers.org or call 1-212-626-0500 (Global).

Criteria for submitting articles: Potential writers for CSTA should send a brief description of the proposed article, estimated word count, statement of value to members, author's name & brief bio/background info, and suggested title to the editor at cstapubs@csta.acm.org. The final length, due date and title will be negotiated for chosen articles.

Notice to Authors Contributing to CSTA Newsletter: By submitting your article for distribution in this publication, you hereby grant to CSTA the following non-exclusive, perpetual, worldwide rights:

- to publish in print on condition of acceptance by the editor
- to digitize and post your article in the electronic version of this publication
- to allow users to copy and distribute the article for noncommercial, educational or research purposes

However, as a contributing author, you retain copyright to your article and CSTA will make every effort to refer requests for commercial use directly to you.

Reclaiming the Roots of CT

Irene Lee

There has been much debate about the definition of computational thinking (CT) and the relative merits of different definitions. In this article, I argue for a focused definition of CT that clearly distinguishes it from other forms of thinking.

CT was popularized by Jeannette Wing in 2006 as the "thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent." Lee and Martin, CSTA CT Task Force co-chairs, further simplified this definition in 2015 to "CT refers to the human ability to formulate problems so that their solutions can be represented as computational steps or algorithms to be carried out by a computer." For the remainder of this article this will be referred to as the "thought processes" definition of CT.

The Computing at School (CAS) curriculum supports this description of CT. It explicitly states, "the thinking that is undertaken before starting work on a computer is known as CT" (*barefootcas.org.uk*). Further, in UK's Computing at School's "Computational thinking: A guide for teachers," CT is clearly described as "a thought process, not the production of artefacts or evidence" (*community.computingatschool.org.uk/files/6695/original.pdf*).

Abstraction, automation, and analysis, the three pillars of CT as described by Cuny, Snyder, and Wing in 2011, have been observed in students as young as middle school.
Students have demonstrated that they can develop abstractions and automations as they study and solve real-world problems in modeling and simulation projects and robotics projects. Key to this definition is that CT takes place when students are "looking at a real-world problem in a way that a computer can be instructed to solve it." In the context of modeling and simulation, students were actively engaged in CT when they selected features of the real world to incorporate into their models (abstraction), determined which elements of the model need to be updated as simulation time advanced (automation), and analyzed the model's inclusion the features necessary to mimic the real world (analysis).

One of the merits of the "thought processes" definition is that it is very specific to humans harnessing computers as information processing devices. This specificity makes it different from critical thinking, mathematical thinking, and scientific thinking. A difficulty with this definition is that a teacher or student new to computer science (CS) may not be able to relate to this definition of the practice. How does one formulate a problem and its solution so that it can be carried out by a computer if one does not know what a computer is capable of doing and how to give a computer instructions? What differentiates a poor formulation from a strong one?

Thus, after Wing's definition was publicized, other groups published their own interpretations of CT, including the ISTE/CSTA Operational Definition, "Computational Thinking Practices" (*AP CS Principles* and *Exploring CS*, 2012), "CT concepts, practices, perspectives" (Brennan & Resnick, 2012), "CT Patterns" (Repenning, 2012), and "CT" (*Exploring CT*, Google, 2014).

Common among these definitions is an expansion of CT to include many other practices. For example, the ISTE/CSTA "operational definition of CT" was constructed to aid teachers in seeing themselves as already teaching skills that are components of CT.

While well-intentioned, the ISTE /CSTA operational definition has caused confusion. It is not well understood that an "operational definition" is intended to be a definition of the operations that make up a practice. Each operation is a part of the larger practice but does not by itself equal the practice. Thus the operational definition of CT describes various operations that make up CT but conducting a single operation does not equal "doing CT."

Unfortunately, all too often, the interpretation of the operational definition is that if you are doing any one of the listed operations, you are a computational thinker. This is not correct. For example, logically organizing and analyzing data is an operation described in the operational definition of CT. But a student who organizes data, without consideration of how a computer program would direct a computer to read in, store, and manipulate the data, is not doing CT.

In a similar vein, the AP CS Principles and Exploring CS curricula describe "CT Practices" that extend beyond the original "thought processes" definition of CT. The CT Practices include "communicating computational thought processes" and "collaborating with peers on computing activities." While these are both valuable practices in CS education, they are not necessarily part of "formulating a problem and

It's Election Time

CSTA BOARD OF DIRECTORS CANDIDATES

9 –12 Representative Candidates

Stacey Kizer
Chinma Uche

At-Large Representative Candidates

Myra Deister
Michelle Lagos

International Representative Candidates

Miles Berry
Michael Jones

State Department Representative Candidates

Anthony Owen
Doug Paulson

University Faculty Representative Candidates

Darcy G. Benoit
Fred G. Martin

Read their personal statements on page 10

its solution so that the solution can be carried out by a computer."

The expansion of definitions of "CT" and definitions of "CT Practices" have led to an erosion of the integrity of the "thought processes" definition of CT. Some have come to believe that CT means everything and, consequently, nothing at the same time.

Furthermore, the two terms "computational thinking practices" and "computational thinking" often get conflated (or taken to mean the same thing). In some circles, CT has come to encompass "everything people think kids should learn in CS," including the iterative development of software artifacts.

This losing of the original definition of CT has serious ramifications. 1) We lose what is special about CT—that the human is formulating a problem and its solution so that the solution can be carried out by a computer (not by a human); 2) CT can be viewed as any task that involves students thinking while on a computer —troubleshooting hardware involves thinking and computers, is it computational thinking? and 3) We lose sight of the power of CT to study and solve real-world problems. If a student is doing CT by writing and debugging some code, why go further and address real-world problems?

I believe that CT is a skill that is developed through repeated exposure to how real-world problems are represented, studied, and solved using computers as information processing devices, and progressively deeper understanding of what computers are able to do and how to instruct them. Students can develop CT skills through opportunities to map real-world problems into abstractions and algorithms that can be represented and operated upon on a computer.

ISTE/CSTA "Operational Definition of Computational Thinking for K-12 Education" (2011)

1. **Formulating problems in a way that enables us to use a computer and other tools to help solve them**
2. **Logically organizing and analyzing data**
3. **Representing data through abstractions such a models and simulations**
4. **Automating solutions through algorithmic thinking (a series of ordered steps)**
5. **Indentifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources**
6. **Generalizing and transferring this problem-solving process to a wide variety of problems**

CT Driving Computing Curriculum in England

John Woollard

Computational thinking (CT) has come to the fore for many teachers in England with the advent of the new *National curriculum in England: computing programmes of study* in September 2013 (*goo.gl/SklB9O*). It is explicitly and thoroughly embedded in the curriculum for K–12. The first sentence states, "A high quality computing education equips pupils to use computational thinking and creativity to understand and change the world."

CT lies at the heart of the computing curriculum but it also supports learning and thinking in other areas of the curriculum. CT gives a new paradigm for thinking about and understanding the world more generally. Simon Peyton-Jones, chair of Computing At School (CAS), succinctly explains why learning computer science (CS) and CT are core life skills, as well as being eminently transferable, in a talk filmed at TEDx-Exeter (*bit.ly/13pJLCR*).

CT skills are the set of mental skills that convert "complex, messy, partially defined, real-world problems into a form that a mindless computer can tackle without further assistance from a human," the Chartered Institute for IT (*bit.ly/1Li8mdn*).

In the UK, the term CT has been described in different ways for different audiences but there is a growing consensus that CT is a cognitive or thought process involving logical reasoning by which problems are solved and artifacts, procedures, and systems are better understood. It embraces:

- the ability to think algorithmically;
- the ability to think in terms of decomposition;
- the ability to think in generalizations, identifying and making use of patterns;
- the ability to think in abstractions, choosing good representations; and
- the ability to think in terms of evaluation.

CT skills enable pupils to access parts of the computing subject content. Importantly, they relate to thinking skills and problem solving across the whole curriculum and through life in general.

Where these thinking skills are being promoted we see the pupils adopting approaches to problem solving such as tinkering, creating, debugging, persevering, and collaborating. These are key features associated with successful learning in computing and across the curriculum. Computing, computer programming in particular, enables tinkering to occur. Learners are genuinely learning through trial and improvement. We all know that perseverance is necessary when debugging programs and we appreciate the reward and feeling of satisfaction when creating and collaborating.

A number of techniques can be employed to enhance CT. Think of these as "computational doing," the computing equivalent of "scientific methods." They are the tools by which CT is operationalized in the classroom, workplace, and home: reflecting, coding, designing, analyzing, and applying. These techniques enable CT skills to be developed.

Reflection is the skill of making judgements (evaluations) that are fair and honest in complex situations that are not value-free. Within CS this evaluation is based on criteria used to specify the product, heuristics (or rules of thumb), and user needs to guide the judgements.

An essential element of the development of any computer system is translating the design into code form and evaluating it to ensure that it functions correctly under all anticipated conditions. Debugging is the systematic application of analysis and evaluation using skills such as testing, tracing, and logical thinking to predict and verify outcomes.

Designing involves working out the structure, appearance, and functionality of artifacts. It involves creating representations of the design, including human readable representations such as flowcharts, storyboards, pseudo-code, systems diagrams, etc. It involves activities of decomposition, abstraction, and algorithm design.

Analyzing involves breaking down into component parts (decomposition), reducing the unnecessary complexity (abstraction), identifying the processes (algorithms), and seeking commonalities or patterns (generalization). It involves using logical thinking, both to better understand things and to evaluate them as fit for purpose.

Applying is the adoption of pre-existing solutions to meet the requirements of another context. It is generalization—the identification of patterns, similarities and connections—and exploiting those features of the structure or function of artifacts. An example includes the development of a subprogram or algorithm in one context that can be re-used in a different context.

Computing At School, as a grass-roots and free teacher-membership organization, has been at the forefront of advising on the changes to the curriculum and in providing much needed support to both primary and secondary teachers faced with the challenge of bringing into being a new subject in UK schools.

LEARN MORE:
CT: A guide for teachers: *www.computingatschool.org.uk/news_items/26*
CAS Barefoot: CPD for K–5 teachers: *barefootcas.org.uk*
CAS Tenderfoot: CPD for 6–12 teachers: *www.computingatschool.org.uk/tenderfoot*
CAS Community: *community.computingatschool.org.uk*
CAS Resources for computational thinking: *community.computingatschool.org.uk/resources/2324*
CAS Network of Excellence: *community.computingatschool.org.uk/resources/802*

Meet the Authors

Lissa Clayborn
Deputy Executive Director/Chief Operations Officer, CSTA
Lissa has worked for over 20 years in the nonprofit educational technology sector, including ISTE.

Daryl Detrick
Warren Hills HS, NJ
Daryl is a CS educator and past President of CSTA Central NJ and is Co-chair of the CSNJ.

J. Philip East
University of Northern Iowa
Philip has been teaching computing for over 35 years. He is the Program Chair for the CSTA Annual Conference.

Shuchi Grover
SRI International
Shuchi is a research scientist at SRI's Center for Technology in Learning and a member of the CSTA CT Task Force.

Stephanie Hoeppner
Williamsburg Local Schools, OH
Stephanie has taught CS for 16 years. She has served on the CSTA Board of Directors and is the Workshop Chair for the CSTA Annual Conference.

Joe Kmoch
CSTA Wisconsin
Joe is a retired CS teacher now working as a CS consultant. He is a member of the CSTA CT Task Force.

Irene Lee
MIT
Irene is a research scientist in the Scheller Teacher Education Program and Education Arcade. She serves as a Co-chair of the CSTA CT Task Force.

Fred G. Martin
University of Massachusetts Lowell
Fred is a CS professor. He serves as a Co-chair of the CSTA CT Task Force.

John Woollard
CAS
John is a leading member of Computing At School in the UK. He serves as the Chair of the Assessment working group and Coordinator of the Tenderfoot project.

CSTA Congratulates Members

Champions of Change for Computer Science Education

CSTA members were among those recently honored by the White House for empowering their communities as "Champions of Change for Computer Science Education." We're so proud of you! Learn more about the award at: *www.whitehouse.gov/champions.*

Andreas Stefik

Andreas Stefik, Ph.D., is an assistant professor of Computer Science (CS) at the University of Nevada, Las Vegas. For the last decade, he has been creating technologies that make it easier for people, including those with disabilities, to write computer software. With grants from the National Science Foundation, he established the first national educational infrastructure for blind or visually impaired students to learn CS. He is the inventor of Quorum, the first evidence-oriented programming language. The design of Quorum is based on rigorous empirical data from experiments on human behavior.

"I'm continuing work on the Quorum programming language. We are increasingly investigating how language designs impact people at a variety of levels. My long-term hope is to make a programming language as easy to use as I can by using the scientific method and experiments, while also viable in practice to be used in the field."

Jane Margolis

Jane Margolis is a researcher at the University of California, Los Angeles Graduate School of Education and Information Studies, where she investigates why so few women and students of color have learned CS. Based on research discussed in her books *Unlocking the Clubhouse: Women in Computing* and *Stuck in the Shallow End: Education, Race and Computing*, she and her collaborators, with support from the National Science Foundation, created *Exploring Computer Science* (ECS), a high school curriculum and teacher professional development program committed to reaching all students, especially those in underserved communities and schools. ECS now exists across the nation, including in seven of the largest school districts.

"I would like to be able to give some concentrated attention to researching students' learning and participation in the ECS classroom. Are the classrooms welcoming and engaging for all students? What makes them so? And, how do we capture and tell the stories of students' learning of CS content and the creative, inquiry, problem-solving practices that we hope for all students in the ECS classroom?"

Karen North

Karen North is a retired CS and math teacher from Houston, TX, and has been an advocate for CS education since 1985. She has fought to keep CS certification for teachers and played an integral part in increasing programming and computational thinking in the K–8 Texas math standards. She now serves as a Code.org affiliate, a Code Buddy for Spring Branch Independent School District, and a volunteer with the American Association of University Women (AAUW), the National Center for Women & Information Technology (NCWIT), and the International Society for Technology Education (ISTE) Computing Teachers Network, among others.

"In Texas, I will continue my work to change policy that can bring CS to every school. Nationally, I will scale my outreach with AAUW in 'Solving the Equation' to motivate girls to study CS through their unique passions and NCWIT in Aspirations in Computing."

Chapter Success

Advocacy Success in New Jersey

Daryl Detrick

Two very important bills related to computer science (CS) education were recently signed into law by New Jersey Governor Chris Christie, and one is working its way through the legislature.

Bill A2597 provides that, beginning with the 2016-2017 grade-nine class, an Advanced Placement CS course may satisfy a part of the mathematics credits required for high school graduation (*www.njleg.state.nj.us/2014/Bills/A3000/2597_R2.PDF*).

Bill S2030 requires the Department of Education (DOE) to review Core Curriculum Content Standards to ensure that they incorporate modern CS standards where appropriate (*www.njleg.state.nj.us/2014/Bills/S2500/2032_R2.PDF*). This bill originally required NJ DOE to add CS standards, but was conditionally vetoed by governor. The passed bill is based on the conditions of the veto.

We were able to work with NJ DOE to get some computational thinking standards in NJ technology standards while this bill was being considered. The hope is that this bill will enhance those standards as well.

Getting these bills passed was not easy or fast. In the summer of 2014, about eight members of CSTANJ (we now have 3 chapters) created an informal advocacy group called CSNJ. The first thing CSNJ did was to evaluate the state of CS education in NJ and establish goals. Our four major goals were: 1) add CS standards; 2) make CS count as a math or science credit; 3) create a CS teaching endorsement; and 4) the ultimate goal that all students have the opportunity to take CS courses.

We began by identifying the various stakeholders in NJ who could help us achieve these goals, including legislators, NJ DOE, professional associations, business leaders, educators, universities, and colleges.

The next phase was to reach out to as many of the stakeholders as possible, which meant attending statewide conventions for teachers, administrators, and boards of education. We also presented at various conferences across the state and made presentations to community groups. It took time to make some breakthroughs, but after numerous conversations and many, many emails, we were able to create connections within NJ DOE and with legislators who supported our cause.

The process had lots of ups and downs, but the key was to maintain consistent open lines of positive communication without being too pushy. We also received lots of support and direction from CSTA and Code.org along the way. It took nearly two years to pass the bills that the governor signed in January. Our first two goals have been accomplished.

Bill A4327, which creates a CS teaching endorsement, is our next goal. We did not have time to get it through the last legislative session, but it will be reintroduced in the next few weeks (*www.njleg.state.nj.us/2014/Bills/A4500/4327_I1.PDF*).

Unfortunately, too many students in NJ still do not have the opportunity to take CS classes, but we have made steps in the right direction and we aren't losing sight of that goal.

Classroom Strategies

A Baker's Dozen of Teaching Tips

Shuchi Grover

Block-based programming tools (i.e., Scratch, Alice, Snap!, App Inventor, Blockly, and AgentSheets) are increasingly being used to teach children programming in informal settings, but also to introduce children to computer science (CS) constructs in formal K–12 classrooms.

What these tools afford, in terms of social participatory learning, creative expression, motivation, agency, and choice, is invaluable. Recent advances in learning theories underscore the need to attend to these socio-cultural and socio-emotional aspects of learning.

As decades of research have shown (and teachers probably know from experience), the reality is that most children need help with understanding computing constructs and the computational problem-solving process. Often, they end up using programming constructs without really understanding the purpose in the program, arriving at solutions through trial and error; using simpler concepts more often than others in open-ended projects; and struggling with those that are computationally challenging to use (such as variables or Boolean logic).

Although interest-driven activity remains important, K–12 curricula aim to teach, assess, and integrate problem-solving skills, CS concepts, and computational thinking practices.

For this, I present a baker's dozen of strategies that are informed by my own journey, as a researcher who is guided by literature in the learning sciences and computing education research, a leader of programming and robotics in afterschool settings, and as a middle school CS teacher. These will help teachers balance interest, creativity, exploration, and "learning by doing" that are the hallmarks of informal, free-choice settings, with pedagogical strategies to effectively promote deeper learning.

#1 Situate programming tasks in students' interests. Whether these are examples demonstrated to students, programming tasks, or open-ended projects, draw ideas from students' interests—games, art, interactive stories, pop culture, or school/community contexts.

#2 Use guided exploration. Introduce a problem and provide a framework for guided exploration and tinkering for solutions. Following it with #3 will foster better learning.

#3 Model algorithmic thinking. This is somewhat similar to the use of "worked examples" that has been shown to reduce cognitive load when teaching complex conceptual ideas. Making mistakes during this process is awesome! It helps drive home the idea that nobody codes everything right the first time and provides opportunities to show the process of debugging. Show that there may be more than one way to program a solution. Also, model the process of testing code with various inputs, code tracing, and using computing terminology in context.

#4 Use a range of programming activities. "Learning by doing" is a key pedagogy in learning to program, but it is sometimes the only kind of activity used, and often without appropriate guidelines necessary for optimal learning. Programming activities can range from close-ended tasks with a fixed end state (or goal), semi-structured activities that include guidelines of constructs to be used, to completely open-ended projects. Don't forget to provide rubrics.

#5 Assign "buggy" tasks. Debugging is a key skill for programmers; it involves complex reasoning and is not often explicitly taught. You can re-use students' faulty code samples for such tasks.

#6 Make students read code and make sense of the program. Code tracing and the ability to write code go hand in hand.

#7 Make them plan before they program. Planning is an essential step most students dislike. Writing out the steps in natural language or pseudocode or storyboarding is part of the problem-solving process. Having an algorithmic solution before coding also drives home the importance of computing as problem solving, not simply programming.

#8 Describe a problem solution in natural language or pseudocode. It also helps students "see" algorithmic steps and flow of control outside of the context of the programming environment. Explain code in natural language and encourage students to do the same. You could project a segment of code and have students describe what it's doing and why it does or does not work.

#9 Use different representations of the solution state. In addition to pseudocode, provide opportunities for students to express the same or similar solutions in different programming languages or environments over the course of a term.

#10 Use a mix of collaborative and individual projects. Students benefit from collaborative problem solving, but in order to ensure all learners are learning all concepts, have them work on some tasks individually.

#11 Use professional vocabulary. Don't shy away from calling an algorithm an algorithm. As students slowly learn the shared language of the community of professionals in the discipline, they tend to feel more empowered and connected with it.

#12 Provide opportunities to show and tell. Create opportunities for students to share their work in and beyond the classroom. Create Scratch studios or use your school website to publish students' projects. Dedicate classroom time for students to show their individual or group projects to the whole class. Encourage classmates to ask the programmer questions about how the project was coded.

#13 Use a variety of formative and summative assessment. At the end of the day, you also want to know what students have learned, and especially, what aspects they may be struggling with. Looking at final products alone may not tell the whole story. Assessments can include quizzes with freeform or multiple-choice question types that may involve code

analysis and/or debugging. You can find some examples from my classroom at *edfinity.com/join/9EQE9DT8* (you must register).

So there you have it! The use of engaging introductory programming environments along with these pedagogical tips will go a long way in promoting a deeper understanding of algorithmic thinking and programming. These strategies blend and balance the best of interest-driven informal learning with what we know about conceptual learning, pedagogy, and assessments in formal settings.

CT in Action

Computational Thinking is a Model-Eliciting Activity

Fred G. Martin

A Model-Eliciting Activity (MEA) is a way of engaging students in mathematical thinking developed by educator and researcher Richard ("Dick") Lesh. MEAs were created to help students connect mathematical thinking to the everyday world (Lesh and Doerr, 2003).

In an MEA, students are given a problem situation, and their task is to formalize it by designing a strategy for generating solutions. Rather than teaching specific math content, MEAs bring forth students' existing mathematical knowledge. Students work in teams on MEAs and share their results with each other to gather feedback. Dozens of MEAs have been developed. Here are a few:

- Bigfoot—Measuring footprints, students develop a method to find the height of the person. Students measure their own footprints and heights, and then are given a "news article" about a Bigfoot sighting and an image of the footprint that was found.
- Paper airplanes—After flying some paper airplanes, students are given data on a set of flights done by other students (total distance flown, distance to a target, and time in flight). Students write a memo for selecting best floater, most accurate, best boomerang, and best overall plane.
- Championship volleyball team—Given a set of statistics on players (e.g., jumping ability, serving ability, leadership skills), students design an approach towards selecting the best team.

In these scenarios, students build a model of their understanding of the domain and the problem as they have constructed it. The activities "elicit" this model-building work (Lesh and Lehrer, 2003).

MEAs invite students to bring any of their existing mathematical knowledge to bear on the problem-situation. They are open-ended design problems for which many approaches are reasonable and valid.

From this perspective, model-eliciting activities are like computational thinking (CT). As we come to understand CT, it's clear that it operates at a cognitive level that's above the act of computer programming. The building-blocks of programming ground this thinking, but the essence of CT lies in how these elements can be of service in a design process.

Furthermore, it's evident that to understand CT "problem solving" is too narrow a frame. CT is most on display in the act of "problem-framing"—the more profound work of sense-making such that a situation can be recognized as a problem (Fischer 1994). In other words, per Charles Kettering, American inventor and former head of research at General Motors, "A problem well stated is a problem half solved."

Both MEAs and CT share this problem-framing activity at the high level, and the formalisms of mathematics or computing at the low level. But they share other qualities too:

- **Creating instructions or a recipe.** With MEAs, the end product of the process is a memorandum that students prepare for other teams or an external client. With CT, it's the written program code. Both approaches share a "meta" quality—the objective is not a solution, but rather a process for finding specific solutions, given some input data.
- **A way of evaluating the quality of the solution.** With MEAs, other students can carry out the recipe. With CT, the computer runs the code. In both cases, this feedback should support students' learning.
- **There is not one right answer.** As with true design problems, many approaches are viable and welcomed. From a pedagogical standpoint, this is empowering, because students know they are not merely solving a puzzle, but rather, solving a more authentic real-world problem. Their individual perspectives can contribute to each other's learning.
- **The students' works are external artifacts that embody their understandings.** One of the pleasures of working with students who are solving problems computationally is that their code serves as a socially-shared artifact that represents their thinking and can be discussed. MEAs share this quality.

It's worth noting that code is a much less forgiving medium than prose written for each other. Students working on MEAs don't have to worry about syntax and debugging. For our students who are frustrated, we might consider having them write "code" for ourselves rather than the machine.

There is a rich body of literature on MEAs that we can learn from. Recognizing that CT involves defining "the problem" can help us design better learning environments.

REFERENCES:

Fischer, G. (1994). Domain-oriented design environments. Automated Software Engineering, 1(2), 177-203.

Lesh, R. A., & Doerr, H. M. (2003). Beyond constructivism: Models and modeling perspectives on mathematics problem solving, learning, and teaching. Routledge.

Lesh, R., & Lehrer, R. (2003). Models and modeling perspectives on the development of students and teachers. Mathematical thinking and learning, 5(2–3), 109–129.

Championship volleyball team MEA: *www.cpalms.org/Public/PreviewResourceLesson/Preview/74385*

Case Studies for Kids! (list of MEAs): *engineering.purdue.edu/ENE/Research/SGMM/CASESTUDIESKIDSWEB/case_studies_table.htm*

csta Voice
Computer Science Teachers Association
2 Penn Plaza, Suite 701
New York, NY 10121-0701
We're on the Web: csta.acm.org
Like our Facebook page!

CT Resources

Resources Worth Examining

Joe Kmoch

Take a look at these sites in your quest for more knowledge about computational thinking (CT).
CSTA (*csta.acm.org/Curriculum/sub/CompThinking.html*) offers a wide variety of CT resources, including:

- *CT* flyer: An operational definition for CT, describing its characteristics and the dispositions and essential attitudes
- *CT Leadership Toolkit:* Resources for Creating Systemic Change and Implementing Strategies Guide
- Research Notebook*: CT–What and Why?*: An article by Jeannette Wing presented at the CS Workshop at Carnegie Mellon University
- *CSTA/ISTE CT Workshop:* Includes materials created as part of a CSTA/ISTE project and supported by the National Science Foundation

LeadCS (*leadcs.org*) is a computer science (CS) advocacy site focused on being a resource for K–12 district, school, and teacher leaders who wish to develop CS education in their schools. Interesting articles are found within several Tools.
Communication Tool

- CT, CT practices, and computational literacy are among the terms defined
- Resources for promoting CS

Courses Tool

- Connecting and integrating CS with other disciplines. CT is explored through the infusion of CS content into various courses and domains.

Instructional Resources Tool

- Inventory of CS instructional resources

Exploring CT (*g.co/exploringct*) was developed by Google to provide a better understanding of CT for educators and administrators, and to support those who want to integrate CT into their own classroom content, teaching practice, and learning. CT is defined in terms of a half dozen skills, five dispositions or attitudes, and 11 concepts. This site includes over 130 lesson plans, demonstrations, and programs, all aligned to the *CSTA K–12 Computer Science Standards* (U.S.), as well as standards from the UK, Australia, New Zealand and Israel.

- The repository of curated lessons spans 17 subject areas, including mathematics, the sciences, language, and U.S. history from ages 8 to 18 (roughly U.S. grades 3 to 12).
- There are six videos (4 to 5 minutes) exploring a variety of intriguing problem-solving situations. These videos are focused around the Seven Big Ideas in the *Advanced Placement CS Principles* (APCSP) course.
- Additionally, there is a Resources page with dozens of curated resources for educators and administrators.

Computing at School (CAS) (*www.computingatschool.org.uk*)

- *CT – A guide for teachers*. This resource aims to help develop a shared understanding of the teaching of CT in schools. It presents a conceptual framework of CT, describes pedagogic approaches for teaching, and offers guides for assessment (*community.computingatschool.org.uk/files/6695/original.pdf*).

MARK YOUR CALENDAR

SIGCSE
March 2-5, 2016, Memphis, Tennessee
sigcse2016.sigcse.org
Meet CSTA staff and Board members at the CSTA booth!

ACSL Contest #3
March 11, 2016
www.acsl.org

CUE 2016
March 17-19, 2016, Palm Springs, California
www.cue.org/national

ACSL Contest #4
April 15, 2016
www.acsl.org

ACSL All Star Contest
May 28, 2016
www.acsl.org

CAS National Conference
June 17-18, 2016, University of Birmingham, UK
community.computingatschool.org.uk/events

2016 CSTA Conference
July 10–12, 2016, San Diego, California
cstaconference.org

CSPdWeek
July 18-22, 2016, Colorado School of Mines
www.cspdweek.org
CSTA is a founding partner of CSPdWeek

Check the most recent CSTA events on the CSTA website
www.csteachers.org/ProfDev
List your CSTA event by contacting *t.nash@csta-hq.org*

For the latest and greatest tips & tricks and news & views, check out
The Advocate Blog
blog.csta.acm.org

Meet the Candidates

This is your opportunity to meet the ten candidates running for the five open 2016–2018 CSTA Board positions. In addition to working with the Executive Director in setting the organization's direction, Board members carry out many of CSTA's operational tasks through committees and task forces.

As in past years, the election will take place online using the ElectionBuddy voting system. On February 22, all current CSTA members should have received an email from ElectionBuddy with a personalized link to the ballot. If you have not received your email, first check your spam filter, then contact *customerservice@csteachers.org*.

Voting ends March 22

9–12 REPRESENTATIVE CANDIDATES

Stacey Kizer

Personal statement

I am a strong candidate because I am innovative, dedicated to learning, and am passionate about continually improving the practice of teaching computer science (CS). Throughout my 7 years of teaching high school CS I have been dedicated to integrating new perspectives, developing rigorous (and fun!) learning strategies, and to finding opportunities to connect with industry in my classroom. I would bring enthusiasm, creativity, and interpersonal skills as well as analytical and statistical experience as a researcher to the position that I believe would be an asset. On the local level, I already advocate for CS programs, for improving professional development, for creating opportunity to build community among CS teachers, and for developing innovative curriculum, and would welcome the opportunity to have a similar impact on the national level.

What experiences and/or interests in K–12 CS/ information technology (IT) education qualify you to serve as a leader for the organization?

I have 7 years of experience as a high school CS/IT teacher. I am enthusiastic about promoting CS within the schools where I teach and endeavor to make the programs interesting, relevant, and visible. I'm also passionate about ensuring that underrepresented groups are welcome in my courses and finding ways to motivate these students to consider a future IT career. Further, I am active and involved with the technology community in the Nashville area so that I can be sure that my Programming & Logic curriculum matches what those organizations are looking for in future employees.

What previous experience do you have with CSTA?

I am a member of CSTA and also served as a reviewer for the 2016 conference proposals. I will also be presenting two sessions at the conference. I am eager to become more involved with CSTA and hope to do that as the 9–12 representative.

What leadership skills do you have that would enrich the Board and the organization?

I possess strong relationship-building skills that I believe would be an asset to the Board and to CSTA. I am self-confident and have good communication skills— both of which contribute to my ability to be decisive, ask good questions, and build teams. Further, I'm skilled at being responsive and not reactive, often drawing upon strong analytical and reasoning skills to diffuse a stressful situation. I'm also positive and outgoing, which makes me approachable. Additionally, I have a talent for being able to take large projects and break them down into measurable steps with specific goals and outcomes.

What do you think are the most important issues for K–12 CS education?

Three important issues in K–12 CS education are: (1) preparing CS teachers, (2) teaching a rigorous set of in-demand skills, and (3) attracting underrepresented groups to CS courses. Understanding how CS teachers learn their content and also how they learn to teach it is an interest of mine, as well as the topic of my current dissertation research. Additionally, I think it's important that students learn programming using industry-standard IDEs and also gain employability skills important in IT careers. Lastly, I consider motivating underrepresented students to enroll in CS courses and developing strategies to keep their interest to be vital.

Chinma Uche

Personal statement

The fast changing CS landscape and influx of new members to CSTA demand structures that support teachers locally, nationally, and internationally. As someone who has worked closely with teachers for more than ten years, I want to be a voice for the 9–12 CS teacher in this landscape. I will lend my 14-years' experience as a 9–12 CS teacher, my six-years' experience as a CSTA chapter president supporting Connecticut teachers, my experience as a K–5 Code.org affiliate who trains K–5 teachers, and my *Mobile CSP* PI experience as one who introduces CS into schools to support the work of CSTA; while being a committed voice for the 9–12 teacher. Furthermore, I want to work towards the provision of appropriate professional development opportunities for teachers so they can support their students.

What experiences and/or interests in K–12 CS/information technology (IT) education qualify you to serve as a leader for the organization?

I have been a CS teacher since 2002, teaching AP CS in C++ and Java. I have hands-on experience with most of the tools for teaching CS in K–12 including Scratch, Snap, App Inventor, Alice, Pencil code, and App Lab. As an official CSP pilot instructor for the last four years, I have led several teacher workshops, including presentations at CSTA conferences. Furthermore, as president of the Connecticut chapter of CSTA, I have led advocacy efforts for more access to CS for all K–12 students and for adequate PD provision for teachers.

What previous experience do you have with CSTA?

I was privileged to be part of the 2008 cohort of CS Advocacy Leadership Team (CSALT) representing Connecticut. The training I received led to Connecticut joining CSTA as a chapter in 2009. I have since then, as president of CTCSTA, worked to promote the vision of CSTA within my state and

supported its work through attendance at conferences and leading of workshops, while advocating for K–12 CS. I currently serve as an Ex-Officio member of the CSTA Board and a member of the CSTA 2015 –2016 Standards Revision Task Force.

What leadership skills do you have that would enrich the Board and the organization?
As Connecticut CSTA chapter president, I have worked with many stakeholders in the CS community. I also have experience as a Connecticut Education Association (CEA) building representative, working closely with union leadership in promoting respect for teachers and valuing their time. I will bring to the Board a diverse perspective as a high school classroom teacher, and PD facilitator for 9–12 teachers and K–5 teachers. In addition, I have been privileged to work closely with our higher education colleagues and the general CS education community who provide the resources that are changing our landscape.

What do you think are the most important issues for K–12 CS education?
The rapid growth and availability of free online resources mean that students come into CS classes with varying levels of need/experience. Professional development needs to prepare teachers to know where their students are and how to support students at their different levels, to maintain engagement. Additionally, recruiting, maintaining, and supporting CS teachers who are committed to supporting all students are important K–12 CS education issues. Hand in hand with this is maintaining communications with school districts, providing research-based resources, all the while encouraging districts to bring CS to all of their students.

AT-LARGE REPRESENTATIVE CANDIDATES

Myra Deister

Personal statement
I was drawn to CS over 25 years ago and accepted employment in another school district for the opportunity to teach CS. My passion for CS has provided the opportunity to take on an advisory role to the feeder district about building a pipeline for CS beginning with elementary students, to junior high students, and on to my district high school students. Additionally, I have worked as a member of ACCESS to update CS teacher credentialing in California, added CS Principles to my district course catalogue, and actively participated in my local CSTA chapter. I desire to serve on the board of directors because I am excited about the new direction that CSTA is embarking on. I want to continue to assist with not only the governance changes but in furthering the opportunities to support current and new computing teachers with recruiting, pedagogy, and curriculum.

What experiences and/or interests in K–12 CS/information technology (IT) education qualify you to serve as a leader for the organization?
As an active member of my local CSTA chapter, I have presented at our meetings both the information I gathered at conference attendance and my experience with new pedagogy and curriculum that I have used to increase the interest and understanding of the students on my campus in CS. I also serve on the CS curriculum committee for my district where I share the importance of CS with district administrators and have taken on the leadership role in negotiating articulation agreements with the local state college for the district CS courses.

What previous experience do you have with CSTA?
My experience with CSTA includes serving as the K–12 Board Representative for 2 years and At-Large Representative for 6 years as well as a local chapter officer. As a CSTA board member I have served on the membership, communications, and mini-grant committees, served on K–8 Task Force, submitted blog posts, participated in and moderated the CSTA Listserv, participated in the K–8 Twitter chats, consulted with state assembly people, assisted other teachers with starting CS classes at their schools, and have served as Governing Committee Chair. I continue to promote membership in CSTA to all teachers.

What leadership skills do you have that would enrich the Board and the organization?
I have honed my leadership skills through my participation as a local CSTA chapter officer, accreditation team member, local Computer Using Educators (CUE) board member, school district budget committee member, NEA CTE Caucus Cabinet member, and campus representative and negotiations team member for the teachers' union. Through my service in these organizations, I have learned to present my point of view succinctly, listen to others, and suggest time-efficient solutions to problems. I have, in every position, carried through with assigned tasks promptly and researched and presented innovative solutions.

What do you think are the most important issues for K–12 CS education?
The important issues facing CS education are access to CS courses, professional development, and diversity. The announcement of the president's "Computer Science for All" initiative has created an opportunity for CSTA to increase its effort in these areas. This announcement has opened the door to more advocacy efforts with state legislators for expanded CS offerings through CSTA support to the local chapters. With expanded offerings comes the need for professional development. CSTA should be at the forefront of this effort as a result of the quality conferences and professional development opportunities that CSTA is known for.

Michelle Lagos

Personal statement
I have been a CS teacher for over 15 years, working in an underdeveloped country in Latin America. I've had the privilege to serve on the CSTA Board of Directors in the past and recently on several task forces. The International community of CSTA and its members require special attention since we share the same struggles as U.S. teachers regarding budget restrictions, lack of resources, a diminishing importance given to the subject matter by the government and its administrators, as well as a lack of professional development opportunities. CSTA has an enormous opportunity to provide resources and support to these teachers in many countries, making CS education a truly global affair. My international experience and ongoing relationship with CSTA motivates and qualifies me to help

others in a similar situation.

What experiences and/or interests in K–12 CS/information technology (IT) education qualify you to serve as a leader for the organization?
I have been a K–12 CS teacher for over 15 years. I have taught all levels starting in Kinder all the way to 12th grade. I am currently the CS department head for my school, member of the School Improvement Committee that analyzes and recommends how to improve the school in all areas. I also work with the Core Leadership Team that is in charge of reviewing and improving the school's academic curriculum and work as technology integrationist. With this I have enabled my school from making the transition of teaching technology education to offer CS courses throughout all divisions, adopting the CSTA standards.

What previous experience do you have with CSTA?
I had the privilege of serving as the International representative for the term of June 2012–July 2014. I also served in the Curriculum committee, K–8 committee, and chaired the International committee. One of the projects that I promoted during my time on the board is the translation of the CSTA K–12 Standards to Spanish. I am currently working as a member of the Computational Thinking task force. I have also volunteered to work during the CSTA conference last summer.

What leadership skills do you have that would enrich the Board and the organization?
I already work in a leadership position at my school and know the amount of time and work it implies. I believe that the best leader is one that is not the head of the team all the time but is a team worker and knows how to balance delegating work and the amount of self responsibility. The fact that I already served in this position before allows me to know the needs our International member community has and the kind of projects and events that might will create a greater impact on this part of the CSTA community.

What do you think are the most important issues for K–12 CS education?
1. Explaining the difference between technology education and CS education as clearly as possible. I deal with many people involved in education who still believe they are the same thing and the courses they offer as CS are really technology education.
2. Believing students must have a certain level of previous knowledge to start learning CS. This generates a debate as to which grade is the best one to begin a CS course.
3. Advocating for "non-techie" students, helping them understand that CS is used in many areas of their real life, as well as debunking the CS student stereotype.

INTERNATIONAL REPRESENTATIVE CANDIDATES

Miles Berry

Personal statement
I am convinced that a high quality computing education should be part of every young person's entitlement to a broad, balanced curriculum, equipping them for the opportunities, experiences, and responsibilities of later life. Computational thinking provides unique insights into problems and systems, and programming offers a medium through which creativity can be expressed; these things ought to be part of all students' education. I have eighteen years of experience in teaching and school leadership, and a further six in teacher education. I have played a central role in the design, development, and implementation of England's 5–16 computing curriculum, with experience in developing resources and continuing professional development programs. Recent travel has provided a broader International perspective. I maintain an active presence across social media, see *milesberry.net* and *@mberry*.

What experiences and/or interests in K–12 CS/information technology (IT) education qualify you to serve as a leader for the organization?
- Served as IT coordinator in three schools
- Currently Principal Lecturer (equivalent to Associate Professor) for Computing Education, working on undergraduate and postgraduate initial teacher education programs
- Author of *Switched on Computing* (adopted by some 5,500 UK schools, BETT and ERA award winning)
- Member of drafting panel for English computing curriculum
- Member of Computing At School and Naace management boards, Raspberry Pi Foundation and UK Forum for Computing Education
- Author of computing materials for England's Department for Education, the British Computer Society, BBC, Microsoft, and Code Club
- International keynote speaker and consultancy on curriculum design and CPD.

What previous experience do you have with CSTA?
- Joined ACM including CSTA and SIGCSE in 2009
- Used CSTA K–12 standards when developing school curriculum and university modules
- Avid reader of much content, particularly valuing CACM, Inroads, Voice, and the Digital Library
- Contribute regularly to CSTA Members list and occasionally the SIGCSE Members list
- International reviewer for K–12 CS framework
- Most of my work has had a UK focus, as a board member of Computing At School, the closest equivalent to CSTA here (including as a member of the primary task force and assessment and research working groups). I'm also a Chartered Fellow of the British Computer Society.

What leadership skills do you have that would enrich the Board and the organization?
- MBA in educational management
- Six years as deputy head teacher; three years as head teacher
- Serve on trustee board for educational charity, management board for CAS (UK equivalent to CSTA), and

as a member of the Raspberry Pi Foundation
- Extensive experience of project leadership related to information technology in education and computing education
- International perspective on curriculum development, not limited to UK
- Effective partner engagement (industry, voluntary sector, government, and education institutions)
- Currently leading a small, agile, highly effective computing education team at Roehampton.

What do you think are the most important issues for K–12 CS education?
- Inclusion—ensuring that all students have equitable access to a high quality computing education
- Pedagogy—what makes for the most effective teaching of computing to young people?
- Professional development—particularly equipping teachers to teach effectively a subject of which they may have little prior knowledge
- Advocacy—making a persuasive case for including computing on the school, local and national curricula
- Getting the relationship between coding and computational thinking right—I don't see coding as an end in itself for most students, but it does seem the most effective way to develop computational thinking.

Michael Jones

Personal statement

I am a highly pro-active advocate of CS education. As well as a classroom teacher, I collaborate with a range of bodies in the development of CS pedagogy. I work with existing teachers who are moving into teaching CS. To develop a broader view of the ways in which CS can be taught, in 2015 I undertook research in the U.S. The results of this have been published in a national report. Ideas from the U.S. are being applied—in particular, the engineering approach. In my role as an advocate of CS, I am a Master Teacher with responsibility for delivering a 6 hub meetings/professional development sessions a year. As an SLE I have a remit to spend 10 days a year with schools in support of their curriculum.

What experiences and/or interests in K–12 CS/ information technology (IT) education qualify you to serve as a leader for the organization?

Throughout my career I have volunteered and have been asked to support schools locally and nationally. This has enabled me to build a comprehensive picture of good and poor practice and then to develop curriculum models across age groups. As one of the first teachers of CS under the new mandate in England, my experiences enable me to create a mature program in my school. Outside my standard teacher responsibilities I am able to assist in the shaping of pre- and in-service professional development, thereby helping to secure the next generation of CS/IT teachers.

What previous experience do you have with CSTA?

Member of CSTA since 2012. I have contributed two articles to the CSTA Voice: "Bricklayers to Architects: Transition for CS in England" (Volume 9) and "Computer Science at the Start of a New Era" (Volume 10). In 2015, released resources for the development of Python programming to CSTA members under the title "High School in a Box". I attended the CSTA annual conference in 2014. The conference enabled me to create a relationship with the App Inventor team at MIT. This cross-Atlantic partnership is of great benefit to teachers in England delivering to students in the 14 to 17 age range.

What leadership skills do you have that would enrich the Board and the organization?

As an advocate of CS and IT, I work with a range of NGOs. These include universities and bodies, such as the CAS, the PiXL Club, and NCTL. Within each of these I hold a mandate to provide direction of curricula. Working at a national level, I am a member of the National Executive for the PiXL Club to develop in-service programs. At a regional level, I work with CAS as a Master Teacher. This year has seen me take a regional lead on the use of blocks programming environments and the launch of the BBC micro:bit.

What do you think are the most important issues for K–12 CS education?

Provision of a relevant curriculum reflecting the needs of students as they mature into contributing members of society. This in turn requires that we are clear in what constitutes a relevant curriculum. Preparation for and delivery of this needs immediate attention to the preparedness of teachers, both pre- and in-service educators. As technology evolves, any teacher preparation model must be able to rapidly adapt. This requires highly focused advocacy from NGOs to government organizations is a priority. Attention to these issues will overcome the issue of promoting qualified students who not only use, but contribute to, CS.

STATE DEPARTMENT REPRESENTATIVE CANDIDATES

Anthony Owen

Personal statement

As a result of Governor Hutchinson's vision for CS, I, with the support of many partners, have been able to lead Arkansas's initiative to the forefront of CS education. It is with great enthusiasm that I submit my application to serve the greater CS community as the State Department Representative to the CSTA Board of Directors for the 2016–2018 term. I look forward to this opportunity to further CS education for all teachers and students across the nation using the diverse skills and knowledge that I have gained through years of varied public service.

What experiences and/or interests in K–12 CS/ information technology (IT) education qualify you to serve as a leader for the organization?

I am currently serving as the Arkansas Coordinator of CS. I have worked for the Arkansas Department of Education helping develop mathematics and CS curriculum and policy for over four years. Under my tenure as Coordinator, Arkansas has become nationally recognized as a leading state in CS education. As a result, I am currently serving various national entities through curriculum development support, policy discussions, and implementation suggestions. I also hold an Arkansas Department of Education license in 4–12 CS Education.

What previous experience do you have with CSTA?
I have worked with Ms. Deborah Seehorn, Mr. Carl Frank, and Mr. Daniel Moix on Arkansas's CS standards development and interacted with other individuals with CSTA since taking on my current role.

What leadership skills do you have that would enrich the Board and the organization?
I have an extensive background serving in numerous leadership roles prior to my current position. Since being appointed as Arkansas's Coordinator of CS, I have worked with multiple partners to successfully drive Governor Hutchinson's CS initiative. My successes, demonstrate an ability to work effectively with a team; a strong work ethic; a careful attention to detail; my ability to resourcefully and creatively brave tasks outside my normal job function; my ability to absorb, disaggregate, and make timely, yet informed decisions based on data; an exceptional technological prowess; and my passion to champion laudable organizational missions and values.

What do you think are the most important issues for K–12 CS education?
My belief is the most important issues for K–12 CS education continue to revolve around the unification of efforts. At this moment, multiple states, including Arkansas, are all implementing what they think is best. Through the work of CSTA, states can continue to be provided with guidance on best practices and lessons learned, which is increasingly necessary now that CS is becoming a high profile initiative across the nation.

Doug Paulson

Personal statement
CS is an essential component of a strong STEM education strategy. In Minnesota, we have been focused on identifying opportunities to support opportunities for students from early childhood to post-secondary to engage in meaningful CS learning. As a part of my role as the state STEM specialist, I have been involved with supporting these initiatives. I am interested in being a board member for CSTA as the organization's work have been informative for our work and I would like to give back to the organization as well as continue to advocate for systemic CS opportunities across the country. My vision is to inspire interest in elementary school, make connections across the curriculum in middle school, and challenge and prepare students for career and college opportunities in high school.

What experiences and/or interests in K–12 CS/ information technology (IT) education qualify you to serve as a leader for the organization?
Providing leadership for CS has been critical for my work. I have provided professional development for elementary school teachers in pedagogy for computational thinking and worked with PLTW on the relationship between AP courses and the PLTW courses. To have a better understanding of this work last summer I designed and delivered a coding camp at the Hormel Gifted Symposium. Most recently I have worked to develop a contract with Mouse, Inc. to provide IT certification curriculum for all Minnesota High Schools. I have also attended NSF meetings on CS to inform Minnesota.

What previous experience do you have with CSTA?
I have worked with the board of our state affiliate, Gopher State CSTA.

What leadership skills do you have that would enrich the Board and the organization?
I have a broad STEM perspective to provide to CS. With so many responsibilities on schools and teachers agenda, strategic integration is essential. My background and perspective provides this skill. I have also worked to engage stakeholders from K–12, higher education, and industry to set goals and identify key indicators to measure progress. Finally, I have experiences on multiple boards of directors and held leadership positions to understand how to utilize the board to influence change within the organization and in the larger community.

What do you think are the most important issues for K–12 CS education?
Two of the most important issues for CS is to define the discipline so it is uniformly understood by the larger education community and to identify its role in the system so all students can be engaged in CS. First when many in education think of CS coding is the first thing that comes to mind. While coding is important, CS is more than that and we need to communicate it clearly. Secondly, to be successful in implementing CS for all, more opportunities need to be identified as how it supports and is supported by other disciplines rather than supplanting each other.

UNIVERSITY FACULTY REPRESENTATIVE CANDIDATES

Darcy G. Benoit

Personal statement
My motivation for this position is my desire to ensure that all children have the opportunity to learn computing in the public school system. My experience over the past several years includes leading a team to run several local events for international outreach programs—FIRST Lego League, Robofest, Hour of Code, etc.—with the hopes of exposing more children to computing ideas in the classroom. I recognize that computer programming is a multiplier—it multiplies the advances in all fields where it is applied—and it is important that all students have the opportunity to have their educational experience enhanced. As a member of the executive of the Canadian Association of CS, I promote CS nationally, and believe in the importance of CS for our future.

What experiences and/or interests in K–12 CS/ information technology (IT) education qualify you to serve as a leader for the organization?
I have been interested in CS education for a long time. One of my motivations for choosing a small, primarily undergraduate university was so that I could meaningfully participate in undergraduate teaching. My teaching workload includes not only core CS topics such as first year programming, but also service classes that allow me to interact with students not in the field. These classes give me perspective as to why students select (or avoid) CS as a major, and the gaps that exist in their knowledge of fundamental computing ideas.

What previous experience do you have with CSTA?
I have been involved in CSTA events for the past three years, although I have only been a member for the past year. I have run Hour of Code events locally for the past three years, including two years of public events, and visits to local schools for this past year. Previous to this, I had organized a coding event focused specifically for teenaged girls, and was involved in several coding camps at my local university. I currently have an application pending to create a province-wide CSTA branch in Nova Scotia, Canada.

What leadership skills do you have that would enrich the Board and the organization?
I have extensive experience with university committees both at the faculty and university level, and I am involved in several province-wide and national organizations at the executive level. I am passionate about CS education, and have published several papers on teaching at the university level. I have worked with the provincial Department of Education and Early Childhood Development to revise CS curriculum at the high school level, and continue to champion CS education at all levels in the school system.

What do you think are the most important issues for K–12 CS education?
I believe the most significant issue for K–12 CS education is educating current and new teachers of the importance of CS for all students. This includes the students who will become biologists, English majors, and business students, as computers are going to daily touch their lives and work. We know that passionate teachers at the high school level can change how a student thinks about a subject. Until we have passionate CS teachers—and supportive, understanding teachers from other disciplines—we will have a shortage of CS students.

Fred G. Martin

Personal statement
Now is the most critical juncture for the CSTA since it was founded in 2006. President Obama has just announced $4B of funding to bring CS Education into the school day. Now more than ever, CSTA's singular focus on the K–12 teaching professional is needed. I have been working in computing education for 25 years. I co-created the Lego Mindstorms Robotics Invention System and launched robot contests as way to learn about computing. As a university faculty researcher and teacher, I am passionate about understanding how students learn the key ideas in our field. Recently I am working with MIT on the App Inventor project, collaborating with two urban districts on a middle school computing curriculum where students make apps for social good. To me, the most important thing is teacher professional development. This is the mission of the CSTA, and I believe its success and your success are paramount.

What experiences and/or interests in K–12 CS/ information technology (IT) education qualify you to serve as a leader for the organization?
I was a member of Seymour Papert's research group beginning in 1986. Some of my earliest experiences were in teacher workshops, helping teachers learn the Logo language, driving the turtle around the screen on Apple IIs and IBM PCjrs. I internalized Seymour's central idea that when writing a program, you are making a concrete object that reflects your understandings. Since then, I've organized teacher workshops to learn robotics, science using data visualization, and mobile app design. Most recently I served as a panelist for the state of Massachusetts, helping develop Digital Literacy and Computer Science standards for K–12.

What previous experience do you have with CSTA?
My academic department joined the CSTA as an institutional member in 2009. I have served as university representative of the Greater Boston CSTA chapter since its formation in October 2010. I worked with the chapter's co-presidents to produce a workshop co-sponsored by CSTA and Google in summer 2012. Over 50 teachers participated in the two-day program. I was elected as University Representative to CSTA in 2014. During my term, I have contributed to decision-making at the board level, served as a member of the curriculum review committee, and served as co-chair of the computations thinking task force and funding committee.

What leadership skills do you have that would enrich the Board and the organization?
As a university faculty member, I engage in many leadership roles. I lead research grants with faculty and staff colleagues, working with teams of students. I serve on and co-chair university committees—sometimes producing quick, results-oriented products, and other times addressing complex, long-term challenges. I consider teaching classes as a leadership role too. I thrive on working together with other people toward common goals. I enjoy learning from others' insights, and I always want the teams that I am part of to reach a consensus on how to carry out or work.

What do you think are the most important issues for K–12 CS education?
The most important issue for K–12 CS education is teacher professional development and the recognition of the crucial role of the classroom teacher. Recent efforts have focused on creation of learning standards and curricula. These are essential, but the classroom teacher brings these materials to life for our students. It is the teacher who inspires our students and encourages those who are less confident. As I have come to more deeply understand the mission of the CSTA, I recognize how important it is for teachers to be represented and heard.

Institutional Member Alert

Post your K–12 CS professional development and student events on the CSTA website.

Don't delay! Send the details today.

customerservice@csteachers.org

www.csta.acm.org/ProfessionalDevelopment/sub/TeacherWorkshops.html
www.csta.acm.org/Resources/sub/CSEventsforKids.html