As the summer days get longer and hotter, it can only mean one thing—the CSTA Annual Conference is approaching.

This year’s conference will be held July 8–11 at the Renaissance Baltimore Harborplace Hotel in historic Baltimore, Maryland (a.k.a. Charm City). It looks to be our biggest and possibly our best, with 20 workshops, 38 presentation sessions, five birds-of-a-feather sessions, two keynote talks, and a cybersecurity panel. More than 40 sponsors and exhibitors will be there to display the latest technology and educational resources for teachers. And we will recognize many outstanding teachers, administrators, CS advocates, and students through CSTA awards. As always, the highlight of the conference (at least for me) is the opportunity to meet with fellow educators and share ideas and experiences.

The theme for the conference is “Identifying Teacher Superpowers.” As educators, we all have superpowers that impact our students. Perhaps it’s x-ray vision, being able to spot a bug in a program or the fault in a student’s reasoning. Perhaps it’s super hearing, being able to listen to a question and know just what piece of understanding is missing. Perhaps it’s telepathy, being able to enter into the mind of a young student and know how to explain a concept in an understandable way. Perhaps it’s invisibility, being able to disappear when appropriate, to allow students to drive the learning. We may not be able to fly, but we can all be superheroes in the lives of our students.

On a personal note, this will be my last conference as a member of the CSTA Board of Directors. My term as Board Chair ends just before the conference, but I will continue as Past Chair for a year to support the new Chair, Fred Martin. It has been my pleasure to serve on the board for the past eight years, along with amazing colleagues, Executive Directors, staff members, and volunteers. Despite our growth, CSTA and the annual conference continue to be driven by educators, and the “small community” feel is always there. I look forward to seeing everyone in Baltimore and continuing to work with you over the next year.

Dave Reed
CSTA Board Chair
CSTA AND DEGREED
EMPOWER TEACHERS WITH A PD PIPELINE

Sarah Danzl

Many of us have experienced firsthand the digital disruption currently taking place, from the way we shop to the way we consume news. Many things are now taking place on screens instead of in person. But this technology must be created and developed by someone, hence the increasing demand for qualified computer science (CS) professionals and teachers. You might be surprised to learn that “fewer than 10% of schools offer a CS program,” shared Mark Nelson, Executive Director for CSTA.

So how are these individuals finding the resources they need to be effective CS teachers? Many come to CSTA. To deepen the learning available, CSTA recently partnered with Degreed, the world’s first career-long learning platform, to create a professional development (PD) platform for K–12 CS teachers. This pipeline is known as the Continuing Professional Development Pipeline (CPD Pipeline) and is supported by a grant from Infosys Foundation USA.

The first of its kind, the CPD Pipeline is designed to grow the pool of teachers who are both competent and confident in teaching CS concepts and practices. With many CS teachers from non-CS backgrounds, CSTA recognized the need to continually train, retrain, and boost capability. After looking at many platforms, the CSTA team chose Degreed for the user-friendly environment and structured pathways that will help teachers find content they need, discover new learning, and demonstrate mastery.

“In capability and philosophy, Degreed and CSTA are a match,” commented Dr. Nelson. “We realized our challenge wasn’t in providing training—we actually had a workforce development problem. We needed to make sure our workforce was continually learning, growing the skills that matched the current needs.”

The partnership between CSTA and Degreed delivers on four core learning components:

1. Self-assessments of skills, interests, and experience
2. Personalized PD roadmaps to guide the process
3. Digital badging, with support from Badger
4. Connections to the CS community to interact with other teachers; and digital portfolios to highlight PD and manage career paths.

While CSTA is excited about the future possibilities of rethinking K–12 certification, what inspires Dr. Nelson is how this partnership will empower teachers. “Having control over what you do in your role is big for satisfaction in any job. Teachers have the least control of any profession. We can now empower them to learn any time, whatever they want. This produces good teachers that produce good students that will one day be in our workforce.”

Start discovering new learning and track your accomplishments with the CSTA PD Pipeline on Degreed today!
Editor’s note: David is the newly-elected At-Large Representative on the CSTA Board of Directors.

Computer Science for All Students in New Hampshire

David Benedetto

CS for New Hampshire (CS4NH) began just last year with an opportunity. I had just started working at the New Hampshire Department of Education as Director of STEM Education. We were discussing the idea of having an event in the fall, “something with coding.” People were talking about coding at the State level—it was certainly a big part of the reason I was hired—but there wasn’t much talk about K–12 computer science (CS).

Working with limited resources, and not having prior event-planning experience, I phoned a friend, Mihaela Sabin at the University of New Hampshire (UNH). I met Mihaela a year earlier at the Creative Computing Challenge (CCC), an NSF-funded teacher professional learning program. The CCC program was my introduction to the ideal of CS for All. There hadn’t been much diversity in my college CS program, but I hadn’t really thought much about why that was the case, or how it was a problem. She gave me a copy of Stuck in the Shallow End, by Jane Margolis. The book and the CCC activities and discussions opened my eyes.

Plans for our fall event really began when Mihaela secured funds from the Google CS4HS program. UNH Manchester offered to host the event and help with organization. I was free to focus on connecting with stakeholders and spreading the message. The ECEP Alliance provided support for two of the speakers, Pat Yongpradit from Code.org, and Mark Guzdial from Georgia Tech. Over 100 educators participated in the event.

The benefits from the event were enormous. The planning, event, and the aftermath have all been significant in getting CS4NH off the ground.

Advocacy

We have formed a CS4NH Steering Committee with representation from the NH Department of Education, the NH Charitable Foundation, the NH High Tech Council, the University and Community College Systems of NH, the NH House of Representatives, and others. This group is sharing the message around the state, and it is resonating with decision makers. This is critical in NH, with its strong tradition of local control.

Policy

We are defining CS as an academic content area with teacher and student standards. We are reviewing program requirements in several areas, and the courses that schools are using to fulfill these requirements. We also work with lawmakers on relevant legislation, such as our engineering/technology grant program and robotics fund.

Implementation

We are working with institutions of higher education and PD providers to establish CS programs for educators and incorporate more computing into other STEM educator preparation and PD programs. These are taking several forms for preservice and in-service — regular credential, add-on endorsement, graduate certificate, or other option. Most critically, we are working with K–12 personnel at every level, to raise awareness, identify broadening participation in computing as a critical objective, and establish local plans to expand CS education in ways that works for them.

I think the most important thing about this work is that it’s collaborative and responsive. This effort will succeed on the backs of a lot of hard-working people—many of whom have been working on this for years before the CS4NH brand existed. We are simply in the right place at the right time, working with a lot of great people.

LEARN MORE

NH Department of Education
CS4NH
Yes They Can!  
CS for Kids Ages 5–7  
Vicky Sedgwick

Editor’s note: Vicky is the newly-elected K–8 Representative on the CSTA Board of Directors.

As we strive for Computer Science for All, it is important not to forget our youngest learners. I often hear statements such as “they have never even used a computer” and “they can’t even read, so how can they write programs?” and “there is no time to add anything more into the school day.” Yes, these are all valid concerns. Still, there are many reasons to start teaching computer science (CS) in the primary grades. Here are a few ideas on how you can integrate CS concepts into the elementary school day.

Computing systems
From the moment students first sit at a computer or use a tablet, they should begin to learn about the device. We need to teach them to use correct terminology when referring to both the hardware and the software they are using. Additionally, we need to help them understand that computing devices are not magic; they are tools to accomplish tasks.

Students also need to learn how to turn the device on and off, how to adjust the volume, how to start and close applications, and how to describe the problems they encounter. Imagine having to adjust the volume for all the students in the class every time they use the devices or having to help every student locate and open the programs they need. We won’t have to do these things if our students learn these basics. It will save time, not take time.

Data and analysis
Not long ago, the kindergarten students at my school asked me to select my favorite fruit from five options. They were collecting data! After they had polled the entire school, they graphed the results on the board, drew conclusions, and made predictions based on the data they had collected. Data collection and graphing is already part of the math and science curriculum for students in the primary grades. We can leverage their learning by using a spreadsheet program to create graphs that show how computers can be used to process data.

Algorithms and programming
Much of the school day consists of routines that follow specific steps. This sounds like an algorithm to me! Don’t say “What are our rules for coming into the classroom?” Instead, ask, “Who can describe the algorithm for coming into our classroom in the morning?”

If you watch students in a coding app, you can almost see the learning happening and the connections being made. It was amazing to watch as my first graders figured out the sequences and patterns needed to create loops in Kodable. It wasn’t easy.

continued on next page
One student tried a level twenty-four times before he finally got it right. Persistence and analytical problem solving are important skills at every age. Programming can give students ways to express themselves creatively and to show what they know in other curricular areas. When students write programs to demonstrate knowledge, not only do they need to know the content, but they need to organize their thoughts in logical ways to write the programs. Caterpillars seemed to be everywhere on our campus this year. Many classes taught or reviewed the butterfly life cycle as we waited for our butterflies to emerge. ScratchJr gave my kindergarten students a way to show their understanding of the butterfly life cycle by writing programs. **Impacts of computing** In many schools, students have login credentials for the school network, websites, and apps. This provides the opportunity to teach our youngest students about protecting their information on devices and online every time they login. We should use this opportunity to teach them why we have passwords and the importance of safeguarding them. CS for All must include our youngest students. It can and should be integrated into the school day. **LEARN MORE** **Primary Programmers** **Beyond the Hour of Code** Primary Coding Adventures (iTunes) Primary Coders:Teaching with Unplugged Activities by Leticia Batista (iTunes) Primary Coders: Introducing Coding in Kindergarten by Dustin Carlson (iTunes) **CSTA Thanks....** Outgoing members of the CSTA Board of Directors for their years of dedication and service. Stephanie Hoeppner Tammy Pirmann Alfred Thompson Aman Yadav Sheena Vaidyanathan **MEET THE AUTHORs** **David Benedetto** State of New Hampshire David is the Director of STEM and CS Education for NH. He works to broaden participation in STEM and CS through policy work and partnerships with schools, universities, the business community, and others. **Doug Bergman** Porter-Gaud School, SC Doug has been in education more than 20 years and has used project-based learning throughout his career. The Porter-Gaud School is a private school in Charleston. **Sarah Danzl** Degreed Sarah leads the enterprise communications and content development efforts on Degreed’s product marketing team. Prior to Degreed, Sarah served as the senior customer marketing manager at Xyleme. **Chuck Gardner** NICERC Chuck is the Director of Curricula. He oversees the creation of content and workforce development initiatives for the National Integrated Cyber Education Research Center (NICERC). **Steve Goodgame** KISS Institute for Practical Robotics Steve is the executive director of the KISS Institute, an independent nonprofit organization that uses robotics to engage students in CS, technology, engineering, science, and math. **Maya Israel** University of Illinois-Urbana Champaign Maya is an associate professor in the Department of Special Education at UIUC. She is the research director of the Creative Technology Research Lab, which studies strategies for increasing access and engagement of all learners, including those with disabilities, in K–12 CS education. **Cindy James** Technology educator Cindy has taught CS and IT at the K–8 level for eight years. She is the Technology Integration Specialist for the Norwood District 63 in Peoria, IL. Continued on page 12
Our work in computer science (CS) classrooms focuses on studying strategies that support teachers who have students with learning and cognitive disabilities in their classrooms. This work reinforces that “CS for All” can, and should, include ALL learners, and shows that when CS activities are planned using Universal Design for Learning (UDL) all students can learn and find joy in CS education.

UDL is an instructional framework that addresses learning barriers through flexible planning that includes multiple ways of (1) engaging learners, (2) representing instructional content, and (3) allowing learners to express their understanding.

Students with disabilities are obviously diverse, with different strengths, challenges, and support needs. The table below includes strategies that our work has shown effective for a wide range of learners.

### General Strategies

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Explanation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Talk with students’ special education teachers about supports and accommodations</td>
<td>The IEP has critical information. Unfortunately, it may not include goals/accommodations specific to CS. Special educators can help translate these supports to CS education.</td>
<td>If students use text-to-speech software for reading, see whether that software is compatible with the tools and materials used in CS.</td>
</tr>
<tr>
<td>Work with your paraeducators</td>
<td>Paraeducators can provide significant support to help students participate in CS activities. They have knowledge about student needs and supports, but they may hang back because of limited CS knowledge. Explain and model for them how you want them to assist in the classroom.</td>
<td>When possible, give lesson plans ahead of time to the paraeducators so they know the big ideas of the lesson(s). Also, encourage them to provide verbal prompting during instruction. This “hands off” approach creates a physical space for students and will prompt students to implement CS problem-solving strategies.</td>
</tr>
</tbody>
</table>

### UDL-based Strategies

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Explanation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create flexible activities that allow students to engage in multiple ways</td>
<td>Students can engage in CS activities differently and still have meaningful experiences. Consider creating a tiered activity wherein everyone does a base project with extension activities for students who complete the base project.</td>
<td>In a robotics activity, create a menu of activities at different challenge levels. All students select an activity from the first challenge level, such as navigating a maze with movement commands and a loop. An extension option could be to refactor the pathway through the maze with fewer commands and multiple loops.</td>
</tr>
<tr>
<td>Provide students with project planning strategies</td>
<td>Students often benefit from using planning documents and graphic organizers that help them plan the steps of their computational projects. Teachers can model the use of these documents in project planning and engage in collaborative “think alouds,” prior to students working on individual computational projects.</td>
<td>Planning sheets help students plan out their ideas using pseudocode. For some students, this might include checklists of required components (e.g., use of conditional statements. The teachers can then check these planning documents and discuss steps with students. If students get stuck, they can then go back to these planning documents for support.</td>
</tr>
<tr>
<td>Encourage and teach collaboration</td>
<td>Teach and practice collaboration strategies. It may be helpful to use conversation protocols and sentence starters to help students collaborate productively. Additionally, strategically seat students with disabilities next to peers without disabilities.</td>
<td>Students may use sentence starters when they need help with debugging, such as “I want my sprite/program to...” and “Right now my sprite/program is...” They could also place a “Help” card on their desk if they struggle with seeking out peer support. Lastly, the Collaboration Discussion Framework provides a guide for productive peer collaboration.</td>
</tr>
</tbody>
</table>

continued on next page
## UDL-based Strategies

| Use concrete representations tied to computing | Many students with learning and cognitive disabilities benefit from starting with concrete representations when developing abstract concepts. Tactile and physical activities in the CS classroom provide this concrete introduction. | Teachers can create cards with Scratch or JavaScript commands for students to use to give each other directions to move around the room. They can write a function for a popular dance move. Later, the students can create a similar program with sprites or turtle graphics in a visual programming environment. |
| Include relatable and interesting context | Motivation plays a key role in learning and persevering through challenges. Tying CS activities to familiar concepts and personal interests will help students make meaning of new concepts. | Exploring the shortest path (Dijkstra’s) algorithm: Some students go on road trips while others visit friends using public transit systems. Teachers can use this personal experience to create a CS activity related to finding the shortest path between nodes in a graph. |

## CSTA Thanks....

CSTA Committee Chairs for their outstanding contributions in revitalizing important CSTA committees—the work engines of CSTA.

- Marketing, Communications, and Publications: Hal Speed
- Funding and Business Development: Lien Diaz
- Research: Leigh Ann DeLyser
- Awards: Betsy Hargrove
- Advocacy: Ryan Torbey
- Professional Development: Beverly Vaillancourt and Joseph Pistone
- Annual Conference: Stephanie Hoeppner
- Curriculum and Certification: Deborah Seehorn
- Membership: Carl Frank
- International: Margot Phillipps
- IDEA: Joanna Goode
- Chapters: Fran Trees
- Volunteers: Joe Kmoch
- Bylaws, Policy, and Procedures: Myra Deister
- Nominations and Elections: Fred Martin
- Finance: Dave Reed
CSFORALL INCLUDES ALL TEACHERS

Jennifer Rosato

Editor’s note: Jennifer is the newly-elected Teacher Education Representative on the CSTA Board of Directors.

Computer Science for All is an “…initiative to empower a generation of American students with the computer science (CS) skills they need to thrive in a digital economy.” This original initiative from the White House in 2016, has now morphed into a thriving community, including the CSforAll Consortium and a virtual CSForAll Teachers community. Schools across the country are adding CS to their middle and high school course offerings and integrating CS at the elementary level.

However, for the CSforAll initiative to truly succeed, all schools must offer CS to all students. Early adopters are often larger school districts in metropolitan areas, such as Chicago, Broward County, and San Francisco. This leaves out students in many rural and smaller districts that already have difficulties recruiting teachers and offering elective courses. CS for all students must include all students at all schools.

To improve access for all students, all teachers should have experience and competency in CS. Teachers need to communicate what CS is to students and parents, and be prepared to teach it at a basic level. Much of the professional development for CS teachers targets those already in the classroom. To truly reach all teachers, teacher preparation programs can play a key role by exposing all students to computational thinking (CT) and providing opportunities for deeper exploration of CS.

To that end, Google has awarded grants to the College of St. Scholastica, University of California-Irvine, University of Texas-Austin, and Huston-Tillotson University to support integration of CS and CT in teacher preparation programs, including a focus on the equity issues inherent to CS education.

At the College of St. Scholastica, our project began this spring with the development of a CT module for all teacher candidates. This module will be integrated in an educational technology course (for undergraduates) and in a technology lab (for graduate teaching licensure candidates) over the next year. Successful integration of CT/CS in teacher preparation programs will depend on all faculty, not just students, understanding CT, articulating why it should be taught, and making connections to it in their own courses. For the work to be sustainable as well, ownership of this process must reside with the education faculty, including its leadership, so that it continues beyond the end of the grant.

Our CSS education faculty are now completing the CT module themselves, and this summer they will be learning about equity issues specific to CS. Beginning this fall, faculty will then create and implement modules for their own courses that integrate CT and CS concepts with their subjects. This model ensures students will have multiple opportunities throughout their program to learn about CT and CS. The students will also complete required methods courses as part of their program, one of which will include a field experience component specific to computing. During the methods course, they will learn relevant pedagogy and develop a curriculum unit they can use in their field experience. For those who want to explore CS in more depth, a minor in CS Education is being developed that will include undergraduate CS courses and a specific CS methods course.

In the coming year, we will be developing a MOOC (massive open online course) based on our CT module that can be used by other teacher preparation programs with their faculty and students. Our faculty will also be sharing their integration modules on the project website and sharing their experiences at conferences, including CSTA. We are looking forward to sharing our experiences and helping other teacher education programs support the CSforAll movement.

LEARN MORE
St. Scholastica Certificate in CS Education
A DIFFERENT KIND OF CLASS: PROJECT-BASED CS

Doug Bergman

Editor's note: Doug is the newly-elected 9–12 Representative on the CSTA Board of Directors.

It is exciting to see computer science (CS) at the forefront of educational discussions around the country. Finally, people are starting to understand its value to people and to the industries people care about. But I’d like to suggest something as we move farther down this road. The world of technology is so awesome and dynamic. People absolutely love their new mobile phones, game consoles, tablets, GPS units, and other devices. Give a young person any electronic gadget, and they instantly explore, click, experiment, and use it. In fact, as adults and parents, we are amazed at how mesmerized, obsessed even, they are with technology. It is THAT energy that we can tap to enable our CS classes to be the most engaging and valued classes they take.

But I’ll challenge us to approach these classes differently than most other disciplines. I don’t see this CS-learning environment being conducive to a passive, textbook-led, lecture-style class. The technology and the student can be the center of the learning environment. Projects can be the medium through which they learn. But I’ll take it one step farther—use the projects THROUGHOUT the learning, not just as a tool to demonstrate proficiency AFTER acquiring the skills.

Use the projects to help them learn the skills. That means starting the projects before they have the necessary skills to complete it. As students make progress in their projects, they’ll need new and different tools to accomplish their project goals. That is when a just-in-time mini-lecture, online video, book, blog post, discussion board, peer, or tech forum can help.

Students can use a variety of resources to help them learn the skills and commands they need, in addition to the teacher. Natural collaboration between students happens in this environment. The grade becomes decentralized, even a side product. Learning and making progress in the project are what drive the class, as well as student motivation. It’s a different kind of classroom.

But it requires the teacher to step back and let learning happen. There is an energy that forms around students creating something new using CS tools. The focus of the teacher becomes to guide, point, offer advice, and of course, teach as needed, but only when needed.

Well-constructed project assignments also provide a natural connection to the real world. In a game design class, Skype with professional game designers. In a web design course, take a field trip to a local web design firm. In cybersecurity lessons, invite security experts to speak. The project-based CS classroom is a place where students can use a cool set of tools to help express themselves about things they care about and that matter in the world around them.

LEARN MORE
Doug Bergman’s blog
Pure Genius: Building a Culture of Innovation
Assessing Progress Towards #CSforAll

Bryan Twarek

Editor’s note: Bryan is the newly-elected School District Representative on the CSTA Board of Directors.

The San Francisco Unified School District (SFUSD) and many large school systems across the nation have made recent commitments to expand computer science (CS) education to all students. In this nascent stage of K–12 CS education, simply providing access is often regarded as the end goal. I would like to challenge schools and districts to set bolder goals: to develop a strong foundation in CS among all students and to disrupt the inequitable access and achievement in CS.

When reaching beyond the goal of simply providing access, evaluating progress in this work becomes more challenging. Answering the following questions will help schools and districts assess their progress towards meeting these goals of the CSforAll initiative:

1. Who is taking CS?
   We should use course enrollment data to measure increased enrollment in CS. However, simply offering CS classes does not ensure equitable access, as research has shown that elective opportunities are inequitable (e.g., see Margolis et al., 2008). Thus, it is critically important to conduct subgroup analyses in order to measure the participation of groups traditionally underrepresented, including females, African Americans, and Latinos, as well as other vulnerable populations, including English-language learners and students with special needs.

2. How well are you preparing and supporting teachers?
   Few teachers have background or experience with CS, let alone with teaching CS. Because of this, it is important to ensure that schools and districts are adequately preparing and supporting new CS teachers. We can use teacher survey data to measure the efficacy of professional learning experiences, identify teachers’ needs, and plan for future professional development. PD should not end in the summer; it should continue throughout the school year, and the evaluation should continue, as well. I recommend giving pulse surveys on a monthly or quarterly basis. If holding regular, ongoing PD, this is the perfect time to administer them.

3. What is the quality of instruction?
   We must move from simply tracking the number of classes and students to measuring the quality of instruction in those classes. If schools seek to provide rigorous, relevant, and engaging CS education for all of their students, it is important to measure this through classroom observations. For example, through observations, we can learn how teachers use culturally relevant pedagogy, how students collaborate and communicate, and how traditionally underrepresented students and vulnerable populations engage in the class. We can use rubrics and other observation tools to track data in a consistent way and measure growth over time. This process also allows schools and districts to learn effective practices and scale these to more teachers through ongoing PD.

4. Are students developing positive attitudes toward, and identity with, CS?
   A primary goal for SFUSD’s CSforAll initiative is to help students develop positive attitudes toward and identity within CS, because we believe this will ultimately disrupt the inequitable trends of who takes and succeeds in CS at higher levels. To measure attitudinal and affective items, we can survey students at the beginning and end of each CS class. It is ideal to collect unique student identifiers to correlate to other demographic and performance data and to track data over time. We should analyze responses both in aggregate and by demographic subgroups, with particular focus to traditionally underrepresented groups. Examples of questions include:
   - Are CS classes interesting?
   - Do you think you can be good at programming/coding if you try hard?
   - Do you feel comfortable participating in your CS class?
   - CS can be challenging. What will you do when you get stuck on a hard problem?

5. What are students learning?
   Measuring student learning is particularly challenging given the emergent stage of K–12 CS education, lack of research-backed learning progressions, and how little is known about assessment, even at the higher levels. One way that we can measure student learning is through traditional (paper and pencil) assessments. SRI Education has created Thinking Outside the Box, which provides examples for Exploring CS and introductory middle school CS classes. Additionally, because of the highly creative and constructive nature of CS, we can also measure student learning through the students’ computational artifacts (e.g., programs). Artifact analysis is the most authentic and meaningful way to assess what students are learning and are able to do as a result of their CS classes. Examining student work across schools allows us to identify trends and make adjustments in instruction, curriculum, and PD.
CLASSROOM TOOLS
USING NICERC CONTENT TO SUPPLEMENT YOUR TOOLBOX

Chuck Gardner

As a classroom teacher, what resource do you most lack? If you’re anything like most teachers, you lack time. Time to adequately plan, grade, listen to student needs, and any other number of tasks that need attention.

Since leaving the classroom in 2015, and now as the Director of Curriculum for NICERC, the National Integrated Cyber Education Research Center, I’m working to provide teachers with easy, off-the-shelf solutions for one- or two-day classroom activities that require minimal preparation but provide quality learning experiences on cyber topics. This has resulted in a collection of lesson materials that enable teachers to easily access, download, and make available to students, lessons to fit various curriculum areas and recapture some of that precious time. Here is an example of such an offering.

The following content focuses on how students interact with their online environment, including social media, news-gathering, commerce, or any other number of online interactions. It’s applicable to a variety of curricular areas and learning settings.

In January of 2017, the Pew Research Center published “Americans and Cybersecurity,” by Kenneth Olmstead and Aaron Smith.

The premise of the article is that many Americans do not trust modern institutions to protect their personal data—even as they frequently neglect cybersecurity best practices in their own personal lives. The report paints a frighteningly self-confident image of American users of technology. While the internet-surfing public maintains great suspicion over government and private industry abilities to keep personal data secure, those same individuals are content with their own personal lapses in protecting their digital security.

Over 80% of Americans polled by Pew indicated that they either store their passwords in their heads or write them on paper.

What does that tell me? It tells me that they are using easy-to-remember passwords that can most likely be hacked by a bot or a hacker. Furthermore, the report states that 64% of Americans maintain at least one online account including accounts with banks or financial institutions, utility providers, healthcare providers, or some other types of institutions with which they make online transactions.

The data in the rest of the report is no less frightening. Nearly two-thirds have experienced some form of data theft, meaning they have personally been affected by a malicious event, yet still maintain relaxed standards when it comes to protecting their own data.

So, how is this information used in the classroom? The report is 43 pages. It may be a bit much to expect students to process the entire report. However, the paper is broken down into three sections: Americans’ experience with data security, password management and mobile security, and attitudes about cybersecurity policy.

One strategy may be to assign a single topic and have students identify best practices that they can prescribe as a public service announcement to be broadcast to their local community.

Another strategy would involve examining the survey and resulting data. The last 11 pages of the document detail the questionnaire that was used and the responses for each question. Students could look at the questionnaire in detail and identify any misleading or unclear questions. Or, they may want to use the survey with their own school or community and compare the results to the original study.

Finally, one of the best implementations of new humanities-based content in the classroom is to use discussion questions. As described by the Stanford University Teaching Commons blog, “a good question is both answerable and challenging. It will inspire analysis, synthesis, interpretation, and critical thinking.” However, to create a good set of discussion questions, the content should be read and well understood to the author of the discussion questions. To fulfill my promise to save time, here are some possible questions to analyze this article that you can use with your students.

1. If every online action can theoretically generate bits of data that can be stored by companies around the world, using reputable sources, calculate how much data is generated by the typical user of the internet over one day, one week, one month, and one year.

continued on next page
2. The report indicates that adults are reluctant to open online accounts because they are worried about how their personal information might be handled by the institution. What advice would you give that would reassure potential users? Conversely, what advice would you give to institutions seeking more online customers?

3. List reasons why people who have been the subject of a malicious online event continue to use passwords that are easy to remember.

4. What are some solutions currently available to those who want better password protection? Include computer-based or mobile-platform protection options.

5. Cybersecurity and the threat of cyberattacks are not simply domestic problems that only affect our country. Research a recent, major cyber event and discuss how its impact affected a global region and how the event was resolved. Answer questions such as these: Was it primarily private citizens or public industry that were affected? How many were affected? What was the dollar value of any damages suffered because of the attack?

For access to additional resources like the discussion questions above or to access our library of classroom content, please visit the NICERC website and request access to our content. NICERC provides hands-on professional development, curricula, programs, and competitions to engage students in STEM disciplines. Professional development programs for K–12 teachers enable them to motivate creativity and innovation in students through problem solving, critical thinking, and communication.

CSTA Thanks....

Organizations that provided conference attendee scholarships so that many CS educators can grow their CS educator super powers.

Google

100 SCHOLARSHIPS

Oracle Academy

35 SCHOLARSHIPS

Rolls-Royce

25 SCHOLARSHIPS

Facebook

10 SCHOLARSHIPS
Developing Talent
There’s No Shortage of Talent in Tech
Antonio Tijerino

Kaira Villanueva appears to be an average student at Columbia University, tracing her way through New York with a well-worn backpack, scratched-up MetroCard, and a youthful curiosity. Except she’s not typical. As a Latina computer science (CS) major, there is nothing common about her career path. And after being selected this year by the Hispanic Heritage Foundation (HHF) as a CSL (Code as a Second Language) Fellow, she’s more of a tech action hero. Kaira, and over a dozen other Latino programmers, have been tapped to be CSL Fellows to teach programming to Latino youth across the country. Kaira just finished teaching an eight-session course to a class of high school girls in New York to ensure there are more Kairas in our future. And more, Dantes...

Dante Alvarado-Leon is a student at UC Berkeley who has been a CSL Fellow for the past two years, since the program’s inception. He has taught at elementary, middle, and high schools all over Silicon Valley. He has also taught at the San Jose Mexican Consulate, having to take several buses to reach that community.

As America scrambles to find more programmers, it will take creative and resourceful approaches like the CSL initiative to empower more tech action heroes like Kaira and Dante. Tech action heroes are strategically leveraged to reach the imaginations of younger Latinos and to encourage them in the dynamic, but unfortunately exclusive, tech space—a space which is desperate for programmers, but seemingly not desperate for Latinos to fill the void.

According to the Bureau of Labor Statistics, over the next five years, 1.4 million new CS jobs will need to be filled in the US. But currently there are only about 400,000 CS students in the pipeline, and there are currently over 500,000 vacant tech jobs. It is baffling that nine out of 10 high schools don’t offer CS classes and in 33 of 50 states, CS classes don’t count as high school math or science graduation requirements (according to Code.org).

That’s right, in most states CS is treated more like a shop class instead of a math or science-based course. According to the National Center for Educational Statistics, CS/technology is the only STEM (science, technology, engineering, and mathematics) field in which student participation decreased over the last two decades. It’s no wonder that as a nation we’re so worried about competing globally.

And the long-term solution isn’t to search the globe for talent. The solution is right here in the US—in barrios, rural areas, urban areas, pretty much everywhere. HHF hosted a LOFT (Latinos On Fast Track) Coder Summit at Stanford, which has attracted over 1500 registrants over the past three years. All were programmers and nearly 40 percent were female. Some participants were overwhelmed when they saw that they were not the only Latino or Latina in a room of programmers.

The summit demonstrated that a base of Latino programmers exists and should be built upon and mobilized to grow the talent pool. This is the goal of the CSL program. Progress can be made by tapping individuals like Kaira and Dante. CSL provides them with platforms to teach younger Latinos about coding, spark an interest in tech careers, inspire them to innovate for social impact, and use tools to creatively express themselves in our changing world.

Latino youth currently represent nearly 25 percent of the student population. A new wave of Latino innovators, tech workers, and entrepreneurs need to be added to the burgeoning young Latino population to move America forward. And as we have met labor needs throughout history, Latinos will come through for the US in technology careers. But to do that we need to be more resourceful, creative, and actionable through programs like CSL and organizations like CSTA, in partnership with the private sector companies that understand the needs, as well as the opportunities.
In the summer of 2015, I had the privilege of teaching a very talented young man named Nathan. He was entering the fourth grade and attended my summer Scratch coding course at the Illinois Central College4Kids program. You never know where planting a seed will lead!

Since then, I have seen Nathan from time to time, but a recent encounter was very different. Nathan and his father ran up to me all excited to share the challenge the kids completed in their technology club. The “Animal Allies” challenge prompted Nathan’s team, the Firecrackers, to research video games and to learn Scratch as a tool to illustrate their project, because as Nathan said, “We wanted to reach a lot of people and everyone loves video games!”

The challenge was to identify a problem where people and animals interact and to design a solution. They selected the problem of declining Monarch butterfly populations and articulated the problem to the local Park District Board. In their presentation, they identified why the problem exists and why we should care about the problem. Then, they came up with a solution to help solve the problem.

Their efforts culminated in the dedication of the Monarch Butterfly Garden in Donovan Park in Peoria, IL, on May 1, 2017. The St. Jude Firecrackers and their families brought together the Peoria Park District, the Central Illinois Monarch Butterfly Task Force, and the Peoria Garden Club.

Nathan and his family have shared their excitement for Scratch and the club has branched out into exploring other coding languages. Nathan will be taking a more advanced coding class this summer at ICC College4Kids. It’s important that all K–12 students have an equal chance to experience coding, computer science, and the possibilities they present for a bright future. Planting seeds is critical; you never know where they will take hold or what they will yield.

LEARN MORE
Peoria Garden Club News Release
St. Jude Firecrackers Blog
SJFirecrackers on Scratch
Journal Star

CSTA Thanks....
Organizations that provide awards to recognize CSTA educator excellence and innovative computer science education.

Infosys Foundation USA
Awards for Excellence in Teaching Computer Science (10)

David Cutler and Gordon Bell
Cutler-Bell Prize for Student Achievement (4)
I love computer science (CS) books that approach the topic in new and novel ways. Over the past few months, I’ve collected a few to add to my “summer reading” book bag. I thought I’d share what I’m reading with you because I suspect summer is also your time to catch up on your reading and to gather new ideas for the next school year. The recent excitement about CS education has generated many great books. This is just a small sampling and I hope you are sharing your favorites in social media and through your CSTA chapters.

Happy reading!

**Coding iPhone Apps for Kids: A Playful Introduction to Swift**, by Gloria Winquist and Matt McCarthy. (303 pages)

Swift is a language used for mobile applications that run on all Apple devices. The book is written for new programmers, especially kids in upper middle school and older. But make no mistake, this is a serious CS programming book. Readers will learn about decision structures, loops, arrays, functions, custom classes, and a few unique Swift constructs. Fun app projects include a birthday tracker and a skateboarder game. The directions are clear and are supported by screen images. You’ll need a Mac running OS X 10.11.5 or later, Xcode, and the iOS SDK (available from the Apple App Store).

**The Mathematics of Secrets: Cryptography from Caesar Ciphers to Digital Encryption**, by Joshua Holden. (344 pages)

In this age, when almost everyone has experienced some “breach” of cybersecurity, you’ll find this book fascinating. The Mathematics of Secrets is a tour of the mathematics behind cryptography—the science of sending secret messages. You’ll recognize familiar ciphers (Caesar, polyalphabetic substitutions, autokey, and public key), learn about twists on the familiar, and discover some new ones. The book ends with a discussion of the future of cryptography. Students will need a background in algebra and a willingness to think hard and differently about math to get the most from this content. Holden maintains a blog with new developments and discoveries on the topic.

**Math Bytes: Google Bombs, Chocolate-Covered Pi, and Other Cool Bits in Computing**, by Tim Chartier. (130 pages)

Chartier provides a fun, hands-on approach to examining how math and computing are hidden in our daily lives in surprising ways. Each topic illustrates a famous mathematical theorem. Readers will examine how reposting a Tweet can ruin opening weekend for a movie, how to use math to find their celebrity look-a-likes, do calculus with a bag of chocolate chips, create fractal landscapes with a roll of dice, and explore several other concepts. Math lovers and “reluctant learners” will find plenty to enjoy in Math Bytes.

**Power-up: Unlocking the Hidden Mathematics in Video Games**, by Matthew Lane. (267 pages)

This book does not teach programming; it is more accurately described as math camouflaged within a topic that students love—video games. The goal is to examine interesting, and sometimes unexpected, bits of mathematics by analyzing video games. With entertaining descriptions, Lane discusses how players are working with the traveling salesperson problem in Assassin’s Creed, why it’s mathematically impossible for Mario to jump through the Mushroom Kingdom in Super Mario Bros., and how students are grappling with one of the most famous unsolved problems in mathematics and CS when they play Tetris. There are plenty more intersections of math and CS with games and life’s challenges in this engaging book. The nine chapters, including “The Thrill of the Chase,” “The Friendship Realm,” and “Order in Chaos,” can stand alone to support specific lesson goals.
Robots are a well-known educational manipulative for motivating students while integrating numerous STEM related disciplines that can promote an interest in math, engineering, science, and computer science (CS) concepts. Many of these programs are implemented in after-school programs, extracurricular clubs, and competition teams. Extra-curricular programs, competition teams, and camps are often inaccessible to poor and working-class families because of transportation issues. This can be especially true in rural areas that lack any form of public transportation. There can also be the cost of equipment, the presence of fees, and other expectations that are unreasonable for families of limited means. These factors can severely limit the diversity and the number of students that can be exposed to CS concepts at an early age.

The KISS Institute has been successfully collaborating with the Muscogee (Creek) Nation in Oklahoma for the past three years. The implementation of the Junior Botball Challenge program into the regular class day has addressed these fundamental limitations and has given all children the opportunity to participate. The Muscogee (Creek) Nation understands the importance of preparing their citizens with the skills necessary to obtain high paying jobs. Often, these jobs allow for telecommuting, providing them the opportunity to live and work within the Nation’s boundaries. With this partnership, the program is serving approximately 1500 students, over 400 of which are Muscogee (Creek) citizens in elementary schools and Head Start programs within the Nation’s boundaries.

The strategy revolves around three main components: educator-friendly reusable robotics equipment, a standards-aligned curriculum, and ongoing professional development.

Robotics equipment

Internet access and lack of technology is still a major factor limiting participation in many of the schools we are working with. We get around this by having the robot act as a Wi-Fi hub and serve a web-based IDE, allowing multiple (1–5) students to write and compile code (C, C++, Python) simultaneously. This lowers the initial cost of the reusable equipment to the school. It is also compatible with any device that has Wi-Fi capability (Chromebooks, laptops, tablets, and smartphones) without the need for any software downloads or internet access.

Curriculum

To implement the program into the regular school day, you must first garner administrative support. This is accomplished with professional development and curricular alignment with Common Core Math and Next Generation Science standards.

The modules, activities, and individual lessons are scaffolded for a continuum of higher-level thinking skills. Student skill acquisition and development include computational thinking, collaborative problem solving through algorithmic design (activities and challenges are completed as small teams), optimization of programs, and robotic designs that adhere to engineering constraints and goals. Applications of math and science concepts are identified and built into the lesson design. Formative assessments, with accompanying evaluation rubrics, are performance-based and built into each of the lessons and accompanying activities. Summative assessments include the completion of performance-based challenges through which student groups demonstrate their mastery of the concepts.

The Muscogee (Creek) Nation Junior Botball Challenge event, held in Tulsa, Oklahoma, had approximately 1500 students participate and it was attended by nearly 4000 parents, teachers, school administrators, and the public. The event built a sense of community around the importance of CS and STEM activities.

Professional development

Participating schools were asked to recruit teams of teachers so that they could work together and create a plan for co-teaching and collaborating as they implemented the program. The teachers attended a six-hour introductory workshop that was designed to meet the needs and address the concerns of participants who had little to no experience teaching CS or engineering concepts, or using
robots as a manipulative. It was important to not only address the educators’ self efficacy with technical concerns, (“I don’t know anything about coding or robots”), but also the educators’ concerns of implementing something new into their schools and classrooms. In addition to the introductory six-hour workshop, follow-up workshop opportunities (intermediate and advanced) were provided to help refine, reflect, and learn higher-level concepts. Ongoing support was provided through site visits, online meetings (Skype, Google Hangouts), email, phone, text, and cohort social media groups (i.e. Junior Botball Educators Facebook group). We heard encouraging comments from many participating teachers.

“My teaching really evolved in allowing students to solve problems rather than me showing them exactly how to do it. I loved that there were different solutions for the challenges. I feel like that is very real world, not everyone has the same solution, but we can both solve the problem.”

“It was an incredible way for our students to learn how to code. There are a lot of coding apps available. What I loved about this program was the collaboration between my students. My students showed excellent leadership skills, as they made sure that all their classmates learned to code. It was amazing to see students shine and excel.”

The combination of robotics equipment, standards-aligned curriculum, and ongoing professional development for the educators, has led to a successful and scalable model for providing equity in participation to students who would not have had the opportunity to participate or the exposure to CS concepts at such an early age.
CSTA Thanks...

The 2017 Conference sponsors that make this premier event possible for so many educators.

- ACM-W
- Bootup PD
- Certiport
- Cyber Innovation Center-CIC
- CodeCombat
- CodeHS
- CoderZ by Intelitek
- Codio
- College Board
- Dexter Industries
- Digital Media Academy
- DK Penguin Random House
- Georgia Institute of Technology
- Google
- IBM
- Infosys
- Johns Hopkins Center for Talented Youth
- KOFAC
- Microsoft
- NC Lab
- NCWIT
- NICE
- Oracle Academy
- Project GUTS
- Rising Stars
- SPEL Technologies
- Stylus Publishing
- SUNY Albany
- TEALS
- Ten80
- ThinkFun
- University of California, Berkeley
- UTeach-Austin
- VidCode
- Wonder Workshop
- Zero Robotics
- Zulama
MARK YOUR CALENDAR

NICERC Cyber Showcase
July 8–9, 2017
Baltimore, Maryland

CSTA Annual Conference
July 8–11, 2017
Baltimore, Maryland

NICERC Education Discovery Forum
July 12–14, 2017
Baltimore Maryland

Logic is Fundamental
July 31 - August 4, 2017
Stanford University

CS4NH Summit
Fall 2017 (date TBD)
Berlin, New Hampshire
Contact David Benedetto

CSEdWeek
December 2-17
In your community

CSPDWeek
July 17-21, 2017
• Colorado School of Mines, Golden, CO: Bootstrap, BJC, NCWIT, and others
• College of St. Scholastica, Duluth MN: Mobile CSP
• University of Texas, Dallas, TX: UTeach AP CSP
• University of California, Berkeley, CA: Beauty and Joy of Computing

Check the most recent CSTA events on the CSTA website
List your CSTA event by contacting:
customerservice@csteachers.org

Be a CS Hero
Order your CS Hero t-shirt online today—only $25!
Proceeds to benefit CSTA PD Events
Only available online through the CSTA T-shirt Campaign

TEAM SUPERHERO.
Show Your CS Pride.... Order Your T-shirts Here!