Dear CSTA Members,

As the new chair of CSTA’s Board of Directors, I want to thank you for being part of CSTA. Sharing your wisdom and encouragement with your fellow computer science (CS) teachers is what makes CSTA a supportive and valued community. Here’s what’s been happening with your CSTA.

We had our biggest Annual Conference ever in July 2017. The conference theme was “Identifying CS Teachers’ Superpowers,” and you delivered in Baltimore! The conference hotel and venue were fabulous and Baltimore was a fun and friendly city.

We had 665 attendees with 170 of them attending on scholarships from our sponsors. The exhibitor space with 42 exhibitors was centrally located and a continuous hub of activity. One of my favorite experiences was a VR tour inside a helicopter from our sponsor, Rolls-Royce. I “walked” around, venturing into the helicopter and into its engine—it was amazing.

There were 20 workshops and 40 conference sessions. The two keynote speakers, Dr. Freeman Hrabowski, President of the University of Maryland Baltimore County, and Brenda Wilkerson, Director of CS and IT Education for Chicago Public Schools, gave thought-provoking and inspiring messages.

Mark Nelson, our Executive Director (ED) for the past two years, left CSTA on July 31. I’d like to thank Mark for his dedicated service to CSTA. During his tenure, CSTA grew its conference participation, brought in new sponsors, and connected hundreds of CS teachers to a variety of professional development opportunities.

We recruited Deborah Seehorn to serve as Interim ED. Deborah is widely respected across the CS education community, and with her deep knowledge of the CSTA organization, I am happy to have her take the helm of CSTA during this transition. I’ve been working closely with her the past month, and I can tell you she is doing a great job.

We are presently conducting a search for a new ED. Detailed information about the position is posted on the CSTA website. We want the best possible person as our new ED, and I invite all of you to share this opportunity with your colleagues.

Thank you for your contributions to CSTA.

Fred Martin
Chair, CSTA Board of Directors
A message from the CSTA Interim Executive Director

Hello CSTA Members,

CSTA Board Chair Fred Martin emailed me in late July and asked if I could spend a few minutes talking about CSTA. I assumed that he had questions about the recent CSTA Annual Conference or perhaps about his new role as Board Chair. He informed me that Mark Nelson was about to resign as CSTA Executive Director (ED). Imagine my surprise when Fred then asked me to serve as the part-time CSTA Interim ED until the search committee would be able to hire a new ED.

My initial reaction was to remind Fred that I was retired and ask him if he had considered anyone else. (Fred thought that retirement made me the perfect candidate!) He shared the duties the Interim ED would be expected to perform, and I contemplated the decision for about a week, conversing with Fred, as well as ACM CEO Bobby Schnabel. I certainly want CSTA to continue the great work the organization has been doing to support our computer science (CS) educators, so I accepted.

I have spent my first few weeks meeting with our CSTA Board of Directors, staff, sponsors, and partners on the phone or in virtual meetings. I have also had the pleasure of speaking with many of our CSTA committee chairs who are passionate about CS education and have great plans for advancing CSTA and CS education. If you are not involved with one of our committees, I urge you to volunteer.

Additionally, I have been working to ensure that the current CSTA projects and initiatives can continue to progress and benefit our members. It has been an exciting and challenging time for us and our work remains so critically important. Together we can work to ensure that the CSTA mission to empower, engage, and advocate for K–12 CS teachers worldwide will continue.

I look forward to working with you and I wish you a wonderfully successful academic year. Thank you for all you do for CSTA and for CS education!

Thank you,
Deborah Seehorn
CSTA Interim Executive Director
Did you know some of your greatest allies in advocating for computer science (CS) might work right alongside you every day? CS educators across the nation are finding that collaborating with their school counselors yields positive results in directing students to viable education and career opportunities.

Counselors care about equity and social justice, and are motivated to guide students to studies that prepare them for quality jobs—and many of those jobs are in the computing fields. Here are a few key strategies for creating a positive partnership with your counselors for CS advocacy in your school.

1. Demystify CS – Paint a fresh picture of who’s right for computing

Counselors more effectively guide students to your courses when they understand what CS is and what the courses are like. Further, counselors may have a narrow sense of who is “right” for computing and might not realize how CS can intersect with the diverse interests of every student.

Rather than simply explaining what CS is, invite a counselor to your classroom for an introductory lesson or “unplugged” activity. Many counselors report having a greater understanding of computational thinking after a hands-on experience.

Encourage excited students to drop by the counseling center and share what they created with CS. When students explain what they enjoy about studying CS to their school counselors, counselors can pass along that information and the enthusiasm to other students potentially interested in CS.

2. Use demographic data – Make CS favorable on the master schedule

When school counselors develop the master schedule, they base it on aggregated data reflecting the academic needs and interests of students. With input from you, your counselor can adjust course scheduling to avoid conflicts that adversely impact diverse enrollment in your CS courses. (Is the single choir class offered at the same time as your single CS class? They can adjust the master schedule to alleviate such conflicts.)

Curious about how your school's data compares to your state statistics? Code.org/promote has free, printable state-specific fact sheets. Share this information with school counselors as you discuss how school-based changes can align with state and national trends and initiatives.

3. Recruit strategically

Is your class as diverse as you’d like it to be? Share your class roster with your school counselors and together set a recruiting goal to make the composition of your class more representative of the demographics of your school.

As you discuss recruiting students into the course, share data about the underrepresentation of women and other groups in computing, and use NCWIT's informative, printable infographics to guide discussion:

- Top 10 Ways of Recruiting High School Women into Your Computing Classes
- Girls in IT: The Facts Infographic
- Top 10 Ways to Engage Underrepresented Students in Computing

One small thing school counselors can do to increase enrollment in CS classes is to recruit students in pairs or groups. Students from traditionally underrepresented groups feel a greater sense of belonging when they are in a class with familiar friends or with students who have shared interests or similar backgrounds.

4. Imagine CS for All – Whether students pursue CS or any other field

You know CS isn’t useful only for students who plan to go on to major in CS, but many school counselors are not aware of how CS intersects with other students’ interests. Share with them Google’s Careers with Code magazine, available online and in print, which illustrates a multitude of expressions of CS that span a wide array of interests.

NCWIT Counselors for Computing provides free resources and professional development for anyone in an advising role (including you), which support students as they explore CS education and careers. Explore the website and resources with your school counselors and order a free Counselors for Computing kit, which has talking points, pathways cards, posters, and other exciting resources.

Check out, too, NCWIT’s What’s Your Coding Super Power? poster, which you can download and print. It helps students see how they can combine CS with things they’re passionate about to make real differences in the world.

Final Thoughts

You might not know it, but counselors are obliged to provide career counseling to students every year at every grade level. As does the general public, many counselors have a narrow view of the computing professions and aren’t aware that all job sectors involve computing in some way. You can help them understand that having a background in CS can make your students highly employable and provide them access to creative, gratifying, lucrative, and stable careers.

Take a moment to meet with your school counselors. Through your collaboration, you can expand student access to CS classes in your school and open doors to future career opportunities in computing.
Project Malawi: A Success Story

Dave Ames

I currently work at the University of Manchester supporting teachers across the North West of England on behalf of Computing at School (CAS), as well as working with a variety of groups through the University and other organizations. Late in 2016, the CAS Team was approached by colleagues from within the University with a request for help in planning a project to send undergraduates from the School of Computer Science (CS) to Malawi to volunteer with a charity called RIPPLE Africa. We christened the project with the imaginative title "Project Malawi" and set about examining the details of what support the undergraduates could offer and what was feasible in the times available.

As planning developed for Project Malawi, it was decided that we would work with three secondary schools that RIPPLE Africa has connections to: Kapanda Secondary School, Bandawe Mixed Secondary School, and Bandawe Girls Secondary School. It was also decided that some of the CAS Team at the University would accompany the undergraduates to Malawi. The final version of Project Malawi consisted of three undergraduates and myself in Malawi for three weeks. Two teacher colleagues from CAS and Dr. David Rydeheard, who oversees our CAS Regional Center, joined us for a week.

Throughout our time in Malawi, my CAS colleagues Sarah Zaman, Jo Hodge, and I delivered initial lessons to the pupils; subsequent lessons were then delivered by the undergraduates with our support.

Over the three weeks, we delivered a variety of lessons. Initially these were unplugged, computational thinking-based lessons, but after the first four or five days we moved on to some basic programming.

One obstacle, which we hadn't anticipated, was the pupils' lack of experience using a mouse to drag and drop things on a screen. This was doubly a problem with pupils using the laptops, as they tried to drag and drop using the laptops' touch-pads, something that almost none of them had done before.

After the initial Scratch sessions, we then moved on to some physical computing using the Crumble, Sparkles, and motors from Redfern Electronics. These devices use a free, block-based programming environment similar to Scratch or Blockly to allow the user to interact with their environment through the Crumble board. The undergraduates' newfound familiarity with Scratch allowed us to get going very quickly. We used the boards to create a simple night-light (using a motor and an RGB LED) and programmable robot cars.

were donated by Lowerplace Primary School in Rochdale. We were also lucky enough to acquire six pi-topCeeds, which gave us a total of 12 computers. The first programming sessions we delivered used Scratch version 1.4. We taught a few existing lessons from CodeClub, which the undergraduates found very accessible.
During the final week of our trip, we introduced the pupils (and their teachers) to the micro:bit, a programmable board with a variety of built-in sensors and LEDs. The Micro:bit Educational Foundation and the Lancaster University had kindly donated a number of units for us to use. Although most of the programming resources around the micro:bit use web-based programming environments, we were able to use the Mu offline editor which uses MicroPython to do some basic Python programming with the pupils.

Another unanticipated difficulty was that very few of the pupils had ever used a keyboard before, so not only was this their first experience programming in a text-based language, but also their first experience typing. The pupils were very excited by things as simple as having their names scroll across the LED matrix of the micro:bit and were quickly able to develop ideas for displaying different messages, depending on the physical input that was detected.

Despite a range of difficulties, including a national teacher strike, termite-riddled classrooms, no CS teachers, power blackouts, dodgy (i.e., dangerous) electrical outlets, lack of equipment, no internet access, and the pupils’ lack of keyboard and mouse skills, we successfully delivered approximately 80 hours of teaching in over 50 sessions to approximately 200 pupils during the three weeks we were there.

Currently, we are evaluating the project with a view to repeating the experience with more pupils and teachers next year. Our plan is to train some local teachers to deliver programming lessons. We are also considering the possibility of supporting RIPPLE Africa in funding and equipping a computer lab and library at Kapanda Secondary School.

For more details on this and how you might contribute, see the CAS Manchester website.
Have you ever wished for a place where you could get answers to your teaching questions from experienced educators? Is the time between CSTA Voice issues too long to go without reading about topics in computer science (CS) education, such as pair programming, innovative assignments, or new tools to teach more effectively? Have you ever had a success in the classroom that you wished you could share with other teachers?

There is now a website that meets this need: Computer Science Educators (CSE) Stack Exchange, part of the family of Stack Exchange sites that includes Stack Overflow (for programming questions). CSE was launched in May and by mid-August had over 300 questions, 1500 answers, and 2,500 users.

While it’s impossible to know the demographics of the site, since most users do not include their profession in their profile, the users include high school teachers, professors (from community colleges, undergraduate institutions, and universities), teaching assistants, people interested in becoming CS educators, self-teachers, retired educators, computer professionals, and students, who often have surprisingly good insights on how to teach effectively.

Questions and Tags

Every question is labeled with one or more tags. To give an idea of the site’s content, here are the most popular tags and some sample questions:

- Curriculum-design: What would be reasonable to include in an Introduction to AI class for a 7th–8th grade after-school program?
- Lesson-ideas: How can I stress the importance of testing code?
- High-school: Should CS students be doing their tests on paper?
- Best-practices: Is it important to teach pointers in a first course using Java?
- Teaching analogies: What is a good analogy for “pass by value” vs. “by reference?”
- Student motivation: How do you encourage students to accept constructive criticism of their code?
- Java: Do Python and Java lead students to construct different mental models of memory?
- Resource requests: What resources can be used to explain how the internet and the World Wide Web work?

Stack Exchange sites are not appropriate for all questions. Surveys and questions with purely subjective answers are not allowed. For example, “What’s the best programming language?” would be put on hold by the moderators for being too broad or primarily opinion-based. (Questions that are put on hold can be reopened after their problems have been fixed.) While subjective questions are allowed (and can be the most interesting), they need to be written in a way that requires answers to be backed up with facts and experiences, supported opinions. For example, the following would be appropriate refinements to the “best programming language” question.

- What language should I use for an after-school program for teens interested in writing games?
- What’s a good next language for middle schoolers experienced with Scratch?

Of course, in addition to titles and tags, questions have bodies where further details can be provided, such as the amount of instruction time available and any specific learning objectives.

Answers

On average, each question has about five answers, which can be many paragraphs long and include special formatting, pictures, and links. The first question on the site was about auto-grading software for Java programs, and each of the seven answers describes the strengths and weaknesses of a different system, some including screenshots and configuration examples. In two cases, answers were written by the creators of grading systems, who disclosed their connection.

The amount of time and care that users put into answering people’s questions is impressive. Readers can upvote answers (and questions) that they deem helpful, and the original poster of a question can select one as the best answer. It is also possible to leave comments on an answer (or question), to ask for clarification or to suggest an improvement.
Advice for new users

The first thing a user needs to decide when setting up an account is a screen name. While many users post under their real names (some of which will be familiar to CSTA and SIGCSE members), others prefer to use a pseudonym, to protect the privacy of their students, their colleagues, or themselves. There are many ways to learn how to write a good question, including:

- browsing questions to get a sense of site style
- reading the FAQ on asking questions
- posting a question in the Question Sandbox, to get feedback from experienced users

Of course, it’s also possible to dive right in and post a question to the whole community. While the amount of time to get an answer can vary from minutes to days, new users’ questions usually quickly receive friendly comments welcoming the users to the site and letting them know any ways their questions can be improved. Unlike some forums that are unwelcoming to “newbies,” the administrators of Stack Exchange recently called out CSE for being particularly welcoming to newcomers.

Chat rooms

Little-known features of Stack Exchange sites are chat rooms. CSE has two.

- The Classroom is a virtual teachers’ lounge and a great place to get to know other users. Expect puns and humor, as well as informal discussion of how to improve the site.

- The Guidance Office is a place where new users can get help, if desired, writing their first questions. Experienced users of the site volunteer here because they want new users to feel comfortable participating. Posts to the Question Sandbox are automatically forwarded to this chat room.

As we often tell our students, programming is a team sport. The same should be true of teaching programming and other aspects of CS. There have been times in the past few months when I have struggled with how to explain a concept, and it’s been great to have other teachers quickly respond with helpful suggestions. I also find it rewarding to share what has worked (or hasn’t worked) in my classroom. The next time that you need—or have—a great teaching idea, or just want to see what’s on other teachers’ minds, please stop by the CSE Stack Exchange.

2017 CSTA Annual Conference Presentations Now Available Online!

The 2017 CSTA Annual conference may be history, but you don’t have to miss the inspiring, high-quality, professional development enjoyed by participants. The presentations are now available from the 2017 CSTA Annual Conference site.

Download a list of links to the presentations.

View videos of keynote addresses.
The 2017 CSTA Annual Conference was a huge success—no matter how you measure it. Here are a few statistics to ponder…

- 665 attendees
- 42 exhibitors
- 170 attendee scholarships from our sponsors
  - 100 from Google
  - 25 from Rolls-Royce
  - 35 from Oracle Academy
  - 10 from Facebook/TechStart
- 20 workshops
- 40 sessions
- 10 Excellence in CS Teaching Awards from Infosys Foundation USA
- 3 Cutler-Bell Prizes awarded to students
- 583 downloads of the CSTA2017 Conference app
- 425 attendees at the Oracle Academy Reception on Sunday evening

WORDS FROM ATTENDEES

“This was my 5th CSTA and by far was the best!”

“Awesome from start to finish...wish I had a clone so I could have attended multiple concurrent sessions.”

“The number of middle school topics was fantastic!! I haven’t been able to find that elsewhere.”

“Super fun! Loved the sessions and I always appreciate the collegial conversations.”

“The sessions were varied, allowing opportunity to address diverse needs.”

“Top-notch...whoever vetted the presentations did an exceptional job!”

“I thought they all did a thorough job presenting their topics. I felt their enthusiasm and cannot wait to improve the program in my school, district, and state!”
Learning to program is more than simply learning to write code. It requires the development of computational thinking, a problem-solving process comprising skills such as logical reasoning, decomposition, algorithmic thinking, pattern recognition, and abstract reasoning. For many students, this is an unfamiliar way of thinking that can require adaptation.

Grok Learning aims to help students (and adults) develop their computational thinking skills, and to get excited about what these skills can help them achieve—think cancer cures, space exploration, AI-assisted language technology, or tech entrepreneurship.

Our approach to teaching students of all ages is different because we don’t shy away from the fact that coding and thinking computationally can be hard! Our aim is for students to discover useful solutions to real-world problems with an understanding of how coding and computers work. That is what will really empower them to be creators of technology, rather than simply consumers.

We’ve created an interactive and intelligent platform to support classroom teaching and self-directed learning of computer programming and related technologies, from upper elementary and beyond. Right from the beginning, students learn how to use code to solve problems.

Our classroom-ready learning materials use real-world examples, and we encourage students to apply computational thinking and data science to their interests—from farming to fashion, art to accounting, and everything in between.

Our courses and competitions allow students to learn to program at their own pace. They work through interactive notes written by curriculum experts and receive detailed feedback about their progress from our intelligent auto-marking system. If they have questions, we also offer live tutor support during our annual competitions.

We’re committed to supporting teachers in and outside the classroom. We provide access to our resources at no cost for their professional development, and run regular professional development workshops. In the classroom, teachers can track their students’ progress, so they have an accurate, live-updating snapshot of how the class is learning, and know where to focus their attention for maximum impact.

By supporting both a top-down (teachers and educators) and bottom-up (students and self-learners) approach, we’ve seen fantastic increases in creativity, confidence, and excitement about computational thinking and computer science.

Our annual competitions are ideal for teachers and students just getting started. Unlike other coding competitions, students don’t need to know how to code to take part. Instead, we teach students to program as they progress, with detailed notes and engaging problems to solve.

Competitions offer fantastic opportunities to get classes engaged and excited. They’ll be learning alongside thousands of other students, and be supported by our tutors along the way. Our competitions are friendly, and we encourage students to work together. Students can earn points by solving problems and receive a certificate at the end to mark their achievement.

If the competition timing doesn’t work for you, we also offer self-paced courses, available year-round. Students can learn to program in Python or Blockly (a visual programming language with drag-and-drop blocks), write web pages in HTML and CSS, or program an embedded device, such as the BBC micro:bit.

---

**GROK LEARNING: MAKING CODING FUN**

Christie McMonigal
The beginning of the school year is an exciting time for educators. It’s when we welcome new students, try out new ideas, and take on new challenges. The curriculum team members at Code.org may no longer have our own students, but as the school year approaches, we’re filled with that same anticipation. This is especially true this fall because after years of hearing about the need for computer science (CS) resources designed for middle and early high school, we’re ready to share our newest curriculum, CS Discoveries.

This new full-year course, targeted toward seventh to ninth graders, is chock full of fun activities, programming tools, and resources. The result of eighteen months of work, a national pilot, and multiple iterations on lessons and tools, CS Discoveries features six units that can be taught as a year-long class or two separate semesters. Beginning with a fully-unplugged unit that builds students’ problem-solving skills and a collaborative classroom culture, students embark on a journey that includes web development, animation, game creation, user-centered design, data analysis, and for the first time ever in a Code.org course, physical computing.

The curriculum features robust supports for teachers and students alike. Teachers will find detailed daily lesson plans, handouts, videos, projects, rubrics, and exemplars. Students can look forward to a variety of engaging activities and new programming tools to support learning and channel creativity.

As we look forward to the coming year, we’d like to share a bit about how we designed the course, what we learned along the way, and what you and your students can look forward to in a CS Discoveries classroom.

Seventh to ninth graders are in a time of transition. They want independence but still need structure and support. They are increasingly aware of the world around them but remain primarily motivated by their own interests. Their attention spans can be frustratingly short, but they’re also making decisions about what fields they’ll pursue for years to come. It’s a notoriously tricky (though rewarding) age group to teach, and we knew going in it would be a challenging one to write for.

To meet this challenge, we began with a core set of values, which we embedded throughout the course. We knew we’d need to focus on a positive classroom culture with activities that emphasize collaboration and social development. Students this age want to have an impact on the world, so our projects and tools needed space for them to share their creative voices. With more schools across the country adopting CS, we needed to provide resources and activities that would be effective in a diverse set of classrooms, particularly those without CS experience.

While the core writing team on this project was small, this curriculum is truly a community effort. Last summer we were joined by half a dozen writers who contributed their experiences as classroom teachers. This spring we ran a high-touch pilot with over forty teachers from diverse classrooms around the country. Hundreds more teachers used the materials as we published and updated drafts. In many cases, the input from these groups fundamentally changed the way we designed the course.

This extensive community involvement highlighted the strengths of early drafts and helped us identify where we could improve. We learned that our more open-ended or abstract problem-solving lessons weren’t landing with many students. Early programming progressions took far longer than anticipated, and students wanted more time to play and create. Meanwhile, our engineering team received hundreds of ideas, big and small, on how to improve our tools. This feedback was critical to informing improvements for the course’s official launch this summer.

If you’re interested in bringing CS Discoveries into your classroom, you can check out all our freely-available materials at Code.org. We will continue to improve the course throughout the school year, and we’d love to know how it’s working in your classroom. Feel free to ask questions in the forum, or tweet us at @teachcode and #csdiscoveries to share what your students are doing. We built this course for you and your students. Just like you, we have high hopes for the new year, and we’re excited to see CS Discoveries be a part of it.
What drives a high achieving talent pool? Learning and development leaders would naturally say the quality of the learning and development programs they offer. But what about all the learning experiences individuals have prior to arriving in a specific organization?

What if we went as far back as primary and secondary education? What drives K–12 achievement? While the answer to this question is quite complex and beyond the scope of this article, one component of student achievement deserves attention: professional development (PD) programs for K–12 teachers. If misalignment of learning to strategy is a challenge for the Learning & Development (L&D) community, it is gargantuan for the K–12 PD arena because of the relative lack of data and research. The most recent research study conducted by the US Department of Education is dated in 2007, yet it shows a significant metric that is still relevant today: teachers who receive quality PD can positively impact student achievement by 21 percentile points.

Corporate learning leaders traditionally focus on developing and delivering quality learning and development programs for their organization’s talent to increase performance, productivity, and profitability. In the K–12 world, school principals focus on developing and delivering quality PD programs for teachers, who teach the K–12 students, who become the talent pool chief learning officers (CLOs) care most about.

Today, CLOs and corporations collaborate heavily with universities on this topic, but they do not seem to engage much, if at all, with K–12 principals and administrators. Yet, that is perhaps where the conversation and the collaboration should begin.

One of the biggest challenges in K–12 teacher PD is funding. While several corporates and foundations offer scholarships to teachers, there is a significant gap in funding and access to PD programs for K–12 teachers. The return on investment for K–12 teachers’ PD is 25x. Many corporates miss this multiplier effect by investing only in individual students’ education. Whereas, a teacher can impact more than 25 students at a time and increase student academic achievement by 21 percentile points each.

Who in corporate learning is best positioned to begin this specific conversation? The chief learning officer, of course. Who in K–12 is best poised to engage in such a conversation? K–12 administrators, principals, and teachers, of course. What if learning leaders started a specific and grounded conversation with teachers’ associations and K–12 administrators and school principals around the intersection of L&D and PD on key areas of learning that K–12 students, tomorrow’s talent pool, will need to succeed in the knowledge economy? Such a conversation, if not a collaboration between these groups, could be game-changing at many levels, starting with best practice sharing and articulating skill needs for today’s K–12 students and tomorrow’s talent pool. Learn more how CSTA is changing this narrative with the launch of the PD Pipeline.
NCWIT Aspirations in Computing (AiC) program was designed to inspire young women in computing by providing recognition, encouragement, and visibility. AiC is a talent development initiative of the National Center for Women & Information Technology. The NCWIT Award for AiC honors girls in grades nine through twelve for their computing-related achievements and interests.

Applications are now open.

The Award for AiC offers both a national and regional competitions to increase young women's participation in computing nationwide. Last spring, almost 2,800 women were recognized; they represented all 50 states, the District of Columbia, Puerto Rico, the US Virgin Islands, and US overseas military bases.

We want to honor even more students this year and that means we need you! More than two-thirds of applicants said they applied because they were encouraged to do so by an educator. You inspire these young women every day. It is through your innovative teaching that they see the value of computer science education.

Past Award recipients report becoming more “proud, inspired, determined, confident, excited, and encouraged” about futures in computing from receiving the Award.

In addition to identifying a pool of talented women, NCWIT also recognizes outstanding educators like you who play a pivotal role in supporting young women to continue exploring their interests in computing and technology. Through the NCWIT AiC Educator Award, recipients are awarded cash and professional development funds and are honored at the regional award ceremony alongside the students.

This year, we are recognizing an educator at the national level. The National Award winner receives a cash prize and a trip to the NCWIT Summit on Women and IT in Grapevine, Texas, next May.

“This has opened up my mind to finding new ways to help these ladies achieve goals they once thought were impossible,” one Educator Award winner reported.

Applications for the 2018 NCWIT Award for AiC and the NCWIT AiC Affiliate Educator Award are now open. For applications or information visit NCWIT Aspirations in Computing or contact aspirations@ncwit.org.

CSTA is headed to Omaha, Nebraska

Join us for the 2018 CSTA Annual Conference

July 7–10, 2018
STUDENT COMPETITIONS

CELEBRATING 40 YEARS OF ACSL

Carlen Blackstone

For the 40th year, the American Computer Science League (ACSL) will once again motivate students in grades 3–12 to develop foundational computer science (CS) concepts and coding skills. ACSL organizes CS contests for elementary, junior, and senior high school students.

There are four contests held in the students’ classrooms from December through April. A final “All-Star” contest is held in May at a designated US location. Over 200 teams in the United States, Canada, Europe, Africa, and Asia have participated in recent years. ACSL contests offer a variety of options to fit your classroom needs.

1. The Elementary Division introduces (for students in grades three to six) foundational CS concepts without requiring students to know how to code.

2. The Classroom Division provides a non-programming version of the contest using the 12 concepts from the other divisions for grades 6–12.

3. The two-part, short-problem and programming-problem contest includes a Junior, Intermediate, and Senior Division so that students can participate based on their grade and experience level.

4. Schools may register for either a three-person or five-person team competition in the Intermediate and Senior Divisions.

One of the greatest challenges we have today in the high school classroom is to make the connection between foundational CS skills and the plethora of sophisticated software applications that students use on their smartphones and laptops. In the 1980s, the software they could create rivaled what was available commercially. Thankfully, we can now entice and motivate students with drag-and-drop languages such as Scratch, SNAP, Alice, App Inventor, and Jigsaw that help close the software sophistication gap. However, the need for foundational knowledge still remains.

As I embraced the many changes in the field, from web development to OOP techniques and app design, I also became more convinced than ever that the basic concepts of algorithmic design, the binary number system, Boolean algebra, recursion, and effective use of data structures are extremely important building blocks. Teaching these concepts early gives students a rich appreciation for computer history, as well as a foundation for additional CS learning.

I think that is one of the main reasons I chose to participate in the ACSL. The 12 conceptual topics have not changed much in 35 years, nor has the nature of the programming problems provided. However, I have had thousands of students learn college-level material that has helped them in a variety of fields. The annual All-Star contest trips have also provided a wonderful opportunity for my students to interact with, and be challenged by, the best students in the world.

Because the specific technology students are learning in your classroom today will likely not be what they will use in just a few years, I encourage you to give them the strong conceptual and problem-solving background provided by the ACSL materials and contests.

ACSL is on the approved activities list of the National Association of Secondary School Principals (NASSP). ACSL is also an institutional member of CSTA.

If you are interested in participating in ACSL this year, please visit the ACSL website for more detailed information and contest schedule. If you have additional questions, contact us at info@acsl.org.
CSTA Thanks...

CSTA sincerely appreciates the support of our 2017 Annual Conference sponsors

- ACM
- Code HS
- Google
- Infosys
- Microsoft
- Oracle Academy
- Rolls-Royce
- University at Albany
  Department of Education
MARK YOUR CALENDAR

CS4NH Summit
September 30, 2017, Berlin, NH

Aspirations in Computing and Educator Awards
Nominations open September 1 through November 6, 2017

CAS Northfleet Hub Meeting
September 7, 2017, Gravesend, UK

CAS Chichester-Bognor Hub Meeting
September 11, 2017, Bognor Regis, UK

An Introduction to the BBC Micro:Bit
September 12, 2017, Manchester, UK

#csk8 Twitter Chats
Dates starting September 20, 2017

Degreed LENS Conference
September 27–28, 2017, Chicago, IL

Grok Learning Competitions
October 2017 through February 2018

CSforAll Summit
October 16–17, 2017, St. Louis, MO

More CAS Community Events
CS education events in the UK

CSEd Week
December 4–10, 2017, in your community

ACSL Contest #1
December 22, 2017, in your classroom

ACSL Contest #2
February 9, 2018, in your classroom

ACSL Contest #3
March 9, 2018, in your classroom

ACSL Contest #4
April 12, 2018, in your classroom

2018 CSTA Annual Conference
July 7–10, 2018, Omaha, NE

Find more upcoming CS 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!