

# Voice

The voice  
of K-12  
computer  
science  
education  
and its  
educators

Volume 14  
Issue 4  
November 2018

## FEATURED ARTICLES

Hour of  
Code  
"Unusual  
Discovery"

Nominate  
Your  
Students  
for the AiC  
Awards

CSEd Week  
Driven  
with a  
Theme

Using Zork  
to Inspire  
the Future

Attending  
AP Reading  
Impacts  
Teaching

Culturally  
Responsive  
Teaching

Where Have  
All the  
Teachers  
Gone?

Governors  
for  
Computer  
Science

## Hour of Code "Unusual Discovery"

YVONNE MELTON

Lead your students to "An Unusual Discovery" for Hour of Code 2018! Anticipation is building for the upcoming CSEd Week 2018 and Google is excited to once again partner with CSTA in supporting teachers and will be launching a new activity to celebrate Hour of Code, called "An Unusual Discovery."

In the activity, two characters meet in a world and discover a surprising object. Students get to decide what happens next by creating a story with code. Using basic computer science

(CS) concepts such as events, sequencing, and conditionals, students use their imaginations and creativity to bring the interaction to life. As with other CS First projects, this stand-alone activity is part of the free, video-based, introductory CS curriculum designed to engage a diverse student population in grades 4 through 8 (ages 9–14). Learn more and try it out at: [g.co/csfirst/CSTA-HOC18](http://g.co/csfirst/CSTA-HOC18).

As a long-time founding partner of CSEd Week, Google is excited to support CSTA members who lead this activity for Hour of Code in their classrooms, clubs, and libraries. We'll be partnering with CSTA on funding opportunities, swag, and support for new CSTA+ members around Hour of Code in the coming weeks.

If you are interested in learning more about CS First in the classroom, visit our new teacher-support page to find out about professional development opportunities, [training materials](#), [CSTA and ISTE standards alignment](#), [how other teachers are using CS First](#), and more!

## CSTA 2019 Conference Call for Proposals Now Open

Apply now to present your great computer science (CS) teaching and CS education leadership strategies at the 2019 CSTA Annual Conference, July 7–10, in Phoenix, Arizona! Submit proposals based on the work you're doing in your classroom or community. Proposals should include content and best practices for K–12 CS teachers, from great new ways to teach CS content to integrating CS into elementary grades or building a strong CS program in your school or district. Visit [cstaconference.org](http://cstaconference.org) for more details and to submit your proposal.

**Proposals will be accepted until 11:59 PM (Pacific Time) on December 2, 2018.**

## CSTA Thanks...

2018 CSTA  
Conference Chairs

**Daniel Moix**  
Program Chair

**Tammy Pirmann**  
Workshop Chair

Checkout the  
RESOURCES at  
CSTeachers.org

- [Publications](#)
- [CSTA K-12 CS Standards](#)
- [Align Your Curriculum](#)
- [Highlighted Resources](#)
- [Research and Reports](#)
- [Grant Support Request](#)

# Nominate Your Students for the AiC Awards

Teachers are often some of the strongest advocates for their students' success. And your encouragement and daily support can help inspire them to pursue new opportunities and undiscovered talents for the first time.

The National Center for Women & Information Technology's (NCWIT) [Aspirations in Computing \(AiC\)](#) program was designed to support young women in computing by providing recognition, encouragement, and opportunities to jobs, scholarships, and connections to the tech community. We've recognized over 10,000 young women in grades 9 through 12 for their aspirations and passion in computing and built a supportive network in each of our [79 regional affiliates](#) of parents, industry professionals, community leaders, and educators all working together to increase the meaningful participation of

women in computing and technology within their communities.

[Applications are now open](#) and we need your help to encourage the women in your classroom to apply. More than two-thirds of past applicants said they applied because they were encouraged to do so by an educator.

As AiC Award recipients, they will join the nationwide AiC Community and have exclusive opportunities available to them as they pursue computing and technology in their academic and professional careers. AiC is a research-based program that provides long-term support to program participants, with 91% of past award recipients continuing to study STEM in college as a major or minor—77% of those in computing or engineering.

As their educator, once you endorse your student's application you become eligible for the [NCWIT Educator Award](#). Educators who are selected locally are awarded cash and access to professional development funds, as well as honored at the regional award ceremony alongside their students. For applications or information visit [NCWIT Aspirations in Computing](#) or contact [aspirations@ncwit.org](mailto:aspirations@ncwit.org).

**Additional resources:**  
[Aspirations in Computing \(AiC\)](#)  
[Regional affiliates](#)  
[Applications are now open](#)  
[NCWIT Educator Award](#)

## Know your CSTA+ Membership Benefits!

CSTA+ Members enjoy free access to over 20 courses from PluralSight. Develop your skills and prepare for computer science certification exams with courses in IT, operations, and more. Become a CSTA+ member now at <http://csta.plus>

CSTA Voice ISSN: 1555-2128 / CSTA Voice is a publication of the Computer Science Teachers Association. CSTA Voice is a quarterly publication for members of the Computer Science Teachers Association. It provides analysis and commentary on issues relating to K-12 computer science education, resources for educators, and information for members. The publication supports CSTA's mission to promote the teaching of computer science and other computing disciplines.

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

Reproduction Rights Information: No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording or information storage and retrieval system, without permission in writing from the publisher. Exception: permission to photocopy (individual) items for internal or personal use is hereby granted by CSTA.

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

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

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

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

# CSEd Week Driven with a Theme

BOBBY OOMMEN

CSEd week is coming soon. And for those of us in CS education, it's a week of excitement, planning, and opportunities to spread our love for CS to our schools and communities. Over the years, the number of CSEd Week resources has grown to include Hour of Code tutorials, posters, videos, lesson plans, and more. If this is your first CSEd Week, I encourage you to utilize these resources.

For those of you who have participated in CSEd Week for a few years and have been wondering how to take your CSEd Week to the next level, I'd encourage you to think about creating a theme for the week as a focus for the activities. For example, you might select a theme like cryptography, artificial intelligence, or algorithms, and then find or create activities for the week that focus on that theme. We have done this at my school for the past three years and have found it to be a model that deepens participants' knowledge and understanding, allows for greater cross-grade and cross-curricular collaboration, and helps our CS staff concentrate their efforts in planning and preparation.

My school had held traditional Hour of Code activities for two years with good success — kids participated, teachers were willing to give up a class or two to complete various tutorials, and we found speakers to talk with the students. The negative that we observed, however, was that there was a wide breadth of activities, but not much depth. Breadth of CS experiences was important when our school's CS program was just starting, but as our students have been exposed to more CS, we wanted to offer experiences to deepen their knowledge. Last year, our theme was artificial intelligence (AI). Students in grades Pre-K through 12 were able to learn about AI from outside speakers, in and outside of class activities, and other events. After learning about AI for a week, you can imagine how excited I was to hear a 5<sup>th</sup> grader, when asked by an outside speaker what she thought AI was, state succinctly, "It's getting a computer to do what humans do."

Secondly, we have found that creating a theme for CS Ed

Week encourages greater cross-grade and cross-curricular collaboration. Two years ago, our theme was cryptography. For a cross-grade activity, our first graders created [Caesar Cipher Wheels](#), learned how to use them, and then shared and decoded secret messages with their 8<sup>th</sup> grade buddies. During that same year, I partnered with the Latin teacher to do several activities using a scytale and Polybius square. I also partnered with our school librarian to create several videos to teach about various encryption techniques.

Lastly, creating a theme helped our CS department focus our efforts in planning and preparation. Instead of sifting through all of the CS learning resources available, we created a document in which we listed just the resources we found related to the topic. We then identified opportunities to involve students with the resources, such as school assemblies, homeroom, lunch breaks, or classes. During our focused search, we found excellent tools and resources that were not necessarily designed for CSEd Week, but fit our theme and goals.

Google's *Quick, Draw!* was well received by students in grades 1 through 12. *Quick, Draw!* asks students to draw sketches of several different objects, and then uses AI algorithms based on hundreds of thousands of submissions of similar objects to guess what was drawn. Using this tool, students got a glimpse into machine learning, and the need for quality data to help machines make valid decisions.

Our topic for this year will be "internet of things." Planning and preparation is underway and we are excited about what CSEd Week will look like in 2018. I'd encourage you to select a theme for CSEd Week to deepen student understanding, generate cross-grade and cross-curricular opportunities, and provide a focus and clarity necessary for an exceptional CSEd Week.

**Additional resources:**

[Quick, Draw!](#)

[Caesar Cipher Wheel](#)

**2019 CSTA Annual  
Conference  
Call for proposals  
now open**  
[cstaconference.org](http://cstaconference.org)

# Using Zork to Inspire the Future

MARTIN FISH

Anyone who works with middle school students will tell you that some of their greatest gifts are their enthusiasm, creativity, and exuberance. When they are excited about something, they will grab tight, think critically, and often produce results that are wildly beyond expectations.

Making a connection is not always easy. Manipulating a character on a screen might provide a surface-level connection. A series of discreet puzzles certainly helps to develop pattern-recognition and algorithmic thinking. But I wanted more. I wanted my students to see the power of coding as a creative platform through which they didn't just solve someone else's puzzles, but were instead creating something entirely new.

Time and again, one of the core truths that I rediscover in my teaching is that if I am excited about a concept, project, or activity then my students will also be excited. And for the project I want to tell you about, I decided to take a step back in time.

I am an unabashed child of the 80s. The Commodore 64 was my first computer. And while I played many games sitting in front of that big, blue screen, one game truly stood apart in my memory—Zork I: The Great Underground Empire.

Zork was originally created by a team of students at MIT in the late 1970s and it was dramatically different than the standard arcade games that were available at the time. The game was played entirely without graphics. There were no pictures—just words on the screen to read and responded to. The imagery and action took place in your imagination. Think of it as a “choose your own adventure” book, except on the computer.

And the control scheme? Communication. Want to climb the tree? Type: “Climb the Tree,” and the computer tells you what happens. Want to attack the troll? Type: “Attack the troll,” and the computer will respond, “What do you want to attack the troll with?” Want to mess around and see what it will say when you type “Eat the mailbox?” The computer has an answer for that, too . . . “I don't think the mailbox would agree with you.”

The amazing thing about Zork was that, for the first time that I can remember, I was having an interaction with a machine in which I felt like the computer was kind of listening. It was responding to me in an understandable way when I was communicating with it in plain English. And I wasn't the only one who got excited by Zork. This game was a true phenomenon of its time, becoming one of the top-selling home software games in the early 1980s.

A game like Zork is all about pathways, choices, and branching decision trees. Without needing to focus on the complexities of graphics or gameplay mechanics, I could talk to my students about the structure of the storytelling and the use of nested conditional logic to create adventure stories.

When I introduced this story-telling, game-development strategy in my classroom, the results were amazing. Students accepted the challenge, and many took it far beyond the base requirements of the assignment. Some spent hours developing and refining their stories. I think your students will connect with it also.

This activity is language-agnostic and can be easily adapted. For middle school, limit the story pathways to specific, predetermined options using button choices. More advanced students can create text-based interactions, much like the original Zork. CODE ON!

**Additional resources:**

[Zork introduction](#)

[Text adventure example](#)

[Student planning document](#)

[Text adventure rubric](#)

## NOVEMBER AUTHORS

### KENNEY CHAN / TEALS

Kenney is the TEALS Curriculum, Training, and Instruction Program Manager and has taught CS at Phillips Exeter Academy.

### MARTIN FISH / MASON, OHIO

Martin teaches the Coding and Robotics STEM Lab at Mason Middle School. He is a recipient of the 2018 CSTA/Infosys Foundation USA Award for Teaching Excellence Award.

### CRYSTAL FURMAN / COLLEGE BOARD

Crystal is a former high school AP CS teacher and current Director of AP Curriculum, Instruction, and Assessment for CS at the College Board.

### NATHANIEL GRANOR / TEALS

Nathaniel is the Lead Program Manager for TEALS East Region and has been involved with designing TEALS' Instructional Training program for the past five years.

### ADA IBE / CORNELL TECH

Ada is a Teacher-in-Residence with Cornell Tech. She's a former social studies teacher, turned CS teacher and coach who is passionate about CS education.

### MICHAEL JONES/ CS EDUCATOR

Michael is a UK qualified CS teacher in England. He is involved in developing assessment systems and teacher-trainee development.

### YVONNE MELTON / GOOGLE

Yvonne is a Program Manager on Google's CS Education team where she works with partners like CSTA to increase access to CS education.

### BOBBY OOMMEN / LATIN SCHOOL, CHICAGO, IL

Bobby is a middle school CS teacher. He is one of six 2018-19 Community Ambassadors for CS for All Teachers.

### SHEELA VANHOOSE / CODE.ORG

Sheela leads Code.org's policy work in multiple states along the east coast. Prior to joining Code.org, she served in similar capacities for Charter Schools USA and Broward County Public Schools.

# Attending AP Reading Impacts Teaching

CRYSTAL FURMAN

This year, *Advanced Placement Computer Science (AP CS)* had a record number of test takers, more than 135,000. Along with the increase in student interest comes a need for readers to score student exams. This past year, 323 *AP CSA* readers and 490 *AP CS Principles (AP CSP)* readers scored student exams.

The AP Reading is a long-standing tradition where high school and higher education faculty come together for a week of “reading” and scoring student papers. For those teachers who attend, they claim this is the best professional development opportunity they have ever had. This all-expenses-paid opportunity gives high school teachers insight into how their students will be scored, while providing them with a stipend for their time. Teachers can apply the processes they learn at the Reading in the classroom and become more efficient in scoring their own students’ work. But the benefits are not exclusive to high school teachers—higher education faculty often state that they had no idea they would glean so much from the opportunity to score AP exams. They get to see how students from across the country with different backgrounds are approaching these problems and the typical misunderstandings that students have in high school, as well as college.

In addition, a diverse community of CS educators—high school, higher education, private, public, rural, suburban—is built at the Reading. Opportunities to collaborate and learn new techniques are abundant, from the casual conversations that are had during breaks and lunch, to the planned activities. The evening activities often include guest speakers, opportunities to learn new programming languages or technology, chances to share best pedagogical practices, as well as purely social ventures like walking groups, shows, and baseball games.

Teachers often worry about whether they are prepared and ready to be a reader, but the support and training readers receive is superb. Readers spend nearly the entire first day in training, followed by reading previously scored papers to ensure they understand the rubric. There is a complete hierarchy of support for readers. For *AP CSA*, readers are paired with another reader and sit at tables of eight readers with a table leader. If a reader is unsure of how to score a paper, they consult with their partner. If their partner is also unsure of how to score the paper, the table leader is consulted. For each question, question leaders train readers and remain available to answer questions throughout the Reading. This hierarchical system provides support for readers, while ensuring student papers are scored consistently regardless of who is reading the work.

A similar system is used for *AP CSP*. The *AP CSP* Reading is a hybrid Distributed Reading with some readers onsite and others reading distributed from home. Because of the distributed nature of the *AP CSP* Reading, readers do not have a partner, but they do have a table leader they can chat with or call when they have questions. And each performance task has question leaders that assist with training and support the table leaders throughout the Reading.

For many, spending eight days away from home would be impossible, but the *AP CSP* Distributed Reading allows readers the flexibility to participate in the Reading part- or full-time from home. For these readers, it means gaining the professional development experience that comes from reading student work, while maintaining their home life commitments.

For more information and to apply, visit [AP Central](#).

## Support your local CSTA Chapter with your CSTA+ Membership!

The new CSTA+ membership tier allows us to deliver a much richer level of support and value, without impacting the support we offer our standard, nonpaid members. It will also help us grow and sustain our local chapters, annual events, and central organization.

Best of all, 50% of CSTA+ dues will go directly to local chapter programs.

**Become a CSTA+ member today!**

# Culturally Responsive CS Teaching

ADA IBE, NATHANIEL GRANOR, KENNEY CHAN

Each summer TEALS prepares hundreds of software engineers to enter high school classrooms as volunteers to support classroom teachers who are learning computer science (CS). TEALS is a Microsoft Philanthropies program that helps high schools build and grow sustainable CS programs. Our summer training has focused on general and CS-specific pedagogy and classroom management, including topics like engaging students with questioning, using worked examples, and how to structure a productive lab session.

Our volunteers serve in a diverse array of urban, suburban, and rural schools across 29 states and DC. TEALS Volunteers often support classroom teachers where many students come from different cultural backgrounds than their own and have frequently asked us for training to help bridge that divide.

Inspired by our partners at NYC Department of Education's CS4All Initiative, we've added a new focus on Culturally Responsive Teaching (CRT) to our summer training. We used Zaretta Hammond's *Culturally Responsive Teaching & The Brain* as the foundation for this new training strand, interpreting her ideas for high school CS context. Hammond writes that students of color start with a small achievement gap that grows as they progress through school "because we don't teach them to be independent learners." She defines CRT as a toolset for educators to "recognize students' cultural displays of learning and meaning-making, and respond positively and constructively with teaching moves that use cultural knowledge... to promote information processing... and having a social-emotional connection to the student in order to create a safe space for learning."

Hammond presents the *Ready for Rigor Framework* as a system for educators to engage in CRT, which includes four key areas:

- Having **awareness** of the deep culture, learning behaviors, and societal context of both the students and themselves.
- Constructing **learning partnerships** with each student that establish trust.
- Making use of cultural knowledge and brain science to optimize students' **information processing**.
- Building a safe and productive **community of learners and learning environment**.

In the remainder of this article, we'll explore a few of the ideas from Hammond's book we found relevant for high school CS classrooms and how we interpreted them for our context.

## Wise Feedback

Hammond points out research by Cohen and Steele (2002) finding students of color often didn't receive timely, actionable feedback, either because the teacher didn't want to hurt the student's feelings or didn't want to be perceived as prejudiced. With Wise Feedback, instructors must set and reinforce high expectations, reiterating that mistakes signal a task's high demands, not a student's low capability. Teachers can say things like, "This assignment is very challenging but as computer scientists, we know how to iterate until we reach a good solution." Providing personal assurance is the next part of Wise Feedback, emphasizing that students are capable and will improve with effort ("I know this might seem impossible right now, but you'll get better with practice"). As the class progresses, teachers can also reference prior successes. Lastly, teachers should provide actionable next steps to ensure they have a path forward.

TEALS equips teachers and volunteers to use the Socratic method to guide students towards solutions when they encounter problems. We recommend instructors use Wise Feedback when working with individuals during lab and give an actionable next step before leaving each student. For instance, instead of "Can you think of a previous assignment where we did something similar?" say, "Look at your solution to lab 1.3 for an example of a similar pattern."

## Relevant Examples

Getting to know your students is the first step to building a learning partnership with each student. If you don't already, distribute a "getting to know you" questionnaire. Use questionnaire answers, and information you learn from student work or conversations, to frame your lesson examples. For instance, in a Java textbook, some examples ask students

to write a program to compute compound interest which is not relatable for some students. We recommend swapping in examples based on cultural references and interests of students in your classroom. Instead of compound interest, students could compute the number of “likes” a social media post receives as it’s reshared through a network.

### Representing the Tech Industry

Show your students examples of computer scientists who come from similar cultural backgrounds, so they know people who look like them are in the industry. Look for stories about applications of CS in fields connected to students’ interests, such as medicine, climate science, music, fashion, video games, and sports. You can also showcase people moving the tech industry forward through non-coding roles like designers, product managers, and researchers.

### The Physical Classroom

Your classroom appearance may impact students’ interest in taking, or staying in, your class. Girls were almost three

times more likely to say they’d be interested in enrolling in a CS course if the classroom looked like a non-stereotypical one, according to a [University of Washington report](#) on how stereotypes undermine girls’ interest and sense of belonging. Various organizations offer free or low-cost sets CS posters showcasing diverse representation in computing, for instance: [NCWIT](#), [Code.org](#), [New Relic](#), [IAAMCS](#), and [energy.gov](#). You can also create, or have students research and create, posters.

These are just a handful of examples of meaningful changes we can make in our CS classrooms to ensure all students can succeed in CS. We’re excited to continue to find ways to ensure that our CS classes are culturally relevant. Please share your ideas with us to continue the conversation.

### Additional resources:

[University of Washington report](#)

[NCWIT](#)

[Code.org](#)

[New Relic](#)

[IAAMCS](#)

## Know your CSTA+ Membership Benefits!

CSTA+ members will have access to early registration for the conference with special rates. Become a CSTA+ member now at <http://csta.plus> to ensure you’re the first to know.

### Keep up with CSTA!

#### Twitter

[csteachersorg](#)

#### Facebook

[Computer Science Teachers Association](#)

#### LinkedIn

[search computer science teachers association](#)

### BOARD OFFICERS

Fred Martin, Board Chair  
[fred.martin@csteachers.org](mailto:fred.martin@csteachers.org)

Jennifer Rosato, Board Chair  
Elect  
[jrosato@css.edu](mailto:jrosato@css.edu)

### STAFF

Jake Baskin, Exec. Director  
[jake.baskin@csteachers.org](mailto:jake.baskin@csteachers.org)

Michelle Page, COO  
[michelle.page@csteachers.org](mailto:michelle.page@csteachers.org)

Pat Phillips, Editor  
[publications@csteachers.org](mailto:publications@csteachers.org)

# Where Have All the Teachers Gone?

MICHAEL JONES

*Editor's note: Computer science (CS) teacher preparation in England can be delivered in several ways. An understanding of those programs is helpful for readers unfamiliar with the UK system, so we've provided a description of them here.*

*Postgraduate Certificate in Education (PGCE) When individuals successfully complete the teacher preparation program, they are awarded the status of Qualified Teacher Status (QTS). This is the license to teach. Some trainee teachers opt to acquire the next stage, PGCE. This is equivalent to 1/3 of a master's degree. The PGCE is awarded on top of the QTS in recognition of further academic study in education and pedagogy.*

*School Direct Training (SDT) Several years ago, it was recognized that the standard route to becoming a qualified teacher was predominantly through a university delivered program of study. This skewed teacher recruitment towards recent graduates and a lower age band, typically 30 years old and younger. In recognition that there was typically an older cohort considering a career change to teaching, the SDT program offered a reduced university environment and an in-school experience spent mostly in one school. Teacher trainees in this route are responsible for their fees (typically \$7,000 for QTS plus \$5,000 for the PGCE element, if required). Trainees can apply to a government bursary (subsidy) which ranges from \$12,000 to \$35,000.*

*School Direct Salaried (SDS) SDS follows the same training pattern as SDT but offers a salary to trainees. The salary ranges from \$22,000 to \$27,000. The school pays the trainee a salary and tuition fees. In return, the school can claim funding for each trainee. Schools favor this route because they have teaching assistants and technicians, many with substantial experience in the school.*

Prompted by seeing the closure of an initial teacher training program (ITTP) in the UK to prepare secondary CS teachers, the struggles to recruit students into another program, lack of applicants to school-based training, and an innate skepticism of government released figures, I delved into the world of university student recruitment. I requested information on recruitment to teacher training programs for CS from 13 college and university institutions. Of the 13 institutions approached, 10 responded well within the 20-day window provided by the *Freedom of Information Act*.

The three paths to becoming a teacher in England attract different cohorts of teachers. On first glance, this appears a good idea. However, scratch below the surface and concerns become apparent. Choosing one route over another tends to reduce the intake in all. That is not such a bad event in an "all things being equal" universe. Unfortunately, the universe has laws of unintended consequence. Lower overall enrollments mean that programs close. Reduce the intake across all routes and you see more programs close. For example, assume total intake is 1000 individuals with each program requiring a minimum of 500 to run and you have lack of enrollees across the board. In practical terms, this means that courses close and people lose jobs with universities reluctant to reinstate the position later. Having been involved in the creation of two programs at two different institutions, one of which has closed and the other with a significantly reduced budget and loss of staff, the effect is very real. Colleagues across the country report that programs have closed or are on very shaky foundations.

The numbers across reporting institutions tell the story. The 2013-2016 CS teacher trainee recruitment target was 1282; the actual number was 643. In percentage terms, this means that we have failed to recruit 50% of required teacher trainees over four years. Figures for the last academic year indicate that 66% of the recruitment goals was met. Specific programs:

- **PGCE:** target 869 versus actual of 550 (63% met).
- **School Direct Salaried (SDS)** target 112 versus actual of 31 (28% met).
- **School Direct Training (SDT)** target 301 versus actual of 62 (21% met).

It is clear that PGCE remains as the most popular route into teacher training. The cohort tends towards the younger, new graduates. Armed with a generous bursary this can appeal to many. However, the sting in the tail is that PGCE programs are heavily reliant on meeting teacher trainee recruitment targets. Universities rely upon the income from fees.

School Direct programs, both Salaried or Training, show poor recruitment. This has a double effect for universities. While they are involved in providing aspects of instructions, they attract much lower fees, approaching 50%. To get a viable return requires that more are recruited into these high school-based programs. The figures show that this is not happening.

What does this picture tell the other countries? The short answer is that teacher recruitment to CS is tough and getting tougher. Making it economically appealing helps. However, even with bursaries worth in the region of \$35,000 that does not compete well with fresh graduates' salaries tending towards \$50,000.

There is some scope in targeting career changers. This, however, brings with it a set of other challenges. Partly monetary—moving from a fairly high salary to trainee teacher pay is a hard pill to swallow, but it does hold the prospect of rapid salary increase as early responsibility is given to new teachers in subjects with teacher shortages.

By far the greatest issue facing any institution, be in the UK or other countries, is working with and developing talent of a typically older and experienced cohort. They have often held managerial positions and find themselves in a junior position for at least the first two years of entry into teaching. This is particularly true of males, 35 or older. These are identified as the group least likely to succeed in a teacher training environment. Younger graduates are used to the rigor and general environment of a training position. It is a good idea to mandate that CS is taught to all students, as it is in England. However, not having the staff to deliver this mandate is an issue.

### Effect on Training Providers

Going back to the effects of low teacher trainee recruitment, there is a degree of uncertainty for continued employment of university staff. Highly skilled professionals do not perform well under a regime where programs are likely to be pulled. At the extreme, universities employ staff on a zero-hours approach (if there are no students for a class on a particular day you do not get paid), the coordination of retaining staff becomes more difficult.

So, what of the newer teacher apprenticeship scheme? Fundamentals are scant. However, what is clear is that this is a replacement for SDS. Given the very short history of SDS, the apprenticeship approach does not bode well. It is likely that the total pot of targeted places will be split between PGCE and apprenticeships. The administration and extra support schools will need to provide will most likely increase. The change of name and slight change of process will increase the pressures on universities as the major degree awarding bodies. Further cuts to budgets will emphasize the current staffing issues.

Contact Michael Jones at [michaelpjones@live.co.uk](mailto:michaelpjones@live.co.uk).

### Additional resource:

[Computing At School events](#)

**Aspirations in Computing  
Nominations due  
November 5, 2018**

## Computer Science Recruitment Toolkit Available

While computer science (CS) classes are expanding, too many students, especially girls, Black, Latino, and Native American youth, feel like it's not for them. All they need is a desire to solve problems and a willingness to explore new ideas.

We know that by 2026, there will be an estimated 3.5 million computing-related jobs open in the US. People who study computers will shape the world—and should be as diverse as the world they're shaping. The next generation's innovators are in your school right now!

The resources in the Microsoft CS Recruitment Toolkit will help support teachers and schools to bring more diversity into their middle and high school CS classes. There are even sample copy and images you can post on your organization's social media channels. [\*\*Get the kit now!\*\*](#)

# Governors for Computer Science

SHEELA VANHOOSE

In less than a month, 36 states will hold gubernatorial elections. These races will have a significant effect on the future of computer science (CS) education policy. We are watching as gubernatorial candidates tout their education platform on the campaign trail and then in office as they set out to keep those promises. From state to state, their powers vary, but these chief executives typically have influence over state budgets, policy setting, and appointments of state board members and/or agency heads.

When it comes to CS education policy, there is an opportunity for Governors to lead. In 2016, the Governors' Partnership for K–12 CS launched. This is a bi-partisan initiative led by Governors Asa Hutchinson (R-AR) and Jay Inslee (D-WA). The goal of the partnership is to promote CS education by working towards three key policies in their respective states:

- Enable all high schools to offer at least one rigorous CS course.
- Fund professional learning opportunities so teachers can be prepared to teach these courses.
- Create a set of high-quality academic K–12 CS standards.

Today, that partnership has grown to 18 governors working towards those same goals. You can learn more about the partnership and view a list of current governors at [Govs for CS](#). The reality is that this fall there will be an opportunity to invite incoming governors to make a commitment to expand access to CS. Here is how you can help.

After election day, reach out to your Governor-elect and ask them to join the Governors' Partnership for CS. You can do this by sending an email or letter to their office, or maybe you have a unique opportunity during a school visit to

mention this in person. What you say is far more important. You should share how their support can help expand access across the state. I always like to use talking points that highlight the economy and the equity issue.

Visit [Promote Computer Science](#) and click on your state for data points on the number of open computing jobs and CS graduates in your state. You can also find an overview of which CS-related policies your state has successfully put into place and which policies your state has an opportunity to improve. The goal should be to expand access to CS. The access problem affects every state and every industry. Twice as many Americans use computing in their jobs than previously reported, and half of those are in non-STEM fields. Your new Governor might be surprised to learn that 91% of open software jobs are outside Silicon Valley. And computing occupations are the #1 source of new wages in America.

The data is powerful, but your personal story is far more compelling. I always recommend that teachers and students share their story. Here is one great example: A Detroit teacher began teaching AP CS Principles and in his inaugural class there were 35 students, 100% of whom were underrepresented minorities. His class alone marked a 225% increase in the number of Black students passing the exam from 2017 to 2018 in the entire state of Michigan. That is a story worth sharing. How has CS affected your students? Your classroom? Or you as an educator?

#### Additional resources:

[Governors for K–12 Computer Science](#)

[Code.org: Promote Computer Science](#)

**Save the Date!**  
**2019 CSTA Annual Conference**  
July 7–10, 2019  
Phoenix, Arizona

# What's Coming Up

Find more upcoming CS events on the CSTA [website](#).

List your CSTA event by contacting: [customerservice@csteachers.org](mailto:customerservice@csteachers.org)



## Computing at School (CAS)

Upcoming events in the UK

## CCSC (Southeastern)

November 2–3, 2018  
Salem, VA

## Aspirations in Computing

November 5, 2018  
Nominations due

## DFW CSTA Collaborative Conference Proposals

November 30, 2018  
Proposals due

## 2019 CSTA Conference Proposals

December 2, 2018  
Proposals due

## CSEdWeek

December 3–9, 2018  
In your community

## DFW CSTA Collaborative Conference

February 23, 2019  
Location TBD

## AP Computer Science A Reading

June 2–8, 2019  
Kansas City, MO

## AP Computer Science Principles Reading

June 11–17, 2019  
Kansas City, MO

## 2019 CSTA Annual Conference

July 7–10, 2019  
Phoenix, AZ

## College Board Workshops and Institutes

Various dates and times

## College Board Webex Meetings

Various dates and times

**Enjoy the benefits  
Become a  
CSTA+ member now!**

**2019 CSTA Annual  
Conference  
Call for proposals  
now open  
[cstaconference.org](http://cstaconference.org)**