csta Voice

Professional Development

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

Volume 9, Issue 2 May 2013

Inside This Issue

IN THE NEXT ISSUE OF THE *VOICE*

Gearing Up for Fall

Chapters Host Oracle Academies

Lissa Clayborn

CSTA AND ORACLE have joined forces on a new project that will provide much-needed professional development opportunities. Through this new program, CSTA chapters throughout the U.S. and Canada will hold 32 Oracle Academies in 2013.

Responses from CSTA chapter leaders have been overwhelmingly positive. "Our chapter members are very excited about this opportunity," reported CSTA Greater Boston Chapter leader, Kelly Powers. "Members are constantly looking for opportunities to keep their knowledge up to date. Having this content taught by Oracle experts, and also having the chance to pursue certification, is immensely exciting for us. Educators will grow both professionally and as a community of teachers."

Working together, Oracle and 18 chapters will train up to 600 teachers in Oracle Academy's *Introduction to Computer Science* curriculum. "Oracle works to advance education with state-of-the-art technology programs that awaken and deepen students' interest in computer science (CS) and engineering," said Alison Derbenwick Miller, Vice President, Oracle Academy. "We are pleased to work with CSTA to help promote and strengthen CS education and train teachers across the U.S. and Canada on the Oracle Academy curriculum. These efforts will ultimately help expand the reach and depth of CS education at the K–12 level and encourage students to explore the power of computer programming and the opportunities a CS education makes possible."

The workshops are scheduled from February through August, 2013, and will offer a variety of academies including courses on Alice, Java Fundamentals, Java Programming, and Database Design and Programming. Curriculum Descriptions are available at: *academy.oracle.com/oa-web-introcs-curriculum.html*. To view the list of workshops planned for your area, visit the events calendar at: *academy.oracle.com/pls/html/f?p=31000:80:22472170529226*

You're Invited!

CSTA Annual Conference

FOR 13 YEARS CSTA'S Computer Science & Information Technology (CS&IT) conference has been the premier professional development event for computer science teachers. But this year, the event moves to a whole new level with a new name and many more opportunities for teachers to learn, engage, and network.

The newly-named **2013 CSTA Annual Conference** is scheduled for July 15–16, at the Boston Marriott Quincy, in Quincy, MA. The conference has been expanded to include more hands-on workshops (10), more pre- and post-conference events (in-conjunction leadership and advocacy workshops), more chances to chat with *continued on page 2*

CSTA THANKS

ACM and SIGCSE for their generous support of the 2013 CSTA Equity Workshop.

Deborah Seehorn, Mindy Hart, Stephanie Hoeppner, and Tammy Pirmann for assisting with the CSTA booth at SIGCSE 2013.

YOU'RE INVITED!

continued from page 1

sponsors (sponsor tables), and more fun (an evening event at Microsoft's New England Research and Development Center and the great secret-shopper raffle).

The Conference Planning Committee has also put together an exciting agenda of 20 breakout sessions and two keynotes (stay tuned for keynote speaker announcements via the CSTA announcements listserv). According to Planning Committee Chair David Reed, this was the most competitive year ever for conference proposals. "The workshops and sessions are selected on the basis of a proposal submission and volunteer review process. This year we received 98 proposals, each of which was evaluated by several reviewers. This allowed us to build an incredibly strong and varied agenda."

According to Reed, one session will also be making a much-requested return from 2012. "The Mobile Programming Throwdown was one of last year's most popular sessions. It will be back this year with new challengers and a new app challenge," says Reed.

Conference and workshop registration includes lunch. A discount is available when purchasing both a morning and an afternoon workshop. Be sure to register for the conference and sign up for hotel accommodations soon. Conference registration closes June 16; housing closes June 10. Registration is now open and you can find all the details (including the tentative agenda, housing, and transportation) on the CSTA website at: *www.cstaconference.org*.

The 2013 CSTA Annual Conference is generously sponsored by the Anita Borg Institute for Women and Technology, Microsoft, Microsoft Research, and Oracle Academy.

See you at the 2013 CSTA Annual Conference!

Finch, Hummingbird, and Snap!

CSTA Conference Workshop Preview

Tom Lauwers

PLEASE JOIN ME in the *Finch, Hummingbird, and Snap!* workshop at the CSTA Annual Conference. Experience the drag-and-drop language Snap! with the Finch Robot and the Hummingbird Robotics Kit. Discover how drag-and-drop programming languages and robotics present opportunities to introduce computing concepts to students in elementary and middle school in diverse subjects.

Snap! is a spin-off of the popular drag-and-drop environment of Scratch. It was developed by Jens Mönig and Brian Harvey and is housed at *snap.berkeley.edu*. Snap! runs in a browser, is free, and doesn't require a physical robot. A third-party library and helper program enables Finch and Hummingbird robots to be programmed in Snap!

The Finch is a robot designed for introductory computer science (CS) education. It is the result of the four-year CSbots study at Carnegie Mellon's CREATE lab. The Finch enables students to write richly interactive

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 e-mail at *cstahelp@csta.acm.org*, or call 1-800-342-6626 (U.S. & Canada) or +1-212-626-0500 (Global).

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

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

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

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

programs using a suite of sensory inputs and light, sound, and motion outputs. In addition to Snap!, the Finch is programmable in more than a dozen programming languages.

The Hummingbird Robotics Kit is comprised of a controller, LEDs, sensors, and motors. It supports engineering and robotics activities for ages 10 and older that involve building robots, kinetic sculptures, and animatronics.

The broad aim of the CSTA conference workshop is to introduce attendees to the possibilities of combining a drag-and-drop environment with electronics that can "push on" and "sense from" the real world. They will learn the basics of the Snap! programming environment, as well as the blocks specific to controlling Finch and Hummingbird, and have plenty of time to program the robots. By the end of the workshop, attendees will connect Finch and Hummingbird sensors to create a Pong-like ball and paddle game.

In the spirit of full disclosure, I was a researcher on the teams that developed the Finch robot and Hummingbird kit and now sell them through my company, BirdBrain Technologies LLC.

LEARN MORE:
Snap! *snap.berkeley.edu*
Finch Robot *www.finchrobot.com*
Hummingbird Robotics *www.hummingbirdkit.com*

Programming with Greenfoot

CSTA Conference Workshop Preview

Neil Brown

PLEASE JOIN ME in the *Introduction to Programming with Greenfoot* workshop at the CSTA Annual Conference. Two members of the Greenfoot team will guide you in building your first Greenfoot game—no prior programming knowledge necessary! Come learn how Greenfoot was designed to enhance teaching and learning in the computer science (CS) classroom.

Historically, the primary curriculum choice available to CS teachers was to decide which language to teach: Pascal or BASIC, Python or Java. However, in recent years, teachers have had an additional decision to make—the choice of a programming environment. There can be a noticeable difference in what it is like to learn programming in different frameworks, even when they are based on the same programming language. Greenfoot is an environment for the Java programming language but is a very different experience from learning Java in a command-line environment. Greenfoot is a programming environment designed for beginners.

With Greenfoot, students can create 2-D graphical games and simulations, much like what can be created in Scratch or Alice. However, Greenfoot code is written in standard text-based Java. Greenfoot is a natural next step after Scratch and Alice and can form a bridge from those early beginner environments toward professional-style, full-text programming.

Although Greenfoot uses Java as its programming language, students are not required to start by writing any "public static void main(String[] args)" code that they do not yet understand (a classic complaint with Java). Instead, the framework simply requires students to fill in the act() method of a provided class.

All of the graphical code is included in Greenfoot, so students can focus on the behavior of the actors in the scenario. The first line of code students write is "move(5);". Just this one line will send a character scuttling across the screen. The second line is "turn(10);" which, combined with the first line, will move the character in a circle. As teachers who use Scratch and similar environments know, it is very motivating to see results so quickly and it makes debugging easier when mistakes are animated on the screen.

We have created several extensions to Greenfoot to increase student engagement. An interface to the Microsoft Kinect device enables students to interact with a program using gestures. An interface to the ScratchBoard, a small circuit board with various *continued on page 4*

csta
Computer Science Teachers Association

Let us know if your contact information changes.

cstephenson@csta.acm.org

Contribute to the CSTA *Voice*

The editorial board of the **CSTA *Voice*** is dedicated to ensuring that this publication reflects the interests, needs, and talents of the **CSTA** membership. Please consider sharing your expertise and love for computer science education by contributing newsletter content.

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

Volunteer today!

The CSTA *Voice* welcomes your comments.

E-MAIL: *cstapubs@csta.acm.org*
PHONE: 1-608-436-3050
FAX: 1-928-855-4258

Letters to the Editor are limited to 200 words and may be edited for clarification.

ACM founded CSTA as part of its commitment to K–12 computer science education.

PROGRAMMING WITH GREENFOOT
continued from page 3

sensors, makes it possible to create a wide variety of programs.

Greenfoot is free, open source, and runs on Windows, Mac OS X, and Linux operating systems. We generally recommend it for ages 14–18, but many teachers have reported using it with younger children. You can download Greenfoot and access a variety of teaching resources at *www.greenfoot.org*.

CSTA K–12 CS Standards: Moving Forward

Deborah Seehorn

IN OUR CURRENT ASSESSMENT-DRIVEN ENVIRONMENT, computer science (CS) teachers are under increasing pressure to demonstrate that their CS courses align with recognized standards. To help in this effort, CSTA has created a number of resources that demonstrate that CS is indeed a discipline worthy of national attention.

The *CSTA K–12 Computer Science Standards* (revised and published in December 2011), provide a common reference point for CS curricula and professional development. These standards are organized into five strands: Collaboration; Computational Thinking; Computing Practice and Programming; Computers and Communications Devices; and Community, Global, and Ethical Impacts. The CS standards for learning are progressively scaffolded through these five levels: Level 1:3 (Grades 1-3); Level 1:6 (Grades 3-6); Level 2 (grades 6-9); Level 3A (grades 9 and 10); and Level 3B (grades 11 and 12). There are excellent, descriptive graphics in the standards document that depict this scaffolding. The complete standards are available on the CSTA website for downloading at: *csta.acm.org/Curriculum/sub/K12Standards.html*.

We have also created new resources to help you demonstrate how your CS courses contribute to the teaching of other national standards. The *CSTA K–12 Computer Science Standards* have now been correlated with the Common Core State Standards, the STEM Cluster Topics, and the Partnership for 21st Century Essential Skills. Debbie Carter, a former CSTA Board Member, painstakingly compared the *CSTA K–12 Computer Science Standards* to each of the other three sets of national standards. The CSTA Curriculum Committee reviewed Debbie's meticulous crosswalks and collaborated on the final documents.

These "crosswalk" documents will be exceedingly helpful to classroom teachers who are asked to demonstrate how the content they teach addresses and reinforces national standards. They are now available on the Curriculum page of the CSTA website at: *csta.acm.org/Curriculum/sub/K12Standards.html*. Be the first in your school or district to check out these useful crosswalk documents and put them to good use. Then spread the word!

If you are looking for help determining how to implement the *CSTA K–12 Computer Science Standards* in your CS classroom, or how to use the "crosswalk" documents, the best place to start is with your local CSTA chapter. Several chapters have led or are planning workshops to assist CS educators in using the standards to crosswalk to their existing CS curricula and to develop new curricula. You can find the contact information for all of the CSTA chapters at: *csta.acm.org/About/sub/CSTAChapters.html*

How will you use the standards today?

Congratulations CSTA Members

Bryan Baker, Doug Bergman, Vicki Coffman, Creighton Edington, Mike Efram, Holly Erickson, Steve Fulton, Rebeca Gonzalez, Amanda Goranson, Michael Hanus, Matt Harbinger, Shannon Houtrouw, Keith Jackson, Dean Johnson, Sharon Jones, Tammy Neuhaus, Emmanuel Onyeador, Bethany Petr, Beth Richtsmeier, Jacob Stephens, Adam Swift, and Carol Yarbrough have been awarded a 2013 National Center for Women & Information Technology (NCWIT) Aspirations in Computing Educator Award for their work in encouraging young women in computing.

ww2.ncwit.org/award/award.index.php

Authentic Learning and Computer Science

Responding to Tragedy

Patrice Gans

Editor's note: *This is part two of a two-part series on authentic learning in the computer science (CS) classroom. Part two focuses on activities most appropriate for students in grades 6–12.*

The best classrooms strive to create an environment that encourages independent thought, collaboration, critical thinking, and creativity—objectives in the Framework for 21st Century Learning. I find that authentic learning enables me to put these goals into action.

Authentic learning can take many forms. As an alternative to the current trend in "gamification" (*www.gamification.org*), it can combine technology and social causes into an ideal experience for engaging students. Although most students experience technology as users of social media and games, incorporating social good into the CS curriculum shows students that, as technology creators, they have power beyond creating for media and entertainment.

Authentic learning can take many forms.

In the wake of tragic events in the town I teach in, Newtown, CT, I was looking for ways to empower my students while sparking their interest in technology. With these goals in mind, I partnered with Random Hacks of Kindness to create the first ever Random Hacks of Kindness Junior event (*www.rhok.org/event/rhok-junior*). This collaborative effort brings together technology experts (university professors, college students, high school teachers, secondary students) with middle school students to create smartphone apps, websites, and educational games to help local causes. If registration is any indication, it will be a popular way to empower students and fuel their enthusiasm for computing and potential careers in CS.

Elementary and middle school students are not the only ones excited by engaging in authentic learning projects. High school students are equally enthralled with creating projects that have practical hands-on applications. One approach is the after-school program in mobile computing called "I'll Write that App for You" (*explore.appinventor.mit.edu/stories/ill-write-app-you*). The program is currently operated by Pauline Lake, a student at Trinity College. The lessons use App Inventor as an instructional tool to teach high school students basic CS principles and to introduce them to computational thinking. Pauline explains, "The program plays on the 'cool' factor of working with Android smartphones to spark student interest."

During the program's inaugural season in 2010, 10 ninth and tenth grade students learned App Inventor basics. The original goal of the program was for students to create educational apps. Since its inception, the program has grown from its original four weeks to a six-week, eighteen-session class that brings together students from a variety of high schools in the Greater Hartford area.

The material for the course and additional resources for using App Inventor in the classroom can be found online at MIT's Center for Mobile Learning (*teach.appinventor.mit.edu/resources*).

Educators across the country are turning to authentic learning practices and putting the focus back on the learner in an effort to improve the way students learn, retain, and transfer knowledge. Personal experience has shown that authentic learning can be an ideal vehicle for student success. Linking learning experiences to real-world opportunities that fire students' passions almost guarantees engagement, enthusiasm, and achievement.

LEARN MORE:

Authentic Learning Initiative
net.educause.edu/ir/library/pdf/ELI3009.pdf
Games for Change *www.gamesforchange.org*
Computing 4 Good (C4G)
www.cc.gatech.edu/about/advancing/c4g
Socially Relevant Computing (SRC)
src.cse.buffalo.edu/index.html

Meet the Authors

Lissa Clayborn
CSTA
Lissa is the CSTA Director of Development and has worked for over 15 years in the nonprofit educational technology sector.

Neil Brown
University of Kent, UK
Neil is a computing education researcher. He is part of the BlueJ and Greenfoot teams and is responsible for professional development.

Ria Galanos
Thomas Jefferson High School, VA
Ria teaches CS and is passionate about finding new ways to introduce young people to computing. She has led a variety of teacher workshops.

Patrice Gans
Newtown, CT
Patrice is a K–8 technology teacher at Fraser-Woods School. She also teaches Scratch in summer programs at Naugatuck Valley Community College.

Jeff Gray
University of Alabama
Jeff is an Associate Professor of CS and has offered summer camps, competitions, and teacher workshops. He was a pilot instructor for the CS Principles course 2011-2013.

Michael Jones
CS Educator, UK
Michael has worked in a variety of English schools. He develops assessment systems, guides trainee teachers, and consults on curriculum development.

Tom Lauwers
BirdBrain Technologies LLC
Tom created the Finch robot and Hummingbird Robotics Kit. He received his Ph.D. from the Carnegie Mellon Robotics Institute.

Deborah Seehorn
NC Department of Public Instruction
Deborah is a Business, Finance and Information Technology Education Consultant. She served as the Chair of the CSTA Standards Task Force in 2011 and is the CSTA State Department Representative, Chair Elect, and Curriculum Committee Chair.

CS Principles Update

A Statewide Model for Deployment of CS Principles

Jeff Gray

The University of Alabama is launching a new state-wide professional development project that will prepare 50 new Computer Science (CS) Principles instructors and 1,500 students over three years. This National Science Foundation-funded project will provide year-long professional development (PD) among cohorts of peer teachers who will collaborate on shared experiences, both in-person and through an existing distance-learning network that crosses geographic and socio-economic boundaries.

The project is based on the belief that AP CS courses must be approached much differently than other STEM areas because most STEM teachers lack the background preparation knowledge. The typical approach to AP training in other STEM disciplines involves week-long AP Summer Institutes (APSIs) where teachers in particular subjects are provided training to update their content knowledge and learn about new pedagogical approaches. The assumption of all current APSI offerings is that the attendees already possess most of the desired content knowledge. This is not true for CS and we believe that the CS 10K vision requires new models of teacher PD for investigation and refinement. Applying the current AP training model for new CS teachers is similar to asking a teacher with no mathematics background to initiate a new AP Calculus course with just one week of training. This situation would seem absurd to most administrators in the mathematics context, but it is the common expectation for promoting new AP CS courses. A different model is needed as the progression toward a CS Principles AP exam evolves (*www.csprinciples.org*).

In addition to the teacher preparation concern, this project is also motivated by the observation that most CS education efforts are successful in certain geographic regions for a specific time period, but often lack a unified effort that can scale and sustain the momentum. Although excellent environments (e.g., Alice, App Inventor, and Scratch/Snap!) and curricula (e.g., CS Unplugged, Exploring CS, and Media Computation) exist, a model that grows and maintains a unified effort like CS Principles is currently not in widespread adoption. In collaboration with Haynie Research and Evaluation, our project includes deep assessment to uncover core practices that can be recommended for scaling and sustaining CS education initiatives like CS Principles.

In this project, we are using the Advanced Placement Training and Incentive Program (APTIP) model pioneered by National Math and Science Initiative (NMSI). Since adopting APTIP over the past 5 years in other STEM areas, Alabama's percent increase in qualifying scores on AP exams ranks Alabama #1 among all 50 states in: 1) qualifying scores (3 or above) on AP STEM exams, and 2) underrepresented minority qualifying scores on AP STEM exams. Some of the specific NMSI practices that we are integrating into our project include:

- Master Teacher mentoring and vertical teaming—A statewide "Teacher Leader" model is being pursued where those already teaching more rigorous CS courses (e.g., the existing AP CS exam) will assist in training future peer cohorts who initiate CS Principles in their schools;
- Open enrollment to increase diversity of participation;
- Rigorous course content for professional development—year-long professional development to support teachers, rather than week-long summer institutes that are typical of most AP training programs in other STEM areas;
- Student mentoring through weekend skills development;
- Incentives for teachers and students who participate in the project.

Our project is also pursuing the following ideas:

- Training pre-service teachers by integrating secondary Math/Science Education students from the University of Alabama into the project;
- Exploring certification and other administrative policy issues to extend the eligible pool of teachers available for participation;
- Aligning the course to statewide standards—Alabama already has an official "CS Principles" course on the state approved list of courses, and a CS Principles course is taught each Fall at the University of Alabama (*cs104.cs.ua.edu*).

The Teacher Leader concept borrows from the Master Teacher and Vertical Teaming approach of NMSI. We believe that this idea is particularly needed in CS, due to the large gap in the skills between those already teaching some form of rigorous CS, and those who are focused on general literacy. The Teacher Leaders have the capability to share their deep experience in a peer-oriented arrangement. In the first quarter of our project, we are engaging 12 Teacher Leaders to begin their work with us. In years two and three of our project, we will work with 40 teachers from other STEM areas. A Google-supported CS4HS for Summer 2013 will help us acquaint the other STEM teachers with our project and the CS Principles Big Ideas as they prepare for participation in the future.

If successful, this project would demonstrate how one of the nation's lowest participating states in the current Java-based AP CS exam can emerge with strong participation in the future CS Principles exam.

LEARN MORE:
csprinciples.cs.ua.edu (open May 6)

Chapter News

Exploring Computer Science Initiative in Utah

Editor's note: *In this interview, Helen Hu, President of the CSTA Utah Chapter, describes a new three-year Computer Science 10K grant, funded by the National Science Foundation's Computing Education for the 21st Century program (NSF-CE21 #1240977).*

CSTA: Congratulations on receiving this prestigious grant. What are your plans?
Hu: Our award runs for three years from January 2013 to December 2015. During this time we will prepare and support 100 Utah educators to teach the Exploring Computer Science (ECS) curriculum.

CSTA: Tell us about the process.
Hu: The idea for developing a half-year ECS course for Utah high schools advanced over several meetings of the Utah CSTA chapter during 2011. Then, during a CSTA Birds-of-a-Feather session at SIGCSE 2012, I connected with Don Yanek and Baker Franke of the CSTA Chicago chapter, who were working on an ECS course for Chicago Public Schools. Later at an NSF Showcase, I spoke with Dan Lewis, who introduced ECS to Silicon Valley high schools. I heard about the NSF CE21 program solicitation (with a deadline of April 2012) at that same event.

After returning to Utah, I worked with chapter leaders Jay McCarthy (Brigham Young University) and Carl Lyman (Utah State Office of Education) to write our NSF proposal. Dan Lewis (Santa Clara University) and Lucia Dettori (DePaul University) generously shared their NSF proposals and provided invaluable advice from their experiences.

CSTA: How did you focus your grant proposal on NSF goals?
Hu: To better meet the goals of the NSF CE21 program, we added significantly more support for our ECS teachers. Not only were we able to offer a stipend for attending the summer professional development workshops, but we also added mentoring and monthly meetings to occur during the teacher's first semester teaching ECS. We added plans to recruit and support teachers and students in school districts with high populations of Hispanic, Native American, African American, and Pacific Islander students. To better serve teachers statewide, we created a partnership with Southern Utah University, which will host an ECS summer workshop in 2015. Finally, we connected with Lisa Kohne, who assisted us in developing a thorough evaluation plan and will serve as our external evaluator.

CSTA: Do you have advice to offer other chapters? Are there resources you would recommend?
Hu: Anyone writing a grant proposal should carefully read through the program solicitation multiple times during the writing process. I also read through successful proposals to similar solicitations by Dr. Lewis and Dr. Dettori. Work to complete the project description early because completing the budget online requires more time than most people expect.

I searched online for help in writing NSF proposals. Harvard University has a helpful website (search on "Harvard job aids") and Georgia Tech has a thorough *Proposal Preparation Workbook*.

CSTA: Who supported your application?
Hu: Prior to learning about NSF's CE21 program solicitation, we had created a mailing list of schools interested in piloting ECS during the 2012–2013 academic year. Teachers were asked to provide letters of support from their administrations; letters were received from approximately two-thirds of the schools before the due date.

We also solicited letters from one of the authors of the ECS curriculum, Gail Chapman, who will lead our ECS professional development, and Lisa Kohne, our external evaluator.

CSTA: What has been the impact so far?
Hu: Based on our success in bringing the ECS curriculum to Utah schools, we have been invited to take part in a discussion on the future of CS education in Utah public high schools. One possibility under consideration is to make CS part of the Utah Science Core.

CS Around the World

Bricklayers to Architects: Transition for CS in England

Michael Jones

Since January 2012 the fortunes of Information, Communication Technology (ICT) education in England have waned considerably as the growing discontent and dissatisfaction with the depth of study and resultant qualifications gained by students has led many in education and commerce to question whether ICT deserved a place in an already overburdened secondary curriculum.

Perhaps the best way to describe what is happening in England is to say that ICT has been increasingly viewed as a curriculum that prepares bricklayers rather than a curriculum that prepares architects. This was made clear by Michael Gove, Secretary of State for Education in England, who stated: "Children are being forced to learn how to use applications, rather than to make them. They are becoming slaves to the user interface and are totally bored by it" (*www.bbc.co.uk/news/education-16493929*).

The ICT curriculum delivered over the past decade became viewed, at best, as preparing students to use applications such as word processing and spreadsheet software. While these are valuable skills, problems arose when the pedagogy and student achievement were scrutinized. With figures approaching a 100% pass rate, inevitable comparisons were made between ICT and other subjects, such as mathematics and science, with lower pass rates. In short, ICT was seen an 'easy' subject to pass and therefore of less educational value (*www.education.gov.uk/inthenews/inthenews/a00201864/harmful-ict-curriculum-set-to-be-dropped-this-september-to-make-way-for-rigorous-computer-science*).

The growing dissatisfaction with ICT also needs to be viewed against the backdrop of major curriculum change proposed by the government in which the subjects a school offers can have a very real effect on the future of that school. In England, the quality of a school is based, at least partially, on the benchmark of five General Certificates of Secondary Education (GCSEs) or equivalent subjects in which a student achieves a grade C and above. Targets are set for the number of GCSEs a school must gain at grade C or above. In reality, anything below a C is counted as a failure. ICT qualifications include: Diploma in Digital Applications (DiDA), Nationals in ICT (OCR), European Computer Driving License (ECDL), ICT Business and Technology Education Council (BTEC), and more.

Unlike core GCSE qualifications that equal one GCSE, qualifications such as BTEC provide students the ability to gain up to four GCSEs. There has been concern/controversy about qualifications that are equivalent to GCSEs as to whether these alternatives are of equal academic value. The picture is clouded further in that some of the alternatives offer up to four GCSE equivalents. For example, if a student enrolls in BTEC Art, the more units taken, the more GCSEs are earned. There has been a move to remove this anomaly by allowing a qualification to be counted as equal to only one GCSE regardless of how much has been studied.

There is also no doubt that the elimination of ICT will have a profound effect on teachers. If ICT courses are eliminated from schools, ICT teacher trainees may have fewer employment opportunities. ICT is not computer science (CS) and many graduates choosing ICT education as a subject specialism would not ordinarily be expected to possess a computing degree.

So we come back to the bricklayer and architects question. We have many individuals who are able to use applications to varying degrees of proficiency (bricklayers) but do we have enough people capable of building the applications and hardware for the next generation of users (architects)? From the UK government's perspective, we do not produce a sufficient number of programmers and hardware engineers.

Over the past year, lobbyists from education and commerce have developed a new curriculum that subsumed ICT into CS. More information on the new Computing curriculum which

continued on page 8

SHOW ME THE NUMBERS

WHO WILL YOU MEET AT THE CSTA ANNUAL CONFERENCE?

Teacher/Faculty	**81%**	**Teach CS**	**60%**
High School	**50%**	**Teach IT**	**26%**
Middle School	**13%**	**Tech Coordinators**	**7%**
Higher Education	**18%**	**Administrators**	**2%**

Source: CSTA Conference Attendees 2012

csta Voice

Computer Science Teachers Association
2 Penn Plaza, Suite 701
New York, NY 10121-0701

We're on the Web.
Join our Facebook page!
csta.acm.org

BRICKLAYERS TO ARCHITECTS

continued from page 7

replaces the ICT curriculum is available from the British Computer Society through their Computing at School section (*www.computingatschool.org.uk/*). CS has now replaced ICT as a curriculum subject. To give credence to this new subject, it has been moved into the science arena and will be judged alongside physics, chemistry, and biology as a benchmark for English schools. This brings with it a host of other issues.

The future looks to be interesting. Already there are rumblings that there are woefully small numbers of suitable graduates entering CS teacher training. Even if the number were to rise substantially, it would take some years for expertise to be built and suggested syllabi to embed.

Membership News

Join the Discussion

Ria Galanos

CSTA is pleased to announce the launch of a new member benefit—a listserv to encourage communication on topics and about issues related to computing education. Unlike other computing educator electronic communities that focus on particular languages or courses, the CSTA listserv invites educators to join discussions on overarching questions that transcend tools and courses.

The CSTA community is a unique space for our 13,000+ members to ask questions and share ideas about recruitment techniques, project-based learning, best practices for inspiring students, techniques for teaching college-level work in the high school, textbooks, assessments, and more. CSTA members must opt in to this member benefit. To join the conversation, visit: *listserv.acm.org/scripts/wa-ACMLPX.exe?SUBED1=csta-members&A=1*.

VOTE

All CSTA members should have received their election ballot via email by April 5. If you did not receive your ballot, contact:
nominations@csta.acm.org

Voting closes May 3, 2013

Post your professional development events on the CSTA website by contacting *l.clayborn@hq.acm.org*
csta.acm.org/ProfessionalDevelopment/sub/TeacherWorkshops.html

MARK YOUR CALENDAR

Random Hacks of Kindness Junior
May 4, 2013, Newtown, Connecticut
www.rhok.org/event/rhok-junior

ACSL All-Star Contest
May 25, 2013, Raleigh, North Carolina
www.acsl.org

Summer Academy for Advancing Deaf and Hard of Hearing in Computing
June 21–August 24, 2013, Seattle, Washington
www.washington.edu/accesscomputing/dhh/academy

Carleton College Summer Teaching Institute 2013
June 25–28 and July 9–12, 2013, Northfield, Minnesota
apps.carleton.edu/summer/teaching

Annual CSTA Conference (formerly CS & IT)
July 15–16, 2013, Boston, Massachusetts
www.cstaconference.org

Bootstrap: Videogame Programming with Algebra
July 17, 2013, Boston, Massachusetts
bit.ly/bootstrapMA

Teaching Java and Graphics Programming Fundamentals through Art and Game Creation
July 21, 2013, Anaheim, California
buildingsteam.org/?page_id=21

Summer Computer Science Institute
July 21–August 9, 2013, Northfield, Minnesota
apps.carleton.edu/summer/scsi

Scratch Connecting Worlds Conference
July 25–27, 2013, Barcelona, Spain
scratch2013bcn.org/

CS4HS 2013 at Carnegie Mellon University
July 31–August 2, 2013, Pittsburgh, Pennsylvania
www.cs.cmu.edu/cs4hs/summer13

Java Fundamentals and Programming Teacher Training and Curriculum
August 7–9, 2013, Salt Lake City, Utah
academy.oracle.com/pages/prog_commit_inst_institute.htm

National Cyber Security Awareness Month
October 2013, Department of Homeland Security
www.dhs.gov/national-cyber-security-awareness-month

Check the most recent CSTA events on the CSTA website.

March Voice Correction: The complete "Blocks Programming" article by Michael Tempel can be found at: ***csta.villanova.edu/handle/2378/475***