csta Voice

New in CS and CT

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

Volume 7, Issue 3 July 2011

Inside This Issue

FEATURES

NEW! K–12 CS Standards
Is Computational Thinking the Fourth "R"?
CSTA Membership Benefits
Pex4Fun
Computer Education in the Era of the Greek Digital School

COLUMNS

Research Review
Classroom Tools
Curriculum in Action
College Connection
Show Me the Numbers

INFO BRIEFS

CSTA Thanks
Staff
Membership Renewal
Contribute to the CSTA Voice
Contact Info
Election News
Meet the Authors
Mark Your Calendar
Resources

IN THE NEXT ISSUE OF THE *VOICE*

Back to School

NEW! K–12 Computer Science Standards

Chris Stephenson

CSTA IS WORKING on a new document that will update the comprehensive learning standards for K–12 computer science (CS) education.

It is widely recognized that a prescribed set of standards that detail what students are to understand and be able to do, and how this knowledge and these skills should be scaffolded throughout their school experiences, is critical to student success in school.

In 2003, the ACM K–12 Education Task Force undertook to create such a set of standards and published its efforts as the *ACM Model Curriculum for K–12 Computer Science Education*. (A second edition with a new preface was published in 2006.)

Since that time, the learning outcomes described in the *ACM Model Curriculum* have become the de facto national standards for CS education in the U.S. More than 40,000 hard copies of the document have been distributed and its contents have been used to develop state-level standards and innovative new curricula that have changed the face of CS education both nationally and internationally.

In the last eight years, however, K–12 CS education has exploded with a wealth of new technologies, approaches to curriculum, and strategies to better engage all students. For this reason, CSTA (created by ACM in 2004) put together a new team with representation from all levels of education to re-envision and revise the standards. This team, led by Allen Tucker (Bowden College) and Deborah Seehorn (North Carolina Department of Public Instruction), has now released a draft of the new *CSTA K–12 Computer Science Standards*.

The new standards are organized into three levels:

- Level 1: Computer Science and Me
- Level 2: Computer Science and Community
- Level 3: Applying Concepts and Creating Real-world Solutions

At each level (and where specific courses are described) the learning outcomes are organized into five strands. These strands are:

- Computational Thinking
- Collaboration
- Computing Practice
- Computers and Communication Devices
- Community, Global, and Ethical Impacts

Level 1 provides a set of learning standards, organized by strand, for grades K–3 and grades 3–6 that can be taught across the curriculum. Level 2 provides a set of standards for grades 6–9 that can be integrated across the curriculum or taught in discrete computing courses.

Level 3 outlines a program of study that includes three courses:

- 3A: Computer Science in the Modern World
- 3B: Computer Science Concepts and Practices
- 3C: Topics in Computer Science

continued on page 2

Executive Officers

Michelle Hutton
President
mfh@pobox.com

Stephen Cooper
Vice-President
cooper@cs.stanford.edu

Staff

Dr. Chris Stephenson
CSTA Executive Director
Phone: 1-800-401-1799
Fax: 1-541-687-1840
cstephenson@csta.acm.org

Pat Phillips
Editor
Phone: 1-608-436-3050
Fax: 1-928-855-4258
cstapubs@csta.acm.org

Committees

Certification
cstacertification@csta.acm.org

Curriculum
cstacurriculum@csta.acm.org

Funding Development
cstagrants@csta.acm.org

Membership
cstahelp@csta.acm.org

Professional Development
cstapd@csta.acm.org

Research
cstaresearch@csta.acm.org

NEW! K–12 CS STANDARDS

continued from page 1

In order to ensure that the new CSTA standards reflect the best thinking of the practitioner community, CSTA released a draft of the standards for a period of community feedback. This draft is available from the CSTA website at: *csta.acm.org*. The results of that feedback are now being used by the team to complete its revisions, with the goal of having the final document published by December 2011.

Is Computational Thinking the Fourth “R”?

Joan Peckham

IN THE MAY 2011 ISSUE of the *Voice*, I discussed the origins of computational thinking (CT), provided a few high-level descriptions and asserted the need for infusing it into everyone’s education. In this issue I will discuss a few emerging definitions, connect them to a few concrete examples, and point to additional CT resources.

In a recent discussion about CT, Jeannette Wing and Peter Denning talked about two definitions, both influenced by an earlier definition by Al Aho. Their discussion can be found at: *www.computingportal.org/CE21*. Jeannette Wing currently uses the following working definition that she formulated with Jan Cuny and Larry Snyder in 2010:

> **Computational Thinking** is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent.

Jeannette adds that solutions can be carried out by any processing agent: human, computer, or a combination. Peter provides a consensus definition given by the computing community in the ACM Ubiquity Symposium, “What is Computing?”:

> **Computational Thinking** is an approach to problem solving that represents the problem as an information process relative to a computational model (which may have to be invented or discovered) and seeks an algorithmic solution.

This CT definition includes the thinking skills needed to design and implement the solution once the algorithm is formulated.

As I pointed out in the earlier article, the definition will evolve and emerge from multiple communities. However, the need to infuse these skills into the very best education and training of everyone is pressing. To this end, CSTA and ISTE convened a group of educators, computer scientists, and cognitive psychologists to develop an operational definition of CT for K–12 which is described more fully in an article by Barr and Stephenson in the March 2011 issue of *Inroads*.

These and other CT resources can be found at: *csta.acm.org/Curriculum/sub/CompThinking.html.*

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.

Being more assessable and providing a solid conceptual base, this definition will enable us create concrete examples of standards, materials, and pedagogies that will help us begin to build robust CT learning. This operational definition has two parts: (1) characteristics, and (2) dispositions and attitudes. Let us look at a few examples.

One CT disposition is *persistence in working with difficult problems*. Debugging might be considered as an example. Systematic debugging is a CT skill that is taught to high school and college students when they learn to program, but it is useful in many other settings. However, we typically have not explicitly taught debugging skills in most classes in the same way we would in a programming class.

Let us consider a writing lesson. Today, most student writing experiences involve instruction in composition and grammar followed by a writing assignment, perhaps with some motivational topics or discussions to foster the desire to write. This is followed by comments and a grade by the teacher; frequently, this might be the end of the lesson. But as every good writer knows, the next steps require responding to the comments of other readers with a second, third, or fourth version.

The goal of a more robust lesson might be to write a piece that intentionally communicates information to others with a means to identify if the task has been accomplished. One can imagine using digital means to accomplish this using modern online writing and communication tools. Can the student, for example, successfully communicate to another via a constrained electronic medium (Twitter, e-mail, Instant Messenger, Skype, or other) how to carry out a specific task correctly? And if the recipient of the instructions is tested and does not succeed, can the writer then learn to debug and improve the instructions? One might include another disposition skill to this exercise, "the ability to communicate and work with others to achieve a common goal or solution" by devising a group-based writing exercise.

An example of a CT characteristic is *representing data through abstractions such as models and simulations*. This is an important thinking and analysis skill needed by everyone. Before he disappeared at sea in 2007, Jim Gray introduced data exploration, now widely embraced as the Fourth Paradigm of Science to be used along with hypothesis-driven and other traditional approaches already taught in K–12 classrooms. Simulation is now used to enable deep scientific understanding previously inaccessible to students before they develop deep mathematical skills. See the work of Bob Panoff—dedicated to reform and improvement of mathematics and science education through the incorporation of computational and communication technologies (*www.shodor.org*), or Alexander Repenning—who uses the Agent Sheets gaming environment to teach students the skills needed to build scientific simulations (*www.cs.colorado.edu/~ralex*).

It is time to engage our communities and resolve to strengthen everyone's education by infusing CT across the curriculum. Needed is teacher training, and shared materials appropriate for every stage of development, K-Grey, for formal and informal environments. I hope that CSTA members will embrace this challenge, roll up their sleeves, and help in this difficult but important mission.

CSTA Membership Benefits

Dave Burkhart

A CSTA MEMBERSHIP, individual or institutional, is one of the best deals around. The CSTA website is your connection to a vibrant community of computer science (CS) educators. It provides a wealth of resources including tools for teaching about CS careers, relevant curriculum guides and teaching materials, and professional development opportunities.

Members have access to a variety of resources for promoting CS careers, including classroom posters, brochures, and videos. High quality posters that inform students about

continued on page 4

Election News

Congratulations!

Newly-elected CSTA Directors

K-8:
Patrice Gans
Connecticut CSTA

9-12:
Karen Lang
Boston CSTA

9-12:
Ria Galanos
Georgia CSTA

College:
Dave Reed
Creighton University

At-Large:
Mindy Hart
Purdue University

All of the proposed changes to the by-laws were approved.

Meet the new members of the CSTA Board.

csta.acm.org/About/sub/AboutFiles/CandidatesPage.html

CSTA MEMBERSHIP BENEFITS

continued from page 3

CS careers can be downloaded from the CSTA website or printed copies can be mailed upon request. “Computer Science in Sports” is the newest poster highlighting Kate Starbird, a former Stanford and WNBA basketball star who is pursuing a doctorate in CS (*csta.acm.org/Resources/sub/BrochuresPostersVideos.html*).

The CSTA *Source* online repository includes lesson plans and other resources matched to standards in the *ACM K–12 CS Model Curriculum*. Resources are searchable by curriculum topic or keyword. Items have been submitted by members and most are teacher-created. New resources are constantly being added and CSTA is eager for your submissions (*csta.acm.org/WebRepository/WebRepository.html*).

CSTA individual members have preferred registration for CSTA-sponsored professional development events such as the CS & IT Conference—the premier CSTA annual event. CS & IT 2011 is a multi-day event with hands-on workshops, keynotes, breakout sessions, and the Microsoft Imagine Cup Grand Finale (*www.csitsymposium.org*).

Institutional memberships are intended for organizations such as corporations and university or college departments of CS and engineering. There are many institutional membership benefits, including opportunities to announce K–12 events on the CSTA website, promotion of institutional member events through the *Voice*, and a link to the institutional member website from the CSTA website (*csta.acm.org/Membership/sub/InstitutionalMembership.html*).

A popular institutional member benefit is the opportunity for free e-mail blasts to CSTA’s growing membership regarding institutional events. Institutional members have reported that the response to CSTA email blasts always exceeds their expectations.

Communication to CSTA members through the CSTA bi-monthly *Voice* newsletter publication reaches approximately 10,000 members online and through the postal service.

Membership in CSTA means belonging to a community dedicated to excellence in CS education. Invite your colleagues to join CSTA—the authority in K–12 CS education with membership representing 114 countries worldwide.

Pex4Fun

Cloud Computing in the CS Classroom

Nikolai Tillmann, Jonathan de Halleux, and Tao Xie

IMAGINE TEACHING COMPUTER SCIENCE (CS) with programming activities that begin with “Hit the big button to start the game.” Pex4Fun, a web-based gaming environment for learning CS, makes this engaging scenario a reality.

Students are presented with coding duels in which they are challenged to implement an algorithm to match an expected output of a secret solution. Along the way to solving the problem, students receive hints and analyze the program output as the Pex4Fun website keeps finding unexpected cases where the student’s program does not behave in the same way as the secret program. The live action of dueling students can be displayed to the class. When students win Coding Duels, they can even publish their achievements on their Facebook pages. In addition to solving Coding Duels, students can also submit new Coding Duels to challenge others.

Pex4Fun works on any web-enabled device. It comes with an auto-completing code editor, providing a student with instant feedback, similar to the code editor in *Microsoft Visual Studio*. Because the data is available in the cloud, students can begin in the classroom and continue at home.

Beyond individual Coding Duels, teachers can create entire courses, combining explanatory text, example code, and Coding Duels as exercises. Students register for a course and the teacher is informed about their progress, including which exercises each student has completed—in effect, automatically grading the assignments.

Pex4Fun was developed in the Research in Software Engineering (RiSE) group at Microsoft Research. One promising testing technique, which draws on advances in software verification and automated theorem proving, is dynamic symbolic execution. In this technique a program is executed multiple times with different inputs, while the execution paths are monitored at the instruction level, constructing a symbolic representation of the conditions checked by the branching statements.

Coding Duels in Pex4Fun leverage this technique; by thoroughly analyzing this program with dynamic symbolic execution, Pex4Fun generates a test suite that is customized to both programs. Every time a student submits a new program, Pex4Fun generates a new test suite, showing any behavior mismatches with the secret program, or, if there are no mismatches, indicates that the student has won the Coding Duel.

Since 2010, visitors have submitted about 300,000 programs in attempts to solve the more than 100 available duels. There are duels in C#, Visual Basic, and F#. Ready-to-use courses are available for C# and XNA. Visit *Pex4Fun.com* and try out the experience!

Computer Education in the Era of the Greek Digital School

Socrates Dimitriadis

GREECE IS IN THE MIDST of making far-reaching decisions about the role of computer science (CS) education in the national curriculum but there is no comprehensive CS or information technology curriculum for primary and secondary education. Currently:

- Elementary students (grades 1–6) do not receive any computer education.
- Middle school students (grades 7–9), receive only one hour per week.
- High school students (grades 10–12), are offered CS as an elective course. (CS is mandatory only for 12th grade students in the technological track who are applying to technical departments of higher education.)
- Teachers of non-CS disciplines can become permanent CS teachers after attending a short seminar on basic computer skills.

The Greek government recently announced reform of the Greek educational system and coined the term “the new digital school” to show an emphasis on computer education. Sadly, there is no room for CS in the curriculum of the “digital” high school. No plans have been revealed for the middle school level; retaining the single hour of computer education is still being debated. A promise has been made that the new elementary curriculum is going to introduce an across-the-curriculum implementation of computer applications in several subjects.

The Ministry of Education believes that this horizontal adoption of computing tools will provide all the computer literacy that students require, and that CS courses are unnecessary in secondary education. According to the Minister of Education, Anna Diamantopoulou, in an interview in Silicon Valley, “by the time kids get to middle school they already know how to use computers and the Internet.”

Greece ranks last among EU countries in all issues related to science and innovation. How can we reverse this situation if the digital goal of the new digital school is to provide only the most basic computer skills?

Unfortunately, some education policy makers do not understand the differences between computer literacy and CS. The Hellenic Informatics Union is working to raise public awareness of the importance of CS education and to impact the government’s decisions to provide the skills students will need to be competitive and innovative.

Meet the Authors

Dave Burkhart
CSTA Membership
Dave teaches computer technology at Sheridan HS in Thornville, OH, and is an adjunct faculty member at Zane State College.

Socrates Dimitriadis
Hellenic Informatics Union
Socrates leads an HIU initiative to evaluate higher education CS programs in Greece.

Jonathan de Halleux
Microsoft Corporation
Jonathan is a Senior Research Software Design Engineer, working within the Research in Software Engineering group of Microsoft Research.

Eric Freudenthal
University of Texas at El Paso
Eric is an Associate Professor of CS and studies the pedagogy of computation and engineering.

Mindy Hart
CSTA Board
Mindy is the K–12 Outreach Coordinator for the Department of CS at Purdue University. She is the At-Large Representative on the CSTA Board.

Amber Settle
DePaul University, Chicago
Amber is an Associate Professor in the School of Computing and Director of the Innovation in Technology Education Center at DePaul.

Nikolai Tillmann
Microsoft Corporation
Nikolai is a Software Design Engineer and leads the Pex project, a framework for runtime verification and automatic test-case generation for .NET applications.

Joan Peckham
University of Rhode Island
Joan is a Professor of CS. She is currently on rotation at the National Science Foundation where she is serving as a Program Director in the Office of Cyberinfrastructure.

Tao Xie
North Carolina State
Tao is an Associate Professor in the Department of CS in the College of Engineering and leads the Automated Software Engineering Research Group.

Research Review

Computational Thinking for K–12 Students

Amber Settle

Jeannette Wing defined computational thinking (CT) as "the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent" and she argued that CT is an emerging basic skill that should become an integral part of education.

That vision is taking shape through the work of the researchers on the "Computational Thinking across the Curriculum" project. They have identified and categorized instances of CT in 19 general education courses in the Liberal Studies Program at DePaul University. The 18 DePaul faculty members on the project have helped to produce a framework that allows faculty without training in computing to understand and integrate CT in their own courses. The framework has been reviewed and evaluated by faculty from the University of Illinois at Chicago, Loyola University, and the Illinois Institute of Technology, as well as by Klaus Sutner at Carnegie Mellon University. The project was funded from the NSF CPATH program and led by principal investigator (PI) Amber Settle and co-PI Ljubomir Perkovic.

During the first two years of the project, the focus was post-secondary education. However, a minority of people in the U.S. attends and graduates from colleges and universities, and Wing's vision cannot be achieved unless the kind of work that has been done at DePaul can be replicated in the K–12 curriculum.

In the past year the focus of the project has been the integration of CT into select courses at the University of Chicago Laboratory Schools. Baker Franke, Ruthie Hansen, and Fran Spaltro have created and implemented CT activities for computer science courses in grades five, six and nine and a high school Latin course. In particular, Baker Franke is working to determine if, and how, examples help ninth grade students to better understand Python programming, particularly with respect to knowledge transfer. Ruthie Hansen is extending an existing "choose your own adventure" story project for fifth and sixth grade students using logical maps and networks to determine whether this approach aids student understanding of plot in fictional storytelling. And Fran Spaltro has modified her Latin class to more explicitly emphasize the CT techniques inherent in notating, diagramming, and metaphrasing Latin sentences.

The teachers in the Laboratory Schools are currently in the process of gathering student data to evaluate the effectiveness of the new approaches. They presented their work at the CPATH Research Experience for Teachers meeting in February 2011, where it was well received. In keeping with the focus of the original project, Amber Settle and Baker Franke are now reaching out to Laboratory School faculty who teach history, fine arts, and English to help them integrate CT into their middle and high school courses.

The project participants believe that teaching students CT in the context of other disciplines is the most fruitful approach. We would like to work with researchers from the "Building the Northwest Distributed Computer Science Department" CPATH project on finding ways to integrate CT into multiple disciplines outside of computing in the K–12 curriculum and then evaluate the effect of that work on students' attitudes toward computing.

For more information on that project contact Amber Settle at: *asettle@cdm.depaul.edu*.

Classroom Tools

Virtual Binders Provide Valuable Resources

Mindy Hart

A great resource available to you as a CSTA member is something called "K–12 Virtual Binders." What is a virtual binder you ask? Well, just like a physical binder, a virtual binder is a place to keep important documents. In this case, the important documents are the very best scholarly articles and documents from the ACM Digital Library that were selected by CSTA teacher volunteers for their relevance to K–12 computer science issues. The binders are organized by topics: Equity, Teaching Strategies, and Careers.

The K–12 Virtual Binders are available from the CSTA website under "Resources." The binders are accessed via your CSTA web account which is hosted on the ACM site. If you have not already done so, you will need to set up a CSTA web account. It is quick and easy! (*csta.acm.org/Resources/sub/VirtualBinders.html*).

CSTA members are working to revise the K–12 Virtual Binders; new binder topics and resources will be available starting this summer—just in time to kick off a new school year. If you have ideas for articles that should be in the Virtual Binders, let us know.

Curriculum in Action

Using Graphical Programming to Introduce CS

Eric A. Freudenthal

Imagine being introduced to computer science (CS) in the same way that you might be introduced to Chinese in a conversational language course—producing results with lessons designed to be easily engaged by delaying the complexities of vocabulary and syntax.

iMPaCT is a media-propelled (MP) approach to engaging students in computational thinking (CT) and CS suitable for high school and college math, science, and CS classes. It uses a learner-centric approach which accelerates engagement with CS and algorithm design, even in non-CS courses. iMPaCT's lessons motivate students by rendering problem solutions graphically in a manner that builds, rather than relies upon, mathematical maturity.

A key aspect of iMPaCT is its single-session introduction to programming that enables it to be incorporated into courses from middle school enrichment programs to high school and college math and CS courses. iMPaCT has been taught using interactive language systems such as Jython or even BASIC interpreters within graphing calculators common in math classrooms.

With iMPaCT, students are initially introduced to simple and easily extended idioms that reveal key concepts within easily understood problems. At each step, students are challenged to extend the programs in ways that expose key concepts. In the initial 45-minutes session students learn and practice:

- Assigning values to named variables and simple algebraic operations;
- Using a named function within a program that draws dots on a raster display; and
- Iteration as a solution to drawing horizontal, vertical, and tilted lines.

Depending on the course, subsequent lessons can focus on the exploration of math and/or programming skills. For example, a program that creates a curve may be examined during a lesson motivated by the challenge of rendering a non-linear form, thus exposing students to the relationship between functions and their rates of change. Later, the students are led to recognize that this approach to generating curves can also be employed to animate (and understand the principles underlying) the trajectory of a thrown object.

By focusing on problems with relevance outside of the typical curriculum, iMPaCT can be implemented as an enrichment activity within conventional math, science, and engineering courses, and introductory CS courses. In this manner, a large fraction of students can be holistically introduced to computing. Pilot studies have yielded dramatic results including:

- One quarter of eligible students of *both genders* who attend a high school math course incorporating iMPaCT subsequently enrolled AP CS courses.
- Dramatic increases in pass rates in both high school and college math courses.

We are examining introducing all incoming freshmen at University of Texas at El Paso (UTEP) to programmed computation by embedding iMPaCT into required math courses. iMPaCT has also been incorporated within required introductory programming courses intended for students studying a variety of disciplines. For example, UTEP's electrical engineering students' first programming course has become more effective and engaging to students when modified to introduce core programming concepts in the context of electrical simulation problems written in Python. After this preparation, these students mastered C far more quickly than in a conventional course.

For more information on iMPaCT visit: *cs.utep.edu/impact*. Teachers are encouraged to provide feedback and contribute teaching resources to iMPaCT.

College Connection

University of South Carolina

Editor's note: *This dialogue with Duncan Buell, Professor in the Department of Computer Science and Engineering at the University of South Carolina (USC) is a continuation of our series of interviews with CSTA institutional members. Please share these details about the computer science (CS) programs at USC with your students.*

The University of South Carolina is located in Columbia, South Carolina, and has about 26,000 students. Three undergraduate computing majors are offered: Computer Science, Computer Information Systems (CIS), and Computer Engineering (CE). The CS coursework in the three degree programs is quite similar. However, the CIS students receive a minor in Business Information Management and the CE students earn the equivalent to a minor in Electrical Engineering. Students in the CS program study a second interest area of their choice. At the graduate level, USC offers a master's degree and a Ph.D. in Computer Science and Engineering (*www.engr.sc.edu/dept_progs.html#cse*).

CSTA: What draws students to your program?
Buell: For many in-state students, the draw is that USC is in Columbia and only a block from the state capitol. Because there are many local computing businesses close to campus that provide students with part-time jobs, students can fit work around their class schedules and enhance their resumes at the same time.
CSTA: What skills will help students succeed in your program?
Buell: Getting a solid academic background is even more important than learning CS in high school. Reading, writing, thinking critically, and understanding abstraction are important skills for success. Although it is certainly an advantage to have CS knowledge before coming to USC, it is not necessary. Tenacity is an important personal trait for tackling both educational and CS challenges.
CSTA: Tell us about innovative programs of study.
Buell: As a large public university, USC offers a wide range of other programs to complement a CS major. Students can select from such diverse areas as media arts, gaming and animation, security issues, public policy, and criminal justice. And the Honors College, which includes an honors thesis project, enables students to accelerate their program of study.
CSTA: What cool careers are graduates prepared for?
Buell: Many of our graduates stay in the geographic area. Large finance and business services companies located in Atlanta and Charlotte provide many opportunities. A number of our students become entrepreneurs; some succeed even while in school such as a recent freshman who set up his own Android apps company. Part-time jobs can also open doors to entrepreneurialism.
CSTA: What distinguishes your school and program?
Buell: While our program is academically very similar to that of the other two large in-state universities, we have two distinguishing characteristics. First, USC is a downtown campus in a metro area with many opportunities very close by for part-time work. Secondly, undergraduate students at USC are encouraged to participate with faculty in research and at conferences. Many students value the flexible time schedules and challenges afforded by these experiences that are atypical of most classroom environments.
CSTA: Tell us about the social environment of the CS program.
Buell: Our ACM student chapter is active and students are very supportive of each other. There is an extensive student network of contacts both in and out of the classroom that contribute to their successes.
CSTA: What programs encourage the diversity of the CS student population?
Buell: USC has a larger African-American population than is common at the major public universities in the South. Our enrollments of African-American students and of women in the CS department are low but increasing. There are student chapters on campus of both the Society of Women Engineers (SWE) and the National Society of Black Engineers (NSBE), and they welcome CS students.
CSTA: Describe your programs for high school outreach and professional development.
Buell: The CS department has been active in reaching out to high schools with information about CS careers. Recently we worked with the South Carolina Department of Education to create a version of the Exploring CS curriculum that will be taught at USC beginning in the fall of 2011. We offer an AP Summer Institute and were instrumental in establishing a South Carolina chapter of CSTA. We also work closely with the Consortium for Enterprise Systems Management, a group of more than fifty companies concerned about keeping the CS talent pipeline full, to provide career information resources for the high schools.

SHOW ME THE NUMBERS

CSTA GROWING IN NUMBERS AND INFLUENCE

	MEMBERSHIP	COUNTRIES REPRESENTED	CSTA CHAPTERS
June 2008	4781	100	5
June 2009	6374	103	17
June 2010	7014	105	29
April 2011	8635	114	32
GOAL JULY 2011	10000		38

Source: CSTA Membership Committee (April 2011)

csta Voice

We're on the Web! *csta.acm.org*

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

NATIONAL SCIENCE FOUNDATION

MARK YOUR CALENDAR

CS4HS: Carnegie Mellon
July 6–8, 2011, in Pittsburgh, Pennsylvania
www.cs.cmu.edu/cs4hs

CS4HS Workshop
July 7–8, 2011, at Columbia University in New York City
www.cs.columbia.edu/~cs4hs

CS & IT Symposium
July 11–13, 2011, in New York City
www.csitsymposium.org

Creative Computing CS4HS Workshop
July 27–30, 2011, at MIT in Cambridge, Massachusetts
cs4hs.media.mit.edu

U.S. Innovative Education Forum
July 28–29, 2011, in Redmond, Washington
www.microsoft.com/innovativeeducator

Consortium for Computing Sciences in Colleges (CCSC: Midwestern)
September 23–24, 2011, in Huntington, Indiana
www.ccsc.org/midwest/Conference

Consortium for Computing Sciences in Colleges (CCSC: Northwestern)
October 7–8, 2011, in Richland, Washington
www.ccsc.org/northwest

Consortium for Computing Sciences in Colleges (CCSC: Rocky Mountain)
October 14–15, 2011, in Orem, Utah
www.ccsc.org/rockymt

Consortium for Computing Sciences in Colleges (CCSC: Eastern)
October 14–15, 2011, in Arlington, Virginia
www.ccsc-e2011.org

Grace Hopper Celebration of Women in Computing
November 9–12, 2011, in Portland, Oregon
gracehopper.org/2011

Consortium for Computing Sciences in Colleges (CCSC: Southeastern)
November 11–12, 2011, in Greenville, South Carolina
www.ccscse.org

TCEA
February 6–10, 2012, in Austin, Texas
tcea2012.org/2012/public/default.html

SIGCSE 2012
February 29–March 3, 2012, in Raleigh, North Carolina
www.sigcse.org/sigcse2012

RESOURCES

Here's more information on topics covered in this issue of the *CSTA Voice*.

Page 1: CSTA *csta.acm.org*
Page 1: CS Curriculum Standards *csta.acm.org/includes/Other/CS_Standards.html*
Page 2: CE-21 Portal *www.computingportal.org/CE21*
Page 3: CS & IT Conference *www.csitsymposium.org*
Page 3: Shodor *www.shodor.org*
Page 3: Agent Sheets *www.cs.colorado.edu/~ralex*
Page 4: Pex4Fun *Pex4Fun.com*
Page 4: Microsoft Visual Studio Express *www.microsoft.com/express*
Page 4: CSTA Chapters *csta.acm.org/About/sub/CSTAChapters.html*
Page 5: Research in software engineering (RiSE) *research.microsoft.com/en-us/groups/rise*
Page 5: Hellenic Informatics Union *www.hiu.gr*
Page 6: DePaul University *www.depaul.edu*
Page 6: University of Chicago Laboratory Schools *www.ucls.uchicago.edu*
Page 6: CPATH *www.cpath-community.msu.edu/about*
Page 6: National Science Foundation *nsf.gov*
Page 6: CSTA Virtual Binders *csta.acm.org/Resources/sub/VirtualBinders.html*
Page 6: iMPaCT *cs.utep.edu/impact*
Page 7: University of South Carolina *sc.edu*
Page 7: ACM Student Membership *www.acm.org/membership/student*
Page 8: Consortium for Computing Sciences in Colleges *www.ccsc.org*