csta Voice

BUILDING A CS PROGRAM

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

Volume 5, Issue 2 May 2009

Inside This Issue

IN THE NEXT ISSUE OF THE VOICE

Careers in CS

Attracting the Next Generation of Students to Computing

Dan Lewis

Editor's note: *This is the first of a two-part series on re-designing computer science (CS) at the university level in order to engage today's students.*

The national decline of undergraduate enrollment in computing is a frustrating but familiar story. But while recent surveys suggest a modest improvement, we have yet to answer the key questions that should shape our thinking about the CS curriculum. Will the numbers ever return to those we saw in 2000? Do we really understand all the factors that caused the decline? Why has there been a greater loss of women students than men? Are today's students looking for something fundamentally different? Should we just assume that enrollment is cyclic and the numbers will return?

What works and what doesn't

Adding concentrations or specialization tracks to highlight the application of computing to areas such as robotics, embedded systems, or security have caused, at best, only modest increases in enrollment. While programs and concentrations in game development have successfully attracted increasing numbers of male students to computing, they have not had the same impact on women.

We know that making material more engaging and relevant helps, as evidenced by the widespread success of Randy Pausch's Alice programming environment and Mark Guzdial's Media Computation. While these efforts have been effective for attracting both male and female students to computing, a more comprehensive approach still seems necessary.

Attracting the next generation of students to computing

Social relevance seems to be an important factor for attracting today's students. While enrollment in electrical engineering and computing was falling during the last decade, other technical disciplines whose names suggest a more obvious and direct benefit to humanity—such as environmental science and biomedical engineering— remained popular, especially among female students.

Although there are numerous examples of how computing has benefited society, our discipline is more commonly associated with the image of an introvert who spends most of his time staring at computer screens rather than interacting with people. Accurate or not, the perception has discouraged many from entering the field. One may wonder whether degree titles containing only the phrase "computer science" (or similar variations) will ever be able to escape this deep-seated stigma.

Designing a new program

So how do we repackage computing so that students will take a second look? Can we make social benefit a significant and relevant curriculum component while

csta Voice

CSTA wishes to thank Barbara Ericson

for her work as chair of the CSTA Teacher Certification Task Force and in creating *Ensuring Exemplary Teaching in an Essential Discipline*.

ATTRACTING THE NEXT GENERATION

continued from page 1

preserving technical content?

Most undergraduate programs in computing require a senior capstone design project. When allowed to define their own, today's students often turn to Web-based projects that provide an opportunity for technical innovation developed within the context of a highly interactive and engaging medium.

By their senior year, most CS students have the necessary background in network protocols, databases, client-server architecture and Web programming languages to complete the technical implementation of such projects. Rarely, however, do undergraduate CS programs also require the equally Web-relevant subjects of graphic arts and multi-media design, the psychology of user-interface design, or courses in communication, sociology, or applied ethics—yet these are exactly those aspects of the Web where social interaction takes root and social benefit occurs.

This is not intended to suggest that we should replace our CS programs with programs that simply produce Web designers. As Wendy Hall (quoted by Paulson (2007)) explains, "All this doesn't mean we won't need computer scientists or engineers in the future. But we need a new type of person who understands the technology and human behavior, and the way society works."

To take advantage of this generation's love affair with the Web, we can and should offer them a comprehensive program that combines the part of CS and engineering that empowers the Web, the graphic design and Web programming technologies that are used to create its content, and an understanding of the social implications of its use. As noted in a recent article describing the emerging field known as Web Science, (Hendler, 2008)"… understanding the Web and being able to engineer its future requires not only an understanding of the Web as a computational structure, but also how it interacts with, and supports the interaction of, people."

References:

Paulson, L. (2007). Making Web science an academic field. *IEEE Computer*, 40, 22.

Hendler J., Shadbolt N., Hall, W., Berners-Lee, T., & Weitzner, D. (2008). Web science: An interdisciplinary approach to understanding the World Wide Web. *Communications of the ACM*, 51 (7), 60-69.

Making CS a Required Course

We Did It and So Can You!

Baker Franke

In 2008 the University of Chicago Laboratory Schools approved the addition of a ½ credit of computer science (CS) to the high school graduation requirements.

Before I provide the details on how we were able to bring this about, I want to add an encouraging note to readers: You can do this, too. I am sensitive to the fact

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 *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.

that the Lab School is a highly competitive independent school, and there are many issues that we don't have to deal with that public schools do. But that doesn't mean it was easy to accomplish the task—it took a consistent effort over almost 3 years.

I have been working with public school teachers in Chicago to add a CS graduation requirement and I can tell you that many of their issues are the same ones we had to deal with at our school. The issues are extraordinarily difficult to manage, but they are not insurmountable.

Our implementation plan

We discovered that 95% of our 9th graders had two free 45-minute blocks of time per week, an oddity because of how our science courses schedule lab time. We designed a CS course to be taken by all freshmen during these free blocks of time. It's a ½ credit course because it only meets twice per week (unlike full-time courses which meet 4 times per week). Adding the requirement in this way did not require additional staff and the rooms were already available. It was also very important to us that the required course not displace any other 9th grade courses.

Our plan allowed the required CS course to fit into times when no other full-time course could possibly be scheduled. This meant that there was no systemic or financial obstacle which would prevent the graduation requirement from being added. This may not be possible at all schools, but I should stress that figuring out that the course could be ½ credit (instead of a full credit) was not at first obvious to us. It took a long time to decide that we could teach a proper course with that amount of time.

Our argument for adding a CS requirement

It is difficult to argue for a CS requirement, in large part because there are so many valid arguments and when you start talking to people about educational goals that involve the word "computer," well, there's a lot of confusion. We thought the most effective arguments would involve showing how our course reflected the vision of John Dewey (the founder of our school) and showing how CS should be a part of a modern liberal arts education. We were wrong.

When we talked to various groups about the important role CS should play in everyone's education and that it involved the untapped and important skill of "computational thinking," the room would go silent, a cricket would chirp, some tumble weeds would roll by, and then someone would yell, "YOU'RE GOING TO KILL THE ARTS PROGRAM!" We weren't connecting.

The problem, of course, is that it's hard for many people to understand the importance of computational thinking without having experienced it. So, we just gave up on that strategy and tapped into something that our administration had been saying for some time—students need "tech skills." Most non-CS people have an intuitive sense that computing education is important, but the ideas of what that should be are all over the place. So, in the end we just said, "You're right! Our kids do need technology skills. Now let us design a course to most effectively deliver tech skills to our students." The course we created to accomplish this goal is a CS course.

> …we needed to teach them something about CS in order for technology skills to be retained and valuable.

Now before I get accused of pulling a bait-and-switch on my administration and colleagues, I will argue that this was a necessary step to take. We met very little resistance to the notion that technology skills are important. So we just used that as our starting point and explained why we need to teach technology skills by teaching students about the underlying principles of the technology that is so pervasive in their lives. In other words, we needed to teach them something about CS in order for technology skills to be retained and valuable.

Our answer to the zero-sum problem

When you propose adding CS as a graduation requirement, the first question

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.

MAKING CS A REQUIRED COURSE

continued from page 3

you're likely to get is "what do we have to lose in order to gain a CS requirement?" For a long time, we pleaded that it doesn't have to be a zero-sum game, and that our implementation plan showed that we could augment our academic program without losing anything. This was a mistake.

The problem is that people don't believe you if you say it won't cost anything. There is a cost, and you have to be honest and up front about it. If you try to sweep it under the rug, at best, people won't take you seriously, and at worst, they'll start inventing costs like, "YOU'RE GOING TO KILL THE ARTS PROGRAM!" This is a bad situation.

For us, the cost was 9th graders' free time. Our 9th graders are pretty tightly scheduled as it is and our implementation plan essentially cut in half any existing free time they had. Understandably, many in our community expressed concerns about student stress, additional workload created by the course, etc. These are serious and realistic concerns, and many still oppose the addition of the CS requirement. But we argued that the benefit of all of our graduates having some education in CS far outweighs the loss of free time in their 9th grade year.

If you're interested in working to make CS a graduation requirement at your school, more information is available at *csreq.org*. You can find resources for your proposal and contact information for others who have successfully campaigned to add a CS requirement.

Building CS Standards in Germany

Carsten Schulte

The phrase, "built by the community," is sometimes found in software development but rarely associated with developing educational standards. However, that is the perfect phrase to describe the new educational standards recently developed in Germany for lower-secondary computer science (CS) education.

Educational standards are critically important because they represent a shared vision of computing education; they define a curriculum with important learning goals and express the idea at the heart of computing education. A group of seven persons, lead by Hermann Puhlmann, organized in the German GI, and in collaboration with the community of computing educators and researchers, developed educational standards for lower secondary education.

Over the course of three years, the standards grew from the results of nationwide workshops, online questionnaires, and regional conferences. It turned out that the community shared many common beliefs about the heart of computing education. These common beliefs were shaped around the framework used by the NCTM standards for mathematics education. This framework provides a structure for defining a vision, five content standards, and five process standards.

The vision of the new computing education standards is to provide computing education for all students. The content standards define the five main computing ideas relevant for education. The process standards define the five most important educational processes needed to reach the goal of teaching and learning the standards.

> For many, it was quite surprising to have reached a consensus on many of the issues.

The role of programming in computing education was discussed and questioned: Is it a major idea and should it be a goal defined as one of the five content standards? Or is it a means to reach other goals? Might it be better defined within the process standards? Or is it not worthwhile as a standard for lower secondary education at all? The framework drove the discussion and helped the group find solutions for these and other questions in such

a way that the resulting standards were truly built and accepted by the community. The standards in their current version associate programming with modeling and define it as a process standard. For many, it was quite surprising to have reached a consensus on many of the issues.

The impact of this community of educators is being felt across the region.

The titles reflect the consensus decisions of the community. Content standards include: Information and Data, Algorithms, Languages and Automata, Informatics: Systems, and Informatics: Man and Society.

Process standards include: Model and Implement, Reason and Evaluate, Structure and Interrelate, Communicate and Cooperate, and Represent and Interpret.

It is important to understand the standards in the context of lower secondary education. In grade 5-7, to model and to implement means to be able to describe everyday processes and recipes as informal algorithms, including recipes for computer usage. In grade 8-10, algorithms should be formally defined and implemented.

Now that we have standards for computing education, what has happened?

Even though these standards were not issued by the government, they have had a positive effect on the community. Several initiatives have implemented the standards into teaching practice; e.g. learning tasks and instruments for evaluating the standards. There is a higher interest in context-based teaching approaches as a means to promote computing education for younger students. There is an initiative to construct and publish standards-based teaching examples.

The impact of this community of educators is being felt across the region. The standards are being read by state-based commissions defining curricula for German states, and have spread to other German-speaking countries such as Austria and Switzerland. The standards have even influenced the official German standards for teacher education in CS.

A Growing International CSTA Membership

Dave Burkhart

CSTA has always been an international organization but over the last year its international membership has increased significantly and India is playing a major role in CSTA's growing international presence.

Since June of 2008, CSTA's overall international membership has grown by 198 new members (20%) in 101 countries other than the United States. In the last six months, however, CSTA membership in India has increased by 116 new members (44%), making it both the fastest growing and the largest non-U.S. membership group. There are currently 1204 international members with 381 members in India.

Canada, the United Kingdom, and the Philippines round out the countries with the most international members.

Over the last year, CSTA has made a concerted effort to extend its focus beyond the U.S., producing an international version of *The New Educational Imperative: Improving High School Computer Science Education,* and involving our international volunteers in helping to make sure that all of our new publications are less U.S.-centric. We know, however, that CSTA's greatest international recruiting tool is members who share their knowledge about CSTA benefits with their colleagues.

Making a direct contact with non-CSTA members and encouraging them to join our organization is a great way to help build a strong community of educators dedicated to computer science. We applaud your efforts to help grow CSTA's membership and encourage you to continue to spread the word.

Meet the Authors

Stacey Armstrong
Cypress Woods HS, TX
Stacey teaches CS and is an AP Consultant and AP Reader. He also produces CS curriculum and contest materials.

Dave Burkhart
West Muskingum MS, OH
Dave teaches computer technology and is the K-8 Teacher Representative on the CSTA Board of Directors. He currently serves as the Chair of the Membership Committee.

Gail Chapman
Director, Leadership and Professional Development
Gail is the coordinator for the CSTA Leadership Cohort and its activities and also coordinates the TECS workshops.

Barbara Ericson
Georgia Institute of Technology
Barbara is the Director of Computing Outreach at Georgia Tech, and Teacher Education Representative on the CSTA Board of Directors.

Baker Franke
University of Chicago Laboratory Schools
Baker currently teaches AP CS and Advanced Topics in CS. Starting next year, he will teach 129 freshmen in the first year of Lab's brand new required CS course.

Dan Lewis
Santa Clara University, CA
Dan is the Principal Investigator on a collaborative multi-institution NSF grant headed by Steve Cooper and Wanda Dann to train HS and MS teachers to teach CS using Alice.

Felipe Payan
Los Angeles Southwest College
Felipe is an instructor in computer applications and office technologies at Los Angeles Southwest Community College and a lecturer at California State University - Los Angeles.

Carsten Schulte
Freie Universitaet Berlin, Germany
Carsten is a Professor for Computing Education Research and a member of the steering committee for establishing the German CS Educational Standards.

Committee News

Teacher Certification: A Call to Action

Barbara Ericson

Keeping talented computer science (CS) teachers in classrooms and ensuring that new teachers are adequately prepared to teach CS were the motivations for the CSTA teacher certification recommendations detailed in the recently published, *Ensuring Exemplary Teaching in an Essential Discipline* whitepaper.

The document contains specific recommendations for teacher preparation and certification in CS, and the rationale behind the recommendations. The report also points out the gap between current teacher preparation/certification programs and CSTA's recommendations. Teachers are urged to share the document with principals, state certification bodies, state teacher preparation programs, and state continuing education programs.

The executive summary provides an overview of the findings and recommendations. Chapters 1–3 provide a brief overview of the current issues in teacher certification, research on teacher preparation/certification, as well as details on the current CS teacher preparation/certification requirements in Pennsylvania, Georgia, Texas, Scotland, and Israel. Chapter 4 lists recommendations for new teachers and veteran teachers with various levels of experience and expertise, as well as for computing professionals entering teaching for the first time.

CSTA recommends that all CS teacher preparation and certification programs include:

1. Academic requirements in the field of CS
2. Academic requirements in the field of education
3. A methods course and field experience
4. Assessment to document proficiency in general pedagogy

The following are CSTA recommendations for current CS teachers who do not have a CS degree.

Veteran teachers WITH CS teaching experience

- Bachelor's degree or higher in a field other than CS
- Certification in an academic discipline other than CS
- Teaching Experience in an Advanced Placement CS course (or the equivalent) for at least two years, and/or
- Teaching International Baccalaureate HL CS (or the equivalent) for at least two years, and/or
- Teaching a rigorous introductory CS course (equivalent to the Level II course described in the ACM Model Curriculum for K–12 CS) for at least two years
- Academic Work and Field Experience (to be completed within 3 years) in CS can be documented by completing one of the following:
  - Completion of a minimum of 40 hours of professional development workshops designed for teachers of CS
  - Advanced coursework in programming, object-oriented design, data structures and algorithms, computer hardware and organization, or computer applications
- Methodology requirements documented by the completion of at least **one** of the following:
  - Completion of a minimum of 40 hours of professional development workshops designed for teachers of CS,
  - a methods course,
  - auditing of one complete K–12 CS course,
  - creating a portfolio that documents pedagogy in the CS classroom

Veteran teachers WITHOUT CS teaching experience

- Bachelor's degree or higher in a field other than CS
- Certification in an academic discipline other than CS
- Academic and Field Experience (to be completed within 3 years)
- Academic requirements in the field of CS include advanced coursework in the following areas: programming, object-oriented design, data structures and algorithms, computer hardware and organization, or computer applications
- Methodology requirements can be documented by the completion of at least **one of the following**:
  - Methods Course
  - Auditing of one complete K–12 computer science courses

The complete list of recommendations can be read in *Ensuring Exemplary Teaching in an Essential Discipline* (*csta.acm.org/Communications/sub/Documents.html)*.

SHOW ME THE NUMBERS
CSTA Sprouting New Chapters

Number of CSTA local chapters	**8**
Chapters with CSTA Branches (AZ, CA, IL, OR, NJ, NY, TX, VA)	**8**
States working to create chapters	**2**
States working on additional chapters	**3**

Learn more about organizing a CSTA Chapter in your area.

csta.acm.org/About/sub/CSTAChapters.html

Out and About the Community

Leadership Cohort Update

Gail Chapman

For the last nine months, our CSTA Leadership Cohort members have been revving up K-12 computer science (CS) in communities across the country by helping to establish new regional chapters.

Thirty-two leaders from 17 states began the work of forming new local chapters during the first Leadership Cohort Workshop in July 2008. Members of the Leadership Cohort have been instrumental in establishing new local chapters in Arizona, Buffalo, Chicago, Houston, Southeastern Virginia, Southern California, and Southern New Jersey Shore.

These chapters will provide a support network for CS teachers to share their ideas, plan outreach efforts and professional development, and work with local colleges and universities. The next Leadership Cohort will convene individuals from the remaining states this summer.

Leadership Cohort members have also been conducting a variety of activities throughout the country. These include:

- **Texas:** Local area CS contests and CS-related conventions are being used to build interest in chapters and to recruit members.
- **Maryland:** Meetings with the Director of Curriculum for the state have led to conversations with a variety of stakeholders regarding recognition of CS as a core discipline.
- **Silicon Valley and Oakland, California:** Meetings with superintendents and principals have raised awareness of the need for a more substantive CS curriculum.

- **North Carolina:** Meetings with the Department of Public Instruction are being held to discuss the state of CS in North Carolina.
- **Wisconsin:** A presentation to local businesses was used to raise awareness.
- **Oregon:** Three Fall SuperQuest conferences held in Portland, southern Oregon, and Salem showcased game design, robotics, and Gridworld. (*www.techstart.org/superquest.html*)
- **Ohio:** A CS strand was provided at the state educational technology conference.

Several members of the Leadership Cohort also participated in the Rebooting Computing Summit held in Mountain View, California, January 12-14, 2009, to tackle declining enrollments in CS education and the resulting career issues. The Rebooting Computing Summit is sponsored by the National Science Foundation. Dr. Peter Denning of the Naval Postgraduate School was instrumental in organizing the event.

The Leadership Cohort is the topic of the CS & IT Symposium panel discussion, Building Effective Leadership at the Grass Roots, in Washington, DC, on June 27. Keep up-to-date on cohort activities through the Advocate blog (*blog.acm.org/csta/*).

Curriculum in Action

Teaching CS with Scratch

Stacey Armstrong

Many beginning computer science (CS) students find text-based programming environments frustrating as they try to learn programming concepts at the same time they must learn code and syntax rules. Scratch is an interactive programming environment that enables students to learn fundamental concepts without requiring the added complications of code and syntax.

Students connect commands together to make things happen in the drag-and-drop Scratch environment. They learn solid programming concepts, including looping, decision-making, and variables, within the context of animation, storytelling, music, art, and games. Once these fundamentals are in place, it is much easier to transition students to text-based environments.

In Scratch, students create programs by combining various program steps from command groups including motion, control, pen, looks, sound, and sensing.

The control command, "when green flag clicked," teaches students basic decision making logic. If a "looks command" is attached, the object visibly reacts when the user clicks on the green flag during the execution of the program.

Among the motion commands, the "move X steps" command makes objects move on the screen. The "when key pressed" command can be connected to the "move X steps" command to make an object move X number of steps each time a particular key is pressed.

I use Scratch to teach CS concepts to elementary, middle school, and high school students. Scratch is valuable at all of these levels when assignments are designed to fit students' interests and levels of CS knowledge. The online gallery of projects demonstrates the sophistication and variety of projects that can be achieved with Scratch. Scratch is developed by the Lifelong Kindergarten group at the MIT Media Lab, with financial support from the National Science Foundation, Microsoft, Intel Foundation, Nokia, and MIT Media Lab research consortia (*scratch.mit.edu/*).

Introductory lab activity: Make the Scratch cat or some other object turn 90 degrees and move 90 steps each time the user clicks on the object with the mouse. After moving the object in a square, add the "pen down" command to make the object draw a box. Experiment by changing the angle to create other shapes. Change the number of steps to make the shape larger or smaller. Add creative movement by adding more commands.

Letter to the Editor

Felipe Payan

Dear Editor,

I am a computer applications instructor at the Los Angeles Southwest Community College where I teach Office 2007 and keyboarding. After reading the article in the Voice about Jane Margolis's book, *Stuck in the Shallow End*, I purchased the book while visiting family in Mexico. The findings by Margolis and her team of researchers about the lack of computer science (CS) education directed toward African-American and Latino students are unfortunately true in south and southeast Los Angeles.

I grew up in southeast Los Angeles and easily recognized one of the secondary schools she researched for the book. As a computer teacher in private and public schools for the past eight years, I understand why there is a lack of Black and Latino students in CS or technology fields. The issues revealed by the Margolis, such as low expectations, language issues, and dearth of positive Latino role models in technology, aside from Richard Tapia, are true. African-American role models in technology are scarce; the head of Symantec is the only African-American I know of to lead a major technology company.

Like the students in Jane's book, I received a math education that was sufficient for working in the many factories that surrounded my community. Today, those factories are gone.

We need fresh solutions to attract Latino and African-American middle and high school students into emerging technologies. Educators should not have to "dumb down" the curriculum or relegate minority students to keyboarding. Minority students should be offered challenging CS coursework that will stimulate their interest in technology careers. As CS educators and administrators we have to encourage curiosity in minority students and display enthusiasm for the opportunities in technology. We must promote the idea that high-level technology skills are for everybody.

CSTA MEMBER IN THE NEWS: 2008 PUBLISHERS AWARD

Congratulations, JANE MARGOLIS!

Her book, *Stuck in the Shallow End* (MIT Press), received the 2008 Association of American Publishers Award for Professional and Scholarly Excellence (The PROSE Awards) in the Education category from the Professional and Scholarly Publishing (PSP) Division of the Association of American Publishers (AAP).

The book, coauthored by Rachel Estrella, Joanna Goode (a member of the CSTA Board of Directors), Jennifer Jellison Holme, and Kimberly Nao, examines computer science experiences of students and teachers in three Los Angeles public high schools and reports upon the research revealing an insidious "virtual segregation" that maintains inequality for African Americans and Latino students.

Margolis is Senior Researcher at the Institute for Democracy, Education, and Access at UCLA's Graduate School of Education and Information Studies (*www.publishers.org/02_ProseWinners.htm*).

csta Voice
We're on the Web! *csta.acm.org*
Computer Science Teachers Association
2 Penn Plaza, Suite 701
New York, NY 10121-0701

MARK YOUR CALENDAR

UAB High School Programming Contest (HSPC)
May 10, 2008 in Birmingham, Alabama
www.cis.uab.edu/programs/hspc/

Oregon Game Programming Challenge (OGPC)
May 16, 2009 in Salem, Oregon
techstart.org/ogpc

UAB Alice Festival
May 16, 2009 in Birmingham, Alabama
www.cis.uab.edu/programs/alice-festival

ACSL All-Star Contest
May 23, 2009 in Huntsville, Alabama
www.acsl.org

School Programming summer camps
June-July 2009 in Birmingham, Alabama
www.cis.uab.edu/programs/camps

Computer Science and Super Computing Workshop
June 11, 2009 in Union, New Jersey
www.kean.edu/~cssc

Computer Science and Information Technology Symposium (CS & IT)
June 27, 2009 in Washington, D.C.
www.csitsymposium.org

NECC 2009
June 28-July 1, 2009 in Washington, DC
center.uoregon.edu/ISTE/NECC2009

TECS Workshop CS4HS 2009
July 24-27, 2009 at Carnegie Mellon University
tecs.acm.org/public/TECS/workshops

Alice Educator Workshop
July 13-17 & July 20-24, 2009 in Santa Clara, California
www.scu.edu/engineering/cse/outreach/workshop.cfm

The 21st International Olympiad in Informatics
August 8-15, 2009 in Plovdiv, Bulgaria
ioinformatics.org

Consortium for Computing Sciences in Colleges (CCSC: Northwestern)
October 9-10, 2009 in Parkland, Washington
www.ccsc.org/northwest/2009

Consortium for Computing Sciences in Colleges (CCSC: Midwestern)
October 9-10, 2009 in Chicago, Illinois
www.ccsc.org/midwest

Consortium for Computing Sciences in Colleges (CCSC: Rocky Mountain)
October 16-17, 2009 in Farmington, New Mexico
www.ccsc.org/rockymt

Consortium for Computing Sciences in Colleges (CCSC: Eastern)
October 30-31, 2009 in Villanova, Pennsylvania
ccsce09.villanova.edu

Consortium for Computing Sciences in Colleges (CCSC: Southeastern)
November 13-14, 2009 in Salem, Virginia
www.ccscne.org/2009

CSTA INSTITUTIONAL MEMBER K–12 OUTREACH PROGRAMS

TechTopia Challenge - Neumont University
July 31, 2009 in Jordan, Utah
www.tech-topia.com

RESOURCES

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

Page 1: Alice *www.alice.org/*
Page 1: Media Computation Teachers Website *coweb.cc.gatech.edu/mediaComp-teach*
Page 1: IEEE *www.ieee.org/*
Page 1: Association for Computing Machinery (ACM) *www.acm.org/*
Page 2: *CSReq.org* (making CS a HS graduation requirement) *csreq.org*
Page 2: University of Chicago Laboratory Schools *www.ucls.uchicago.edu/*
Page 2: National Council of Teachers of Mathematics (NCTM) *www.nctm.org*
Page 3: CS & IT Symposium *www.csitsymposium.org*
Page 3: Education CS standards (in German) *www.informatikstandards.de/*
Page 5: Educational Imperative (International) *csta.acm.org/Communications/sub/Documents.html*
Page 6: Association of American Publishers *www.publishers.org/index.html*
Page 6: Jane Margolis *www.gseis.ucla.edu/faculty/members/margolis*
Page 6: Los Angeles Southwest College *www.lasc.edu/*
Page 6: Ensuring Exemplary Teaching *csta.acm.org/Communications/sub/Documents.html*
Page 6: SuperQuest *www.techstart.org/superquest.html*
Page 6: CSTA Leadership Cohort *csta.acm.org/About/sub/LeadershipCohort.html*
Page 7: Scratch *scratch.mit.edu/*
Page 7: A+ Computer Science *www.apluscompsci.com/*
Page 8: Consortium for Computing Sciences in Colleges (CCSC) *www.ccsc.org/index.htm*