csta Voice

10,000 Teachers

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

Volume 5, Issue 6

January 2010

Inside This Issue

IN THE NEXT ISSUE OF THE VOICE

International CS Education

Finding 10,000 Teachers

Transforming High School Computer Science

Jan Cuny

THE CISE (Computer & Information Science & Engineering) Directorate of the National Science Foundation (NSF) proposes to catalyze a clean-slate revamp of high school computing education through the new CS/10K Project. The goal of the CS/10K Project is to develop an effective high school curriculum that will be taught by 10,000 well-prepared teachers in 10,000 high schools by 2015.

The ability to effectively apply computation to solve complex problems is a critical 21st Century skill, yet too few of our students receive rigorous formal education in computing. The CS/10K Project focuses on computational skills at the high school level because engagement programs for younger students will be ineffective if students have no further opportunities to explore computing in high school, nor the chance to discover the exciting opportunities computing careers offer. Likewise, revitalized college computing programs will not have a significant impact on degree production if there are too few students showing up at their doors. And currently, there are too few students. Since 2000, the percentage of incoming college freshman who intend to major in computing has declined more than 70%; for women, the figure is closer to 80%.

High schools are a critical link in filling the pipeline to the university level; yet few high schools offer courses in computing beyond basic literacy. Often, the courses that are available are taught under Career and Technical Education (CTE) and, thus, are rarely taken by college-bound students. An exception is the Advanced Placement Computer Science (AP CS) A course. Where AP CS is available, it typically carries college preparatory status, but it is offered in less than 10% of U.S. high schools.

The new curriculum will feature two courses: an introductory course, followed by an entirely redesigned AP CS course. Both will be rigorous and engaging, will develop computational thinking skills, and will expose students to the breadth of computational applications or the "magic" of computing. The introductory course will be designed for both college-preparatory and CTE students. The new AP CS course will be designed for more advanced students with a broad range of interests including: CS, science, engineering, business, communication, and the arts. This new AP course will provide a single point of national leverage. The College Board will continue to offer the existing AP CS A test, and its corresponding course can be an appropriate third course for some students.

Curriculum development for the two new courses is underway. The College Board, with funding from the NSF, is developing a framework for the AP course. It will not be programming-centric, but instead will focus on the underlying principles of computation involving problem-solving, abstraction, algorithms, data and knowledge creation, as well as

CSTA
wishes to thank
Debra J. Richardson

for her dedicated work as
Chair of the CSTA Advisory Council.

Executive Officers

Michelle Hutton
President
mfh@pobox.com

Stephen Cooper
Vice-President
scooper@sju.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

Grants
cstagrants@csta.acm.org

International
cstainternational@csta.acm.org

Membership
cstahelp@csta.acm.org

Professional Development
cstapd@csta.acm.org

Research
cstaresearch@csta.acm.org

FINDING 10,000 TEACHERS

continued from page 1

programming. The course will explore the limitations of computation, the breadth of computational applications, and related societal technology issues impacting culture and ethics. It will be relevant, inspiring, and rigorous. It will serve as a target for K–9 curricula development and an impetus for revising introductory college courses.

The curriculum for the introductory course will be flexible, allowing teachers to define the scope and sequence of a class within a range of topics and activities. Teachers will have access to a curricular framework, instructional materials, and a set of exemplars. One of those exemplars is the new *Exploring Computer Science (ECS)* course, developed with NSF funding under the direction of Jane Margolis, Todd Ullah, Joanna Goode, and Gail Chapman. *ECS* is currently taught in 20 schools in the Los Angeles Unified School District, and it has just been granted both College Preparatory ("G") and CTE credit status in California. The complete curriculum is available on the CSTA website at csta.acm.org/Curriculum/sub/ExploringCS.html.

Gaining access to an innovative curriculum, however, is not the most daunting challenge. The greatest challenge will be scaling the teacher preparation opportunities and support to reach 10,000 computing teachers by 2015. Too few high schools have computing teachers with formal CS training and many have no computing teacher at all. The CS/10K Project will launch an unprecedented teacher preparation program, working with in-service as well as pre-service teachers, in both traditional and alternative certification programs. It will pair face-to-face training with extensive, state-of-the-art online support that includes curricula, materials, and social networking. We will work to establish high-quality, ongoing professional development programs. To accomplish this, we will build a public-private partnership for funding and we will need to call on the time and energy of university faculty, undergraduate and graduate students in service learning programs, and professionals serving as citizen scientists. Most of all, we will need to call on K–12 teachers who are interested in this exciting new initiative.

Help CSTA to get computing included in discussions on certification, standards, and course classifications in your state.

Help us develop professional development plans that will work. Help CSTA to get computing included in discussions on certification, standards, and course classifications in your state. Help us get these courses to your schools, and to your students.

Please send advice, comments, and suggestions to *jcuny@nsf.gov*, and let me know if you would like to be on our mailing list for updates on the CS/10K effort.

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.

A New AP Course on the Horizon

Reaching 10,000 Schools

Chris Stephenson

WHEN THE COLLEGE BOARD announced that it was discontinuing the Advanced Placement Computer Science (AP CS) AB exam, many of us were deeply disheartened. And while we recognized that the decision was taken because too few students were writing the exam to make it economically viable, our community was shocked and shaken.

As is sometimes the case, however, catastrophe has led to introspection, inspiration, and intervention. In light of this set-back, the National Science Foundation stepped in with an audacious new strategy to reinvigorate high school CS education by creating the CS/10K Project aimed at getting AP CS into 10,000 schools. Currently there are only 2,312 schools in the country authorized by the College Board to offer the AP CS A course. This is by far the lowest number of schools for any of the sciences or mathematics.

A primary component of the CS/10K Project is a new AP CS course, currently being called the AP CS Principles course, that will be offered in addition to the current AP CS A exam. This new course represents a re-imagining of AP CS as a "gold standard high school course," similar to innovations undertaken by other AP science courses. The goal is to provide students with a new course that will be both rigorous and engaging and that will give them a better understanding of what respected Stanford CS educator Eric Roberts refers to as "the beauty and awe" of computing.

Teachers will have to step up and get prepared to teach this new course.

The work of developing this course falls to the AP CS: Principles Commission, of which I am a member. In addition to the several committee members, we have also been fortunate to have the guidance and support of a large number of CS luminaries and thought leaders who are serving as an Advisory Group for the project and with whom we meet to discuss major project milestones.

Being part of this commission demands a significant allocation of time (and a great deal of airplane travel) but it also brings the opportunity to contribute to the much more challenging task of rebuilding the CS pipeline by creating more and better opportunities for students to learn and explore this exciting field.

Despite the work of the National Science Foundation, the AP CS: Principles Commission, the AP CS: Principles Advisory Group, and the AP CS Development Commission, who are continuing to maintain and improve the AP CS A exam, the CS/10K Project will not succeed without a much larger commitment of expertise and resources. It will require the entire community.

Teachers will have to step up and get prepared to teach this new course. Schools district and state policy makers will have to make a profound commitment to getting an AP CS course into their schools, colleges and universities will have to form critical partnerships to provide wide-scale accessible professional development for current teachers and relevant preparation for in-service teachers, and college and university admissions officers will have to stand behind this course by providing credit and placement for students who successfully complete the AP CS exams. Business and industry (the ultimate recipients of the benefits of a highly skilled CS workforce) are also going to have to put some cold hard cash on the line to support all of these activities.

The time for hand-wringing about the pipeline is over. As a community, we know that we must begin now to build a critical, comprehensive, cohesive partnership to change the course of CS education and make sure that all of our students have the opportunity to be ready for the future. To fail at this is to lose one of the best opportunities our discipline has had to make a profound contribution to the future.

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

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

National Science Foundation Grants

Power in Partnering

Steve Cooper

Editor's note: *The author of this article served as a program manager within the National Science Foundation (NSF) for two years. The views and opinions expressed in this article are his alone, and do not constitute official policy or positions of NSF.*

FEW K–12 SCHOOL DISTRICTS or schools partner with their local colleges. Even fewer partner with their local colleges to apply for grants. K–12 school district-university partnerships, however, can be ideal for many grant proposal opportunities including NSF grant programs.

NSF is divided into research directorates which accept grant proposals according to the focus of the request. The Computer & Information Science & Engineering (CISE) directorate funds computer science (CS) research projects. There are two programs within CISE: Broadening Participation in Computing (BPC) and CISE Pathways to Revitalized Undergraduate Computing Education (CPATH). These programs are especially appropriate for partnerships between K–12 and higher education. An existing NSF-funded project can also support a K–12 teacher through funding available through the Research Experience for Teachers (RET) program.

It is likely that the best place to begin the search for possible funding is from NSF's Education and Human Resources directorate. This directorate is divided into four divisions: the Division of Graduate Education (DGE), the Division of Undergraduate Education (DUE), the Division of Human Resource Development (HRD), and the Division for Research on Learning in formal and informal settings (DRL). The latter three divisions all have programs that offer possibilities for K–12 and higher education partnerships.

DRL focuses on K–12 Science, Technology, Engineering, and Mathematics (STEM) education. Two DRL programs jump out as particularly good for K–12 educators to consider when first seeking funding: Innovative Technology Experiences for Students and Teachers (ITEST) and Discovery Research K–12 (DR K–12).

Suggestions for K–12 and higher education grant seeking partnerships:

1. Have a good idea for a project, rather than simply a need for equipment. NSF grants are primarily about projects and less focused on equipment acquisition. Some NSF grant programs allow for the purchase of equipment as part of the project, but equipment purchase tends not to be the focus. NSF proposals must address (and are graded against) two criteria: intellectual merit and broader impact. In educational proposals, broader impact frequently has as an important component for the possibility of replication of the project at other institutions. If the grant submitted is only (or primarily) for the purchase of equipment, the impact is likely going to be local, with little chance of replication or impact elsewhere.
2. Investigate recently funded programs. Learn about successful projects and gather ideas for submitting your own. On the NSF website you can search for lists of all awards made by a given program. Included in this data are the names and contact information of the grant principal investigators (PIs), along with a 250-word abstract about their project. Contact the PIs or Leslie Jensen (*ljensen@nsf.gov*) at NSF to request a copy of the successful proposals.
3. Find colleagues at higher education institutions who are interested in partnering with you. Faculty members at nearly all institutes of higher education are feeling increasing pressure to write grant proposals and to win awards. If partnering with a K–12 institution will increase their chances of getting a grant, they will be especially motivated to form a partnership. Ask the chair of the computing department at a nearby college if there are faculty members who might be interested in partnering with you. K–12 teachers can also meet potential college partners at one of the ACM's 10 annual regional Consortium for Computing Sciences in Colleges (CCSC) conferences,

which are regularly listed in the Voice calendar. I have found the CCSCs to be a great opportunity to talk with faculty at area colleges about potential education and research projects.

4. Talk to a program officer at NSF. Program officers can tell a potential grant proposer how well an idea matches what is being looked for by a given NSF program, and may even suggest a different program more suited to the proposer's idea.
5. Offer to review proposals for the next NSF program solicitation. NSF operates on a peer-review process, where reviewers play an important role in helping to inform NSF about which proposals to fund. I have personally found that reviewing proposals is probably the single best professional development activity I have ever done in learning how to write NSF grant proposals.

Receiving an NSF grant is a competitive process. Many great project ideas don't get funded simply because of the unavailability of sufficient money. However, if you have a great idea for enhancing CS education at your school but no resources, work to partner on a grant proposal with a colleague or two at a neighboring university. It could be a win-win situation for all.

SIGCSE Benefits for CSTA Members

Barbara Boucher Owens

YOU LIKELY KNOW THAT SIGCSE (Special Interest Group in Computer Science Education) and CSTA are both branches of ACM. What you might not know, especially if you are not a university or college faculty member, is that SIGCSE offers benefits for CSTA members.

SIGCSE is a dues-paying ($25 annually) organization that has been effecting change and supporting faculty in computer science (CS) education for over 40 years.

Each year, SIGCSE hosts the annual SIGCSE Technical Symposium at which about 1200 computing educators, from K–12 teachers through university faculty gather to hear stimulating keynote presentations and attend workshops on teaching techniques as well as find a supportive community of like-minded teachers. Over the last few years, the SIGCSE Conference organizers have made substantial changes to make the conference more attractive and interesting to K–12 teachers, including a special one-day conference registration fee just for teachers. This year, the SIGCSE conference will be held in Milwaukee, WI, March 10–13, 2010 (*www.sigcse.org/sigcse2010*).

CSTA members have access to the proceedings of SIGCSE's conferences through the ACM Digital Library at *portal.acm.org/dl.cfm*. (You must be an ACM member with Digital Library access in order to download the articles.)

SIGCSE also produces a member publication called *inroads* that will debut as a classy magazine in mid-2010. CSTA members can view abstracts of *inroads* articles through the ACM Digital Library.

SIGCSE also provides grants and outreach programs. Through the program labeled "Outreach", SIGCSE supports the dissemination of outstanding SIGCSE presentations to other venues. The speaker supply program is open to any non-ACM conference holding "In-Cooperation" status with SIGCSE. Conferences designed for K–12 CS educators may qualify for "In-Cooperation" status, and organizers can request speaker support from SIGCSE.

A second opportunity available to SIGCSE members is the "Special Projects Grant" program. SIGCSE funds a limited number of Special Projects to support members who wish to introduce new ideas for learning and teaching computing. SIGCSE members may apply for grants from the Special Projects fund to a maximum funding of $5,000. The level of funding is subject to the quality of proposals received and the availability of funds. (*www.sigcse.org/projects/special-project-grants*)

To find out more about SIGCSE, visit SIGCSE at *www.sigcse.org* or contact *chair@sigcse.org*.

Meet the Authors

Steve Cooper
Purdue University, IN
Steve is a professor of computer graphics technology. From 2007–2009, he was a program officer at the National Science Foundation's (NSF) Education and Human Resources directorate, and he serves as Vice President on the CSTA Board of Directors.

Jan Cuny
National Science Foundation
Jan is a program officer at the NSF where she runs the Broadening Participation in Computing program. She also leads the CS/10K Project.

Irene A. Lee
Santa Fe Institute
Irene is the program director and principal investigator of Project GUTS (Growing Up Thinking Scientifically), an NSF-funded program that engages students in the computational modeling of locally relevant issues as complex systems.

Robert Luciano
Computer Science Educator
Robert taught math for several years and now teaches CS full time at Pocono Mountain East HS in Swiftwater, PA.

Barbara Boucher Owens
Southwestern University, TX
Barbara has been in the computing field since 1967. She is chair of the ACM Special Interest Group on Computer Science Education (ACM/SIGCSE) and heads an NSF-funded project to collect oral histories of computing educators.

Chris Stephenson
Executive Director, CSTA
Chris has been the Executive Director since it began in 2005. She joined ACM after 16 years at the University of Toronto's Computer Systems Research Institute and the University of Waterloo's Mathematics and Computing Department, where she designed instructional and professional development resources.

Out and About the Community

Fostering Computational Thinking

Project GUTS

Irene A. Lee

Santa Fe Institute's Project GUTS (Growing Up Thinking Scientifically) is a three year-old STEM program where computational thinking is much more than using computers and software tools, coding and debugging; it is the design, creation, use, and analysis of computational models to study complex systems.

There is an urgent need to understand large complex systems to address the problems of the 21st Century such as climate change, loss of biodiversity, energy consumption, and virulent disease. Computational thinking and computing will be an important part of all sciences in the 21st Century because computational methods and tools can be used to study, model, and understand these complex systems problems.

Project GUTS was developed by researchers and educators who noticed that middle school students and teachers were increasingly interested in creating computational science projects but often lacked an idea of where to start. They also observed that students from underrepresented groups in STEM were attracted to problems that related to their communities and were on a "human" scale; students who did not consider themselves "young scientists" or "future scientists" were interested in computational modeling and computational science when they saw a connection and benefit to their community.

Project GUTS fosters many aspects of computational thinking, including:

- Understanding that models are used in various scientific domains
- Conducting experiments by changing variables in models
- Collecting and analyzing data from experiments

Within a Project GUTS unit, students focus on an issue or phenomenon in their community. They investigate it as a complex system using agent-based models they create in StarLogo TNG. For example, in an investigation of pollution, students collected data on traffic in front of their school, including the number of vehicles passing by per minute and the density of vehicles. They also collected air samples from car exhaust and analyzed it for carbon dioxide levels. The group decided to determine whether the number of trees in front of the school could mitigate the pollution and help to maintain acceptable CO_2 levels even as traffic flow fluctuated.

To answer this question, the students customized an existing computer model of traffic flow to create a new model that included pollution being emitted from cars. The data collected in real-life was incorporated into the model by matching the initial density and speed of traffic and the dispersion of pollutants from the vehicles. Finally, trees that absorbed pollutants were added to the model and experiments were run using the model as an experimental test bed. At the conclusion of the investigation, students pondered how many additional trees would be necessary to mitigate the pollution.

In this way, the GUTS activity allowed the students to experience the computational thinking that occurs during the analysis and customization of existing models to reflect local conditions, including:

- Understanding of the conceptual model underlying the computer model
- Deconstructing models into agents with behavior and state, the environment in the artificial world, and interactions between agents and/or agents and the environment
- Developing expertise in evaluating models (What simplifications and assumptions were made and why?)
- Developing decoding skills and sustained reasoning (reading code, understanding the iterative loop and the execution of code)

Project GUTS was funded by the National Science Foundation, the Bengier Foundation, and others. Learn more about Project GUTS at *www.projectguts.org*.

Curriculum in Action

Using Cooperative Logic Groups in Computer Science

Robert Luciano

Cooperative logic groups are the ideal tool for developing a variety of skills in computer science (CS) students. While teachers recognize that problem solving, logical thinking, communication, and social skills are all necessary to succeed in CS, it is sometimes difficult to plan classroom activities to develop these skills.

Groups of three to four students working together to solve an interesting problem provide an ideal scenario for initiating cooperative logic activities. The problem should not require intricate math or academic knowledge. Logic and reasoning should be the skills necessary to find the solution. Logic problems have been around for centuries and most people enjoy the challenge of solving a thought provoking problem.

Requiring students to spend the majority of their time writing and testing code on a computer reinforces the misconception that CS is a solitary pursuit. Intricate programs are built by teams of people who are constantly meeting, discussing, and formulating algorithms in order to write successful programs. Cooperative logic groups provide the structure for students to collaborate to solve problems. Communication skills are enhanced because group members must share ideas and explain their thought processes and strategies. This lays the groundwork for students checking their work and validating solutions with classmates.

Over time, students working in cooperative groups learn that discussing problems with other students can provide insights into solving problems and reduce frustration. Instead of students giving up and staring at their screens without talking, the classroom becomes an assembly of learners interacting with each other.

Mixing groups based upon personality, gender, and skills will ensure support and modeling for others. Working with students who draw, find patterns, and formulate algorithms helps students with less ability to develop these skills. Because cooperation is a key to the success of these groups, students are able to contribute their unique skills toward the common goal. A group with at least two students from any minority group within the class (gender or ethnic) will provide a comfortable environment for those individuals.

Cooperative logic groups are enjoyable for students and teach skills that make people successful in any subject. And importantly, students in cooperative logic groups learn without realizing they are learning.

Here is a fun logic problem for cooperative logic groups to begin with:
A snail begins climbing up from the bottom of a thirty-foot deep well. The snail can climb up three feet each hour. However, the snail gets so tired from the effort that at the end of the hour it slips down two feet. How long will it take the snail to get out of the well?

Answer:
The answer is 28 hours. The snail climbs one foot per hour. After 27 hours the snail is 27 feet up the well wall. During the next hour the snail climbs up 3 feet and exits the well. It does not slide down 2 feet at the end.

College Connection

Santa Clara University

Editor's note: *This dialog with Dan Lewis of the Computer Engineering Department at Santa Clara University is a continuation of our series of interviews with CSTA institutional members. Please share with your students these details about the computer science (CS) programs at Santa Clara University.*

Santa Clara University is a comprehensive Catholic, Jesuit university that provides a values-oriented curriculum to about 4,600 under-graduate students and 3,300 graduate students. For the 20th consecutive year, Santa Clara was ranked second among Master's universities in the West in *U.S. News & World Report*. Students in the Department of Computer Engineering are offered two BS programs in Computer Science and Engineering (BSCSE) and Web Design and Engineering (BSWDE), as well as MS and PhD programs in Computer Science and Engineering, and an MS program in Software Engineering.

CSTA: What draws students to your program and what keeps them there?
Lewis: Santa Clara University is known for its quality academic programs, small classes taught by full-time faculty, personalized attention, and a beautiful campus located in the midst of Silicon Valley. Undergraduates in computing acquire extensive "hands-on" experience through our laboratory-intensive curriculum and project-based courses, and have excellent opportunities to participate in faculty research and for co-op work assignments in the local high-tech industry.

CSTA: What skills can students acquire before college that will help them succeed in your program?
Lewis: No prior courses in computing are necessary, but strong analytical and quantitative skills are essential. Students who did well with courses that require logical thinking (like geometry) usually also do well in computing.

CSTA: What cool careers are your graduates prepared for?
Lewis: Our graduates often take jobs in Silicon Valley at such companies as Apple, Cisco, HP, IBM, Sun, or one of the many startups, while those interested in graduate school have been admitted to schools such as Johns-Hopkins, Carnegie-Mellon, Stanford, UC Santa Cruz, Colorado, Cornell, and Georgia Tech. With their understanding of the relationship between the Web and society, our BSWDE graduates are not only prepared for technical jobs in information architecture and databases, network engineering, user interface design, and overall system design, but also for positions in online journalism, online commerce, social and/or political networking, or Internet marketing.

CSTA: What topics will students study?
Lewis: All Santa Clara students follow a common university core—a collection of courses designed to prepare students to become thoughtful, ethical leaders in our fast-paced technological world. BSCSE students study software and hardware as well as the mathematical foundations of computation, with concentration options in information assurance, game development, robotics, and Web technologies. The BSWDE combines computing with graphic art and Web-related courses in communication and sociology to produce graduates who can analyze, design, and improve the computational infrastructure of the Web. These graduates can develop interactive multimedia content that is appealing, engaging, effective and easy to use, and which is guided by an understanding of and sensitivity to the social, political, ethical, and legal relationships between their work and those that it affects.

CSTA: Tell us a bit about the social environment of the program.
Lewis: A strong "esprit de corps" exists among our students, fostered through an active student ACM chapter and a number of departmental activities attended by both faculty and students, including a freshman lunch series, field trips to and guest speakers from Silicon Valley companies, a senior design contest and dinner, and the end-of-year barbeque. Students enjoy the San Francisco Bay Area's Mediterranean climate, and often spend their weekends on the beach or skiing at Lake Tahoe.

CSTA: What distinguishes your school and program from others?
Lewis: Santa Clara University's Jesuit commitment to educating the whole person in an environment that is academically excellent and ethically oriented sets us apart. Our students enjoy small class sizes of about 20 students, close personal attention due to a student-faculty ratio of 12:1, and unusual opportunities made possible by our proximity to Silicon Valley industry. In addition, Santa Clara's BSCSE and BSWDE programs offer unique options within a regular four-year program for a one-quarter study abroad, a six-month co-op work experience, or a head start on a graduate degree through our combined five-year BS/MS option.

More information is available at *www.scu.edu/engineering/cse.*

SHOW ME THE NUMBERS

Bachelor Degrees awarded in 2007

Business	327,531
Social sciences and history	164,183
Education	105,641
Health professions and related clinical sciences	101,810
Psychology	90,039
Biological and biomedical sciences	75,151
Engineering	67,092
Computer and information sciences	42,170

Source: U.S. Department of Education,
National Center for Education Statistics
nces.ed.gov/programs/digest/d08/tables/dt08_271.asp

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

FETC Florida Educational Technology Conference
January 12–15, 2010 in Orlando, Florida
www.fetc.org

ISSEP (Int. Conference on Informatics in Secondary Schools)
January 13–16, 2010 in Zurich, Switzerland
www.issep2010.org

TCEA (Texas Computer Education Association)
February 8–12, 2010 in Austin, Texas
www.tcea.org/convention/2010

SIGCSE Technical Symposium
March 10–13, 2010 in Milwaukee, Wisconsin
www.sigcse.org/sigcse2010

Consortium for Computing Sciences in Colleges (CCSC: Southwestern)
March 26–27, 2010 in Thousand Oaks, California
www.ccsc.org/southwestern

Consortium for Computing Sciences in Colleges (CCSC: Mid-South)
March 26–27, 2010 in Searcy, Arkansas
www.ccsc-ms.org

Consortium for Computing Sciences in Colleges (CCSC: Central Plains)
April 9–10, 2010 in Parkville, Missouri
www.ccsc.org/centralplains

Consortium for Computing Sciences in Colleges (CCSC: Northeastern)
April 16–17, 2010 in West Hartford, Connecticut
uhaweb.hartford.edu/ccscne

Consortium for Computing Sciences in Colleges (CCSC: South Central)
April 23–24, 2010 in Austin, Texas
www.sci.tamucc.edu/ccsc

ISTE 2010 (International Society for Technology in Education) (formerly known as NECC)
June 27–30, 2010 in Denver, Colorado
center.uoregon.edu/ISTE/2010

Project GUTS / Supercomputing Challenge Summer Teacher Institute
July 11–24, 2010 in Socorro, New Mexico
challenge.nm.us/STI

CS & IT Symposium
July 13, 2010 at Google Headquarters,
Mountain View, California
www.csitsymposium.org

RESOURCES

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

Page 1: CS/10K *10000teachers.org*

Page 1: Broadening Participation in Computing *www.bpcportal.org*

Page 1: CSTA *csta.acm.org*

Page 1: Exploring Computer Science curriculum *csta.acm.org/Curriculum/sub/ExploringCS.html*

Page 3: College Board AP Central *apcentral.collegeboard.com*

Page 4: National Science Foundation *nsf.org*

Page 4: Consortium for Computing Sciences in Colleges *www.ccsc.org*

Page 5: ACM *www.acm.org*

Page 5: SIGCSE Outreach *www.sigcse.org/projects/outreach*

Page 5: SIGCSE 2010 Conference *www.sigcse.org/sigcse2010*

Page 5: SIGCSE Special Projects Grants *www.sigcse.org/projects/special-project-grants*

Page 5: ACM Digital Library *portal.acm.org/dl.cfm*

Page 6: Project GUTS *www.projectguts.org*

Page 6: StarLogo TNG *education.mit.edu/drupal/starlogo-tng*

Page 6: Santa Fe Institute *www.santafe.edu*

Page 6: Supercomputing Challenge *challenge.nm.org*

Page 7: Santa Clara University *www.scu.edu/engineering/cse*