IN THE NEXT ISSUE OF THE VOICE

Mentoring

Incorporating Culture into CS Projects

Valerie Taylor, Ron Eglash, Juan Gilbert

The inclusion of minorities in science and engineering occupations is important for fostering a diversity of views in technological development, in addressing racial income disparity, and in ensuring a democratic culture in which technological literacy is spread across all communities. Hence, it is critical that we maintain students' interests, especially underrepresented minority students, in math and computing courses throughout K-12 education, as well as higher education.

Computing and information technologies are widely used in a multitude of career areas. According to statistics in *Tomorrow's Job*, published by the U.S. Department of Labor, close to three-quarters of the projected job growth for the time period of 2006 through 2016 will come from three groups of professional occupations, with the top group being computer and mathematical occupations. This factor, combined with the projection from the U.S. Census Bureau that underrepresented minorities are projected to constitute almost 32% of the American population by 2020, thus outnumbering White males (30.1%), underscores the importance of insuring the diversity of the students interested in computer and mathematical occupations.

Research has illustrated a relationship between faculty practices and student engagement. Findings suggest that students report higher levels of engagement and learning at institutions where faculty members use active and collaborative learning techniques, engage students in experiences, emphasize higher-order cognitive activities in the classroom, interact with students, challenge students academically, and value enriching educational experiences. While this work is focused on the higher education community, we believe that the results are applicable to the pre-collegiate level as well.

One way to achieve this engagement and foster inclusion is through problem-based learning (PBL). Key characteristics of PBL include:

- Ill-structured, complex problems that provide the focal points and stimuli for the course or ideas originated by students that can be developed during the learning process;
- Learning is student-centered with teachers acting primarily in the role of facilitators;
- Faculty act as a coach or facilitator as students assume greater responsibility for their own learning;
- Students work in small groups to solve/provide multiple solutions to problems.

It is important that students be able to relate to project problems based upon their experiences. We note that this is a hard task to achieve when one must take into consideration the unique aspects of each culture represented in a classroom.

Even the first program that students create should take into account the demographics of the students. This is important as the first task in a course is considered indicative of future tasks for the course. The classic "Hello World" first program

Voice

CULTURE INTO CS PROJECTS
continued from page 1

can be modified to allow for differences, give students some ownership of the task, and provide an experience that better reflects the uniqueness of each student. This approach can be applied to more complex projects throughout the course.

There are many ways to open CS opportunities for diverse expressions, identities, and unexpected connections. Readers who would like to learn more about cultural approaches to computing might want to query the phrase "Ethnocomputing," a growing sub-discipline. We hope this article can be used to start the discussion about problem-based learning in the field of computing for diverse populations. Lastly, it would be great for the community to develop a repository of projects for computing courses that successfully engage diverse students.

Virtual College Fair

Exploring the Real World in Second Life

Kelly Czarnecki

Editor's Note: *Not only does the virtual world of Second Life offer unique computer science educational opportunities for learning programming concepts as described in* Virtual Programming *in this issue of the* Voice, *but also opportunities for students to explore real-world education and career exploration.*

In October 2008, college representatives, professors, librarians and, of course, high-school students, participated in the second college fair held in the virtual world Second Life (secondlife.com). The fair was organized as a first attempt to examine the feasibility of using this new technology to provide college information to prospective students, to give students more options in communicating with college representatives, and to connect colleges with tech-savvy teens.

Second Life represents the potential for a low-cost method of reaching students outside a school's usual geographic recruiting zone, and an opportunity for teens to talk with representatives from schools that may not otherwise come to their community.

Institutions and organizations from the U.S., the U.K., and New Zealand participated in the event. Over two days, unique teen avatars from several states and countries visited the fair, exploring objects on display tables that open Web browsers to a college admissions site, touching other objects to leave their contact information to get recruitment material by email, and talking to representatives and other teens attending the fair.

The students could also pick up material to take with them, including note cards of college information or virtual college t-shirts.

Speakers from NASA, Amazon, Linden Lab, several colleges and universities, as well as teen entrepreneurs presented on topics ranging from preparing for technology careers to succeeding in college. Informational displays that included Web

links, e-mail feedback objects, and video streaming allowed asynchronous communication of college information as well.

This year's event offered more opportunities for interaction between teens and college representatives through avatars from the participating institutions.

Participating in the fair is very inexpensive for most institutions. The only significant cost is a one-time background check for visiting adult avatars, which is a minimum of $40. The only other expense associated with participation is the cost of uploading content into Second Life, primarily for institutions that want to use images to customize their display objects with graphics. Each image costs a few cents to upload. Basic accounts in Second Life are free, as is the software download.

Teen Second Life is restricted to users aged 13-17 although adults can participate in educational projects by completing a background check to become an "approved adult" avatar in Teen Second Life. (There is a "main grid" within Second Life for adults, aged 18 and over.)

The college fair is organized by teen and adult volunteers with Eye4You Alliance, an educational island within Teen Second Life and sponsored by The Public Library of Charlotte & Mecklenburg County, NC. More information is available at *eye4youalliance.youthtech.info* and *teen.secondlife.com/*.

Virtual Programming

CS in Second Life

Jemsher Gill

A virtual world such as Second Life (SL) can be an exciting way to attract students to computer science (CS). Many students are already familiar with a virtual world environment through the games they play such as *World of WarCraft*. SL is a lot like *World of WarCraft*, but without the war.

I teach CS using SL because it offers numerous opportunities for me and my students. SL is free and therefore easy for students and teachers to access. The code that students create can be easily peer-reviewed and can also be sold in SL or freely shared. And most importantly, it's solid CS learning.

Programming in SL is done with Linden Scripting Language, which is based on the Java and C languages. Linden Scripting Language employs elements found in modern programming languages such as states and events. In SL, these events are often triggered by "touching an object." The classic CS "Hello World" example looks like this in Linden Scripting Language:

```
default
{
   touch_start(integer total_number)
      {
         llSay(0, "Hello World");
      }
}
```

SL provides a huge advantage over most programming environments in that it allows students to witness the result of coding in an immediate and exciting way. This is highly motivational. Often, beginning students produce code that is not very exciting. Beginning programmers in SL, however, make cool things happen quickly. Students can make their avatar jump from a tall building, open a parachute, and gently float to the ground. They can make their avatar drive a racecar or ride a horse. They might even cause the horse to change into a dragon! The virtual world is limited only by one's imagination.

All of this is made possible through solid CS knowledge. Students must learn the language and develop involved algorithms to solve classic CS problems. At the end of the process they have a virtual solution that they can witness immediately in SL.

The activities and events that occur in a virtual life can model real life, and therefore, many problems that students work to solve in one environment can be translated into the other. An example of this happened in *World of WarCraft* when an incident referred to as "corrupted blood" created a virtual plague in which avatars died very quickly. The avatars in that virtual world began avoiding large cities and gatherings of people. This virtual incident has led to the sug-

VIRTUAL PROGRAMMING
gestion of using SL to study the effect a viral outbreak would have on people in real life.

The virtual world of SL can be a valuable tool for motivating students and teaching CS. Many resources are available for learning about SL and the Linden Scripting Language at *secondlife.com/ whatis/scripting.php*.

Teaching Problem-solving Skills

Formalizing the Process

Evan J. Dembskey

Editor's note: *CSTA is proud to have a membership of teachers with diverse computer science (CS) teaching experiences and backgrounds. In an effort to meet the professional development needs of all of our members, we are pleased to publish a variety of articles on CS teaching methodology.*

Tshwane University of Technology in Johannesburg, South Africa, initiated a plan for teaching CS that would move beginning programming students from the basics toward becoming creative problem-solvers. The plan sequences four areas of knowledge and skill to be taught in the CS programming classroom.

The first skill set taught is the basic syntax of the language. This is a logical first step for students who are new to programming. Students must understand the basic building-blocks of the language before attempting to solve problems.

The second skill taught is a problem-solving process. Often students are taught some analysis and design skills, and a variety of programming methodologies, without clear guidance for when and why to use any particular method. A problem-solving process is important because it allows for control and understanding, and perhaps more importantly, it allows students to repeat the process to solve new problems.

The third set of knowledge critical for student success is programming standards. This is usually learned on the job but it may be beneficial to teach some basic programming standards at the university or even high school level, as the application of standards leads to repeatability, consistency, and a higher overall level of professional programming.

The fourth skill, the ability to solve problems with imagination, is difficult to teach. Experience shows that many students who are very capable in the first three areas cannot easily employ those skills to produce creative solutions to problems. Who has not heard the student lament: "It's easy to do now that you have shown me, but how do I know to do that?" The ability to creatively solve problems is partly dependent upon mastering the first three skill sets and partly upon thinking abstractly and creatively. There have been a number of attempts to formalize the creative process, including

> TRIZ is a methodology, tool set, knowledge base, and model-based technology for generating NEW ideas and solutions for problem solving.

de Bono's Lateral Thinking, TRIZ, and DEAM, as well as collaborative techniques such as brainstorming.

Lateral Thinking, a series of techniques developed by Edward de Bono, focuses on changing inaccurate concepts, perceptions, or reasoning that are not immediately obvious. It is essentially a set of idea-generation techniques. More information is available from *www.edwdebono.com/*.

TRIZ is a methodology, tool set, knowledge base, and model-based technology for generating new ideas and solutions for problem solving. It provides tools and methods for use in problem clarification, system analysis, failure analysis, and patterns of system evolution. TRIZ aims to create an algorithmic approach for inventing new systems and refining old systems. It was developed by Soviet engineer

Genrich Altshuller and his colleagues starting in 1946. More information is available at *www.triz-journal.com/*.

DEAM, or the Double-entry A-Ha! Method, is another problem solving technique. It is based on the novel (and difficult) idea of writing with both hands simultaneously. More information can be found at *www.winwenger.com/deam.htm*.

When a group of students all share effective problem solving skills, it is predictable that different programmers will produce similar solutions to identical problems. Thus, top-gun programmers are reigned in from producing brilliant, but difficult to understand or repeat solutions, and programmers of lesser ability produce satisfactory solutions to problems.

To formalize creative thinking while students are battling with syntax, strategies, and standards is perhaps ill-advised. What is advisable is to give students opportunities to think deeply about the problem and to understand the details of the problem, and the time to develop the insight needed to solve complex problems.

ACSL Provides Opportunities to Learn and Excel

Carlen Blackstone

The American Computer Science League (ACSL) competition extends the typical CS curriculum with interesting topics and an added objective for studying.

In order to provide schools with every opportunity to participate in ACSL, schools can register 3-person or 5-person teams, depending on the total number of participants and the size of the school. There are also four divisions, depending upon students' abilities.

- Senior Division—for AP-level, advanced programmers
- Intermediate Division—for students in grades 9-12 with some introductory programming experience
- Junior Division—for middle school students through 9th grade
- Classroom Division—for schools who don't wish to compete, but would like to use the content to enrich their CS courses
- Novice Division—for schools participating for the first time that would like to try out the League as an Intermediate 3-person team

There are four local contests throughout the year and an invitational All-Star contest that occurs on Memorial Day weekend.

Students explore topics including number systems, recursive functions, graph theory, digital electronics, prefix and postfix notation, Boolean algebra, data structures, finite state automata, and assembly language as they prepare to compete in the short problems component of the competition. In the programming challenge, students have 72 hours to solve a creative problem.

Over the years, students have reported how these contests have prepared them for college-level CS courses and motivated them to become better programmers and competent problem solvers. ACSL has sponsored competitions for 30 years in the U.S., Canada, and Europe. For more information visit: *www.acsl.org/*.

Congratulations, JEFF GRAY!

Dr. Jeff Gray, Associate Professor of Computer and Information Sciences at the University of Alabama at Birmingham (UAB), has been named the Alabama ***Professor of the Year*** by the Carnegie Foundation for the Advancement of Teaching. At UAB he co-directs the Software Composition and Modeling (SoftCom) laboratory.

Jeff was recognized for his leadership in K-12 outreach, developing tools that foster positive change, and enhancing learning. More information about the U.S. Professors of the Year Awards Program is available at *www.usprofessorsoftheyear.org/*

Meet the Authors

Carlen Blackstone
Emmaus HS, Emmaus, Pennsylvania
Carlen taught CS for 28 years and has participated in ACSL since 1983. She has led numerous CS and math teacher workshops.

Kelly Czarnecki
Public Library of Charlotte & Mecklenburg County, North Carolina
Kelly is the Technology Education Librarian at ImaginOn and works with teens and youth.

Ron Eglash
Science and Technology Studies, Rensselaer Polytechnic Institute
Ron is an Associate Professor and currently, chair-elect for the Coalition to Diversify Computing (CDC), a joint organization of the ACM, CRA, and IEEE-CS.

Juan Gilbert
CS and Software Engineering, Auburn University
Juan Gilbert is the TSYS Distinguished Associate Professor and a Center for Governmental Services Fellow. He has been active with CDC for over ten years.

Jemsher Gill
Garibaldi HS, Maple Ridge, B.C., Canada
Jemsher uses Slackware Linux Servers and provides students with access to Virtual Desktops, PHP/MySQL Databases, Command Line, Telenet, FTP, Web, Samba, C++ and Java.

Eileen Malick
Atlee HS, Mechanicsville, Virginia
Eileen has taught CS for 6 years and spent 5 years as a Department of Defense contractor. She is experienced with robotics teams, and engineering and video game programming camps.

Margot Phillipps
Educator, New Zealand
Margot teaches HS math and CS. Prior to that she taught CS in polytechnics and worked as an analyst programmer.

Valerie Taylor
Computer Science, Texas A&M University
Valerie is the Royce E. Wisenbaker Professor and Department Chair. She has been active in the CDC since its inception in 1996.

csta Voice

College Connection

Olivet Nazarene University

Editor's note: *This dialog with Larry D. Vail, Ph.D., Professor of Computer Science, Olivet Nazarene 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 Olivet Nazarene University.*

Founded in 1907, Olivet Nazarene University is one of the nation's premier Christian colleges where faith is at the heart of outstanding academics, athletics, and social gatherings. Located in Bourbonnais, Illinois, the beautiful park-like campus includes 30 major buildings on 250 acres just 50 minutes south of Chicago's Loop. The CS Department offers bachelor's degrees in Computer Science and Computer Information Systems. The department also offers interdepartmental bachelor's degrees in Computer Engineering and Management Information Systems.
CSTA: What draws students to your program and what keeps them there?
Vail: Come for a campus visit and you'll feel the friendliness of faculty, staff, and students. There is a wide variety of opportunities in athletics, intramurals, student leadership, music ensembles, and service and ministry organizations. Computing sciences are academically challenging but the department promotes a supportive and collaborative environment among professors, teaching assistants, and students where everyone participates in and contributes to the learning process. Personal attention in class work and in career advising helps students to begin careers with confidence, built on a solid foundation.
CSTA: What skills can students acquire before college that will help them succeed in your program?
Vail: Students need good study habits and self-discipline to successfully handle the fast pace of college classes. Reading comprehension and technical writing skills are important for problem analysis and documentation. Quantitative skills and problem solving ability are often developed in mathematics courses and physical sciences like chemistry and physics because computing requires problem solving. General computing skills such as word processing and spreadsheets are very helpful in computing classes as well as general education classes. Some students have opportunities to take programming classes or help with computer system and network administration. Any and all of these skills will contribute to your success in college and specifically in a computing degree.
CSTA: What cool careers are your graduates prepared for?
Vail: Our graduates have been successful in a wide variety of challenging and exciting careers. These include Web developers and architects, software engineers, systems analysts, system and network administrators, database administrators, software and system sales, computer trainers, business analysts, and information technology managers. They work for aerospace, government, Christian missions, education, telecommunications, healthcare, banking, insurance, manufacturing, pharmaceuticals, as well as software and hardware companies.
CSTA: What topics will students study?
Vail: Computing students study a core that includes programming, databases and information systems, systems and networks, and professionalism and ethics. Depending on their particular degree, students may take additional courses such as systems programming, systems analysis and design, digital circuit design, or business management. More information is available at *www.olivet.edu* under academics.
CSTA: Tell us a bit about the social environment of the CS program.
Vail: Students hang out in the lab with friends to work on assign-

ments and personal projects. The CS club is active in presenting student-led technical seminars and service projects. They also sponsor parties, for students and professors, with lots of food, console and PC based games, board games, and crazy competitions like frisbee golf and tennis golf. Programming competitions have been an important part of the department for over 20 years. Participation in competitions has been at several levels, including intra-mural, Chicago area, Midwest regional, and national.

Spotlight

A Video Game Programming Camp
Eileen D. Malick

A video game programming camp is an effective way to build excitement for computer science (CS) and recruit students to your courses, and middle school students are the perfect audience. Conducting a video game programming camp involves a great deal of preparation. Here are a few pointers based upon my experiences to help you plan for a successful camp.
1. Find Funding Support A variety of grants are available for projects that enhance math and science education. However, use grants as backup funding and pursue a partnership with a not-for-profit foundation, or a partnership with the CS branch of a local university for a more lasting program. In most partnership arrangements, you will be the expert for adapting the subject matter to this age group, so you will have the majority of responsibility.
2. Find Support Staff Teachers or administrators from the high school as well as middle school level can provide ideal support staff. Recruit early. Get administrative approval from all of the associated schools. High school and middle school administrators appreciate opportunities to collaborate.
3. Recommended Camp Structure: The Triad Plan camp activities for three separate rooms. Rotate the group through three rooms where each room serves a different function. If you have a large group of students, randomly assign the attendees into manageable groups.
Blue Room: Game programming concepts
What makes a good video game? Why are some video games not entertaining? What distinguishes an RPG, shooter, or platform game? What is the history of games? Predict how games will be different in the future. What affect do certain control statements, loops, and other CS concepts have in games? These concepts are best illustrated and discussed with multimedia support.
Green Room: Meeting with adult role models
Invite adults with CS careers to visit your camp. Students can become comfortable by first asking questions from a prepared list. These prepared questions often spark additional discussion on topics of special interest to the campers. Encourage role models to create a short presentation about their company ahead of time. Some will bring items to share from their company such pens, shirts, or backpacks.
Yellow Room: The programming lab
Ensure that a comfortable computer lab with appropriate software is available. Free sprites and audio files should also be provided.
4. Pick a Software Platform Since the majority of campers have experience playing video games, but very little background in programming, a GUI software package with templates and tutorials is desirable.
5. Daily rewards Parents are your greatest support system so be sure their children have something to share with them at the end of each day. T-shirts, lanyards, CDs, brochures, printed screen shots of their project, or prizes from competitions generate discussions of camp activities long after the completion of your camp.

Promoting CS Education

Student Ambassadors to the Rescue
Pat Phillips

Now in its second year, the University of California Irvine's (UCI) Bren School of Information and Computer Sciences (ICS) Student Ambassadors program is on a mission to reach out to potential students through a near-peer outreach program that profiles student successes.

The program features seven student ambassadors who are the face and voice of Bren ICS to prospective students. They are responsible for outreach and promotion to area high schools and community colleges and also participate in on-campus functions.

The ambassadors use a wide variety of outreach strategies including planning, scheduling, and hosting outreach events; developing and reviewing marketing materials; and providing school and campus information to prospective students and their families by phone, electronic, and in-person meetings. For example, they recently produced a flyer for distribution to schools that invites students to e-mail their questions to them.

Each ambassador also develops working relationships with 3 to 4 feeder high schools or community colleges and contributes to the team blog at *ambassadors.ics.uci.edu/*.

The ambassadors provide valuable information about CS and Bren ICS from a student's point of view. Prospective students from high schools and community colleges learn first-hand about college life and student concerns, as well as about courses and class projects. They believe their success in connecting to students is due to their ability to identify with younger students and to provide a peer-level view of college and CS today.

The ambassadors' outreach is not limited to potential Irvine students. They also work to build an identity and community among students in the Bren ICS program through such events as ice cream socials.

The ambassadors also gain important skills and experience from their involvement in this program. They have the opportunity to strengthen their public speaking, program and event planning, teamwork, and leadership skills.

The 2008 Ambassador team includes: Krishna Patel and Christophere Riviere-Escobedo, Computer Science and Engineering majors; Tae Kim, Sam Kaufman, Gabriela Marcu, and Alex Bretana, Informatics majors; and Martin Pieters, Information and Computer Science major.

The team is directed by Christine M. Leon, Director, Student Affairs for Bren ICS. Teachers can request information by e-mailing *ics.ambassador@gmail.com*.

Listen and Share

Hear first-hand the experiences of the Bren Student Ambassadors. Learn about their motivation to become ambassadors, how the ambassadors can help classroom teachers, their plans for the future of the ambassador program, how they measure their success, and what they love about UC-Irvine and Bren ICS.

Visit the CSTA Snipits podcast at *csta.acm.org/Resources/sub/Podcasts.html*.

Out and About the Community

A Journey to the Olympiad
Margot Phillipps

The New Zealand International Olympiad in Informatics (IOI) team has become a contender in just three short years. A great deal of work and planning made this year's entry a success. Ronald Chan, a first-year university student at Auckland University, earned a spot in the silver medal bracket in this year's competition in Cairo, Egypt, during the August competition.

Our journey to the International Olympiad in Informatics (IOI) began in 2006 when we formed an organization to train and select a team. The first visit to the IOI is typically just that—an observational visit. In August of that year, I took the risk and traveled with the yet-unofficial New Zealand team to Merida, Mexico. Our top student scored only four competitors below the bronze medal ranking.

With the help of the New Zealand Mathematical Olympiad organization, I set up an incorporated society, found staff for a summer training camp, enlisted help from the Australian team, and spent many hours raising the necessary funds. In August 2007, a team of 4 students participated in the competition in Croatia. Again, our top student fell just below the bronze medal reward.

During this last year, Tobias Thierer (a former medalist from Germany) served as the deputy leader. That summer the camp grew from 4 to 14 students and the competition to make the team intensified. Students prepared by practicing online contests and studying with university lecturers. The USACO Training Program Gateway, operated by Rob Kolstad, also offered competitors excellent preparation.

The IOI competition occurs over two intense five-hour days. Three problems, each worth 100 points, are tackled each day. The problems are highly algorithmic in nature and there are often at least two solutions; only the most efficient solution will pass the time and memory constraints. A score of less than 100 is possible for a slow solution, with perfect scores reserved for a very "smart" solution.

The tests are scored by running the contestants' solutions and comparing the results with the judges' output. No feedback is offered during the contest, although at the end of each day contestants learn if they have passed or failed the problems.

More information about the IOI is available at *ioinformatics.org/index.shtml*.

The U.S. Bureau of Labor Statistics released its latest projections of jobs that will experience strong growth.

THESE ARE THE TOP FIVE JOBS IN THE IT FIELD.

	Growth by 2016	Median annual salary
Network/communications analysts	53%	$64,600
Applications software engineers	45%	$79,780
Database administrators	29%	$64,670
Systems software engineers	28%	$85,370
Network/system administrators	27%	$62,130

SOURCE: U.S. Bureau of Labor Statistics *www.bls.gov/*

Voice

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

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

MARK YOUR CALENDAR

Florida Educational Technology Conference (FETC)
January 21-24, 2009 in Orlando, Florida
www.fetc.org

Texas Computer Education Association (TCEA) Convention
February 2-6, 2009 in Austin, Texas
www.tcea.org/convention/2009

SIGCSE 2009
March 4-7, 2009 in Chattanooga, Tennessee
www.cs.arizona.edu/groups/sigcse09

Computer Using Educators (CUE)
March 5-7, 2009 in Palm Springs, California
http://www.cue.org/

Richard Tapia Celebration of Diversity in Computing
April 1-4, 2009 in Portland, Oregon
tapiaconference.org/2009

Consortium for Computing Sciences in Colleges
(CCSC: Mid-South)
April 3-4, 2009 in Martin, Tennessee
www.ccsc-ms.org

Consortium for Computing Sciences in Colleges
(CCSC: Central Plains)
April 3-4, 2009 in Bolivar, Missouri
www.ccsc.org/centralplains

Consortium for Computing Sciences in Colleges
(CCSC: Southwestern)
April 3-4, 2009 in San Diego, California
www.ccsc.org/southwestern/2009

Consortium for Computing Sciences in Colleges
(CCSC: South Central)
April 24-25, 2009 in Hammond, Louisiana
www.sci.tamucc.edu/ccsc

Consortium for Computing Sciences in Colleges
(CCSC: Northeastern)
April 24-25, 2009 in Plattsburg, New York
http://www.ccscne.org/2009

International Conference on the Foundations of Digital Games
April 26-30, 2009 leaving from Port Canaveral, Florida
foundationsofdigitalgames.org

Oregon Game Programming Challenge (OGPC)
May 16, 2009

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

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

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

**CSTA INSTITUTIONAL MEMBER
K-12 OUTREACH PROGRAMS**

TechTopia Challenge
February 6, 2009 and July 31, 2009
Neumont University in Jordan, Utah
www.tech-topia.com

CSTA Nominations Due
February 6, 2009

RESOURCES

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

Page 1: *Culture, Literacy, and Learning: Taking Bloom in the Midst of the Whirlwind* (Teachers College Press 2007)
Page 2: Eye4You Alliance *eye4youalliance.youthtech.info*
Page 2: Second Life *www.secondlife.com*
Page 2: Teen Second Life *teen.secondlife.com*
Page 3: *Second Life: The Official Guide* (Sybex 2008)
Page 4: Lateral Thinking *www.edwdebono.com*
Page 4: The TRIZ Journal *www.triz-journal.com*
Page 4: DEAM Problem Solving *winwenger.com/deam.htm*
Page 5: American Computer Science League *www.acsl.org*
Page 5: U.S. Professors of the Year *www.usprofessorsoftheyear.org*
Page 5: Carnegie Foundation *www.carnegiefoundation.org*
Page 6: Olivet Nazarene University *www.olivet.edu*
Page 6: International Olympiad in Informatics *ioinformatics.org*
Page 6: New Zealand Olympiad in Informatics *www.nzoi.org.nz*
Page 7: Bren School ICS at UC-Irvine *www.ics.uci.edu*
Page 7: Bren School Student Ambassadors *ambassadors.ics.uci.edu*

csta
Computer Science
Teachers Association

Share your Ideas!

Visit the Advocate Blog today!

blog.acm.org/csta/