Gearing Up for Fall

csta Voice

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

Volume 9, Issue 3 July 2013

## Inside This Issue



IN THE NEXT ISSUE OF THE *VOICE*

Advocacy

# CSTA Conference: A Summer Highlight

**THE 2013 CSTA ANNUAL CONFERENCE**, July 15–16 in Massachusetts, promises to be the highlight of the summer for computer science (CS) educators from across the country and around the world. For those who are unable to join us this year (we hope it's in your plans for next year, July 14–15), this issue of the Voice features previews and content from many of the presentations and workshops. The keynote speakers, ten workshops, 20 sessions, opportunities to network with peers, and even a trip to Microsoft's New England Research Development Center & the Microsoft Technology Center, promise to provide abundant, high-quality resources and energy as you gear up for another exciting year of teaching CS. Come and join us for the biggest K–12 event of the year.

CSTA is grateful for the generous support of the 2013 CSTA Conference sponsors and for their ongoing support of and contributions to CS education.

**Anita Borg Institute for Women and Technology**
**Microsoft**
**Microsoft Research**
**Oracle Academy**

And a special thank you to the 2013 CSTA Conference Planning Committee for their creativity and dedication: Dave Reed, Doug Peterson, Padmaja Bandaru, Duncan Buell, J. Philip East, Patrice Gans, Tammy Pirmann, and Kelly Powers.

Read more about the sessions and presenters:
*csta.acm.org/ProfessionalDevelopment/sub/CSTAConference.html*.
Follow us on Twitter: hashtag #CSTA13.

# Teaching for Diversity through Web Design

**Terrill Thompson**

**WE TYPICALLY USE THE WORD "DIVERSITY"** in reference to cultural, racial, and/or socioeconomic differences, but humans are also increasingly diverse when it comes to the technologies we use to access electronic information, communicate with one another, and perform various functions in our work and daily lives. Gone are the days when computing occurred exclusively at desktop workstations—now it also occurs on phones, tablets, in smart cars, and on an ever-widening spectrum of assistive technologies used by people with disabilities.

*continued on page 2*

## TEACHING FOR DIVERSITY THROUGH ACCESSIBLE WEB DESIGN

*continued from page 1*

According to the World Health Organization, more than one billion people with disabilities worldwide face substantial barriers in their daily lives. Many of these individuals use screen readers, Braille devices, screen magnification programs, speech recognition applications, or any of hundreds of other alternative input or output technologies. This rich technological diversity presents challenges for software application developers, web designers, computer scientists, and engineers. They need to develop products that work across an increasingly broad spectrum of users with highly diverse characteristics and needs.

For computer science (CS) educators, technological diversity presents an opportunity. It can provide context for us to incorporate an appreciation for human diversity into our courses. We can encourage our students not only to learn code, but also to appreciate why they are coding, and for whom, and to recognize that their end users are individuals with widely varying needs. How can this software application be created in a way that will work for people who will access it visually, as well as for people who access it audibly? How can this tool be created so that it can be operated with or without a pointing device? These are interesting challenges, but important ones to address. Students have an opportunity to appreciate human differences, and to understand that few users of technology interact with it in exactly the same way that they do.

AccessComputing is a project at the University of Washington (UW) that was funded by the National Science Foundation to broaden participation of individuals with disabilities in computing careers. It is a partnership between the Department of CS and Engineering and The DO-IT Center (Disabilities, Opportunities, Internetworking, and Technology), both at the UW. One of the outcomes of this project is a curriculum for an Introduction to Web Design & Development course designed especially for high schools. The curriculum encourages students to consider human diversity, and teaches the importance of standards for ensuring accurate communication across devices and platforms.

In the first unit, students learn several core design principles, one of which is the principle that technology should comply with standards and should be usable by everyone. This unit includes a module that focuses specifically on the technologies used by individuals with disabilities and the need for accessible websites and applications for these users. With this foundation in place, students revisit those core design principles throughout the course, and are encouraged to constantly ask whether a new method or technique they are learning might result in barriers

> One of the outcomes of this project is a curriculum for an Introduction to Web Design & Development course designed especially for high schools.

# CSTA THANKS

Outgoing CSTA Board Members

Ria Galanos, 9–12 Representative, and Mindy Hart, At-Large Representative, for their leadership and dedication to CSTA and CS education.


## Executive Officers

**Deborah Seehorn**
Chair
*deborah.seehorn@dpi.nc.gov*

**Stephen Cooper**
Past Chair
*cooper@cs.stanford.edu*

## Staff

**Dr. Chris Stephenson**
CSTA Executive Director
Phone: 1-800-401-1799
*cstephenson@csta.acm.org*

**Lissa Clayborn**
CSTA Director of Development
*l.clayborn@hq.acm.org*

**Pat Phillips**
Editor
Phone: 1-608-436-3050
*cstapubs@csta.acm.org*


## Visit CSTA Online

**Membership Information**
**Curriculum**
**Teaching Resources**
**Professional Development**
**Advocacy and Outreach**
**Research**

*csta.acm.org*

**CSTA Voice ISSN: 1555-2128**

**CSTA Voice** is a publication of the Computer Science Teachers Association.

**CSTA Voice** is a quarterly publication for members of the Computer Science Teachers Association. It provides analysis and commentary on issues relating to K–12 computer science education, resources for educators, and information for members. The publication supports CSTA's mission to promote the teaching of computer science and other computing disciplines.

**Change of Address and Membership Questions:** Contact Member Services via e-mail at *cstahelp@csta.acm.org*, or call 1-800-342-6626 (U.S. & Canada) or +1-212-626-0500 (Global).



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

to accessibility. If it will, they are challenged to think of solutions.

For example, as students learn about the role of graphics on the web, they also learn about how to ensure that the content or function of their graphical content is accessible to users who can't see the images. Similarly, as they learn to add interactive functionality to web pages using JavaScript, they learn how to ensure this functionality is not dependent on users being able to operate a mouse. If a mouse user can trigger a change on the web page, a keyboard user should be able to trigger that change as well.

The curriculum is cross-platform and vendor-neutral, and is available for free at *uw.edu/accesscomputing/webd2*. Nearly 4000 teachers have registered to access the curriculum, representing 108 countries. Countries with ten or more registered teachers include (in descending order) United States, India, Canada, Australia, Philippines, United Kingdom, Nigeria, Ireland, Sri Lanka, Pakistan, Malaysia, Thailand, South Africa, Uganda, Indonesia, and China. If students all over the world acquire web design and development skills while simultaneously learning to appreciate and value human diversity, we like to think something very positive can come of that.

Terrill Thompson will discuss these concepts and share lesson ideas during a session, *Teaching Standards–based, Accessible Web Design*, at the CSTA Conference on July 16, 2013.

# Creative Engagement with Technology

*There's an App for That*

**Kristin Violette and Shaileen Pokress**

**LIVING IN A WORLD** with abundant personal computing devices, such as smart phones and tablets, today's students have become more technology consumers than technology creators. This no longer needs to be the case; MIT's App Inventor can be a game changer for your work in making computer science (CS) relevant to students' lives.

MIT App Inventor (*appinventor.mit.edu*) is a tool for building mobile applications that is free of cost from MIT's Center for Mobile Learning. App Inventor enables students with no previous programming background to learn advanced CS concepts by building apps for Android mobile devices in a user-friendly environment. Empowering students to create their own mobile apps can infuse much-needed energy and excitement into CS education.

Students typically experience computing in very tactile and personal ways. App Inventor taps into these experiences to provide a computing environment that is not only fun and exciting, but also real-world relevant. Educators report that students are drawn by the "wow factor" of being able to create their own apps with meaningful utility that can be shown off to friends and family and even offered on the app market. Students who have designed and developed apps with App Inventor often demonstrate a passion for software development.

This year, Connecticut's Newtown High School launched a year-long capstone course featuring App Inventor. Students are developing a quiz app to be deployed in a physics class as well as a teacher evaluation app to be implemented by a Connecticut regional education service center. A group of Newtown students recently took first place in the Connecticut Student Innovation Expo with a mobile app designed using App Inventor. These projects have forged a personalized connection for students between CS and the community.

There are numerous, free resources available on the App Inventor website. The forums, tutorials, and educational materials offer technical and curricular support. New curricula and online support are under development. Check the "teach" section of the website for information on professional development opportunities.

Kristin Violette and Shaileen Pokress will demonstrate App Inventor, and share successful strategies during a session, *Inspiring High School Students in STEM Using App Inventor* at the CSTA Conference on July 16, 2013.

# NEWLY ELECTED CSTA BOARD MEMBERS

**Patrice Gans**
*Fraser-Woods School*
*Newtown, CT*
K–8 Representative

**Stephanie Hoeppner**
*Clermont Northeastern High*
*Batavia, OH*
9–12 Representative

**Dave Reed**
*Creighton University*
*Omaha, NE*
College Faculty Representative

**Alfred Thompson**
*Bishop Guertin High School*
*Danville, NH*
At-Large Representative

**Meet the New Board Members**
*csta.acm.org/About/sub/AboutFiles/2013Candidates.html*

# A Spark and a Plug

*Computer Science in the Transportation Industry*

**Janice M. Tkaczyk**

*"Computer science (CS) deals with the theoretical foundations of information and computation, together with practical techniques for the implementation and application of these foundations."*

When most students hear this quote, I suspect they think of the technologies with which they are most familiar—smart phones and video games. Students are unlikely to think about the transportation industry and yet, the applications and career-related pathways in transportation are numerous and students should know about the ways in which changes in industry are creating new career opportunities.

**The Spark:** I am a retired high school counselor and I don't know much about CS applications in today's automobiles—but I do know who to ask. The first thing I did in preparation for this article was to consult with Neal Steinkrauss, an instructor at our Universal Technical Institute (UTI) on the Norwood, Massachusetts campus.

"As computers became faster and less expensive they found their way into more automotive systems," Steinkrauss told me. "Antilock brakes, air conditioning, suspension, transmissions and supplemental restraint systems are just a few. These systems operated independently of each other at first but now the challenge is to have all the systems working together. We have a low speed LIN (Local Interconnect Network) for sensors and switches, and a high speed CAN (Controller Area Network) for our modules and computers. With these systems automotive advances have skyrocketed. The high-end cars of today have over 40 sensors and 50 computers interacting as we drive."

There are numerous systems in today's automobile, including systems to:

- Alert us about a car in our blind spot, or pull us back into a lane so we do not hit a car we cannot see.
- Warn us if we are about to have a crash and then fully apply the brakes, roll up the windows, tighten the seatbelts, close the sunroof, and adjust the seats to a safer position. And then after the crash, systems turn on the interior and hazard lights, unlock the car doors, and call for help.
- Use Doppler radar to track whatever is around us as we drive.
- Stabilize and compensate for over-steering and under-steering.
- Park our cars without our hands on the wheel.

Many cars today are also equipped with cloud computing systems. These systems provide added features such as geo-fencing (a dynamically generated radius around a location) and speed alert (a system that lets drivers know when they have reached the maximum speed limit).

We also have new technology to help clean car emissions. With fuel injection, a closed loop feedback system, and the newest catalytic converters, today's cars turn carbon monoxide (CO), hydrogen chloride (HCl), and nitrogen oxide (NOx) into carbon dioxide (CO2) and water (H2O). This gives us better performance, lower emissions, and better fuel economy.

All of these advancements have resulted in an ever-increasing need for advanced training for service personnel. So how do we effectively get this information to our K–12 students?

**The Plug:** What I learned during my thirty-five years in education is that it is all about partnerships; professional school counselors can be wonderful partners in helping students understand career opportunities and requirements.

Invite members of the guidance office to work with you to make the connection between what is being taught in class and post-secondary training and career pathways. Counselors can help add the career relevance piece that builds connections from classrooms to the world of work. They can also help establish business and industry partnerships. For, example, civic clubs such as Rotary and Kiwanis are eager to offer speaker panels on careers. They may be able to provide job shadowing; participate in career and college fairs; or help with expenses for field trips, equipment, and scholarships.

Business, industry, and post-secondary schools around the country are building partnerships to add relevance to classroom activities. For example, UTI has developed both STEM tours and workshops to help students see the relationships between the subjects they are studying and the transportation industry. These tours and workshops highlight the range in STEM-related careers and help students see the connections between classroom learning and real jobs, applied academics and hands-on education.

I invite you to take the "spark" of Mr. Steinkrauss' update on the transportation industry and the "plug" for in-school and community partnerships and use them to ignite your students' interest in new career pathways.

## Planning for Chapter Success

**WITH SUMMER HERE,** it's a great time to get ideas from other CSTA chapters and make plans for the upcoming year. The Puget Sound Computer Science Teachers Association (PSCSTA *www.pscsta.org/*) and the Central New Jersey chapter (CSTACNJ) are active growing chapters—the result of creative ideas, collaborative efforts, and thoughtful planning. Here are some suggestions from Hélène Martin (PSCSTA) and Daryl Detrick (CSTACNJ), for growing a successful, impactful chapter.

### Planning Chapter Activities

- Set goals for the year. Goals from CSTACNJ include:
  - Promote CS education.
  - Provide professional development opportunities.
  - Create relationships from kindergarten to higher education.
- Involve your entire membership in planning.
- Schedule major events such as competitions and celebrations first.
- Brainstorm topics for other meetings.
- Plan meetings for a set date each month, such as the second Saturday.
- Assign a member or two to facilitate each session.
- Plan for variety, including workshops, speakers, visits to schools or tech businesses, and demonstrations.
- Post the schedule and meeting descriptions on the website.
- Add to Google Calendar so members can import into their own calendaring systems.
- Use Google Groups, Facebook, and other social media to involve all members.
- In August, send the meeting schedule and details to all CSTA members in the state.

### Successful Meetings Strategies

- Send a reminder and agenda before the meeting. Request an RSVP.
- Stick to the time schedule and agenda.
- Start with attendee introductions.
- Provide refreshments.
- Share resources, including items from the CSTA website.
- Provide time for attendees to socialize and share ideas.

For information about all 45 CSTA chapters visit: *csta.acm.org/About/sub/CSTAChapters.html*

## Congratulations CSTA Members

The Connecticut CSTA Chapter participated in the successful NSF grant proposal that awarded Trinity College $902,000 to teach high school CS teachers in Hartford and other Connecticut cities. This is a wonderful example of how CSTA chapters are working with local faculty to build capacity and conduct research.

***www.hartfordbusiness.com/apps/pbcs.dll/article?=/20121210/NEWS01/121219981***

## Meet the Authors

**Mark Dorling**
*Digital Schoolhouse, UK*
Mark is the Digital Schoolhouse project coordinator, Board Member of Computing At School, and Lecturer at Brunel University.

**Stacey Jenkins**
*Langley Grammar School, UK*
Stacey is a Department for Education (DfE) funded Computing At Schools Master Teacher and serves as the head of the ICT department.

**Shaileen Pokress**
*MIT Center for Mobile Learning*
Shaileen directs outreach and curriculum development for the App Inventor project to engage students with computing in meaningful and personal ways.

**Alexander Repenning**
*University of Colorado*
Alexander is a CS professor and the founder of AgentSheets Inc. He is directing the Scalable Game Design Initiative.

**Terrill Thompson**
*University of Washington*
Terrill is a technology accessibility specialist and co-author of the *Web Design & Development I* curriculum.

**Janice M. Tkaczyk**
*Universal Technical Institute, MA*
Janice is the National Director for Counselor and Academic Relations for UTI and an adjunct professor at UMass Boston for Professional School Counselors.

**Sheena Vaidyanathan**
*Los Altos School District, CA*
Sheena teaches CS and enjoys combining CS and art to engage and motivate her 6th grade students.

**Kristin Violette**
*Newtown High School, CT*
Kristin has been teaching technology for 17 years. Funded by an Innovation Grant by the State of Connecticut, she is currently teaching a capstone course, "An App for That."

**Pat Yongpradit**
*Springbrook High School, MD*
Pat writes curriculum and advocates for CS education. He is a 2010 Microsoft Worldwide Innovative Educator.

## Research Review

*From Games to STEM*
*A Sustainable Path to Broaden Participation*

**Alexander Repenning**

The Scalable Game Design project (SGD) has developed and tested a strategy to broaden participation in computer science (CS). The strategy is based on a path that introduces students to computational thinking (CT) through game design and then advances to the creation of STEM simulations. As part of ITEST and CE21 NSF research, we have tracked motivation and skill acquisition across gender and ethnicity. Over 9000 students from technology hub, inner city, rural, and Native American communities participated. Over 10,000 student projects were analyzed. We have found extremely high levels of motivation and CT skills acquired.

SGD uses AgentSheets and AgentCubes, computational thinking tools that support game design and STEM simulation creation, provide rich media authoring, and facilitate debugging. The SGD strategy is based on four principles:

**Exposure:** Starting in elementary school with game design modules integrated into existing computing courses, SGD reaches a very large and highly diverse group of students. In large schools hundreds of students per year participate. Girls make up 45% of all students and 48% of the students are from underrepresented groups. This is significantly higher than typical outreach approaches such as after-school programs.

**Motivation:** Motivation and scalability are achieved by balancing skills and challenges. A series of increasingly sophisticated game design and STEM simulation projects, starting at a low threshold, engage students. Projects range from simple 2D arcade-style games that students can build in a couple of hours to complex 3D games based on artificial intelligence. High levels of motivation are expressed by students' interest in taking additional classes: 74% of boys, 64% of girls (100% for some schools); 71% of white, 69% of minority students.

**Education:** To evaluate CT skills, we created the Computational Thinking Pattern Analysis (CTPA) assessment instrument that analyzes student creations for CT patterns. Game design and simulation building share some fundamental CT concepts.

**Pedagogy:** Tracking motivation levels and skills across gender, we found that the most significant indicator predicting CT success with girls is pedagogy. Employing a combination of a "projects first, just-in-time principles" approach with inquiry-based scaffolding that answers "WHAT we do," "WHAT ELSE we could do," and "WHY we do" can broaden participation.

What is a teacher's experience? Mark Shouldice is a middle school teacher at Mrachek Middle School in the Aurora Public School District in Colorado. The school population is 79% minority and 21% white. He has been participating in the Scalable Game Design project since 2008 and reaches 350 students each year. His experiences are "the most stirring I have seen in 22 years of teaching. It is exciting to talk about the high level of student engagement using SGD curriculum. Students are choosing to take additional classes, wanting to learn more about game design."

Alexander Repenning will conduct a workshop, *Computational Thinking, from Game Design to STEM in One Week*, at the CSTA Conference on Monday, July 15, 2013.

**LEARN MORE:**
SGD *scalablegamedesign.cs.colorado.edu*

## Spotlight

*Get a Head Start with a CS Principles Course*

**Pat Yongpradit**

Are you planning to teach the *Advanced Placement Computer Science (AP CS) Principles* course and looking for resources to align to the CS Principles' Big Ideas and Learning Objectives? The *CS Principles: Computation in Action* curriculum engages students in socially-relevant, project-based learning designed to teach the concepts of *AP CS Principles*.

The *AP CS Principles* course was designed around the Big Ideas in Computer Science: Creativity, Abstraction, Data, Algorithms, Programming, Internet, and Impact. The *CS Principles: Computation in Action* curriculum helps students put these concepts into action by creating interdisciplinary computational artifacts that combine CS with music, art, literature, math, and science. The curriculum supports the *CSTA K–12 Computer Science Standards Level 3* by promoting the use of computational strategies to explore and develop solutions to real-world problems. Projects and activities emphasize algorithmic problem solving, ethical computing, and modern collaborative tools and technologies.

The *AP CS Principles* learning objectives offer multiple opportunities for interdisciplinary connections to the *Math Common Core* and *Next Generation Science Standards (NGSS)*. The *CS Principles: Computation in Action* curriculum includes projects to address a wide range of cross-curricular standards such as writing equations with two or more variables to express a relationship (math) and creating computer simulations to model interactions between different systems (science).

*AP CS Principles* promises exciting opportunities for CS educators and students for broadening participation with up-to-date technologies and relevant content. The *CS Principles: Computation in Action* curriculum is similarly designed to be current and broadly appealing. The curriculum package includes lesson plans, activities, assessments, and video tutorials. The first units are now available on the CSTA website at *csta.acm.org/Curriculum/sub/CurrResources.html*. The complete curriculum will be ready in October. Students can look forward to:

**Unit 1: Algorithms**
Algorithms are everywhere! They are responsible for music and movie recommendations on media websites, product suggestions on commerce sites, and even recommendations for friends in social media. In Unit 1, students will examine algorithms in everyday life, from maneuvers involved in dancing and sports to programs that control actions in video games. They will learn about algorithmic instructions, and how to express algorithms to solve problems. The learning culminates in a game created with natural language, pseudo-code, and the Kodu Game Lab.

**Unit 2: Programming as an Expression of Creativity**
In Unit 2, students will move from examining and creating stand-alone algorithms to incorporating multiple algorithms into programs complete with user input and graphical output. Small Basic is used as the programming tool because the language and editing environment are simple and beginner-friendly. Students will work collaboratively employing pair- and agile-programming methodologies. At the end of Unit 2, students will work as designers, programmers, artists, and managers to collaboratively design, develop, test, and refine their programs.

**Unit 3: Data and Abstraction**
In Unit 3, students will explore computational systems, a hierarchy of abstractions, and the nature of digital information. They will use their algorithmic and programming abilities to analyze data sets and abstract data using Microsoft Excel, and then investigate problems and create simulations to answer questions and propose additional investigations. The unit culminates with a collaborative project in which students select a large data set, investigate a problem, select computational tools, and create digital artifacts.

**Unit 4: The Internet and Impact**
In Unit 4, students will learn about how the Internet works and how computing enables innovation. Students will explore the numerous technological products that the Internet has enabled and the significant societal changes that have resulted. To conclude the unit, students will work collaboratively to connect their understanding of the Internet and the global impact of computing to a specific contemporary problem that is associated with the United Nations Millennium Development Goals, propose a computational solution, and create a digital prototype of their solution.

## CS Around the World

*Computing through Dance*

**Stacey Jenkins and Mark Dorling**

Kinesthetic learning is giving a boost to computer science (CS) education with a project developed by the Digital Schoolhouse and Langley Grammar School's ICT Department in the United Kingdom (UK). "Computing through Dance" begins with enjoyable physical activity and ends with mastery of computing concepts.

The dance and music can be selected based upon curriculum needs or by the students; options might include the Hokey Pokey, the New Zealand rugby team's Haka (see the video here: *www.digitalschoolhouse.org.uk/algrithms/294*), Michael Jackson's Moon Walk, a Tudor dance studied by many UK children, or an original choreograph. After learning the dance moves, teams create flow charts of dance-step instructions that will be followed by the other class members as they attempt to perform the dance.

The initial objectives are to develop an understanding of sequencing and the importance of accurate instructions. The concept of looping is introduced to implement repetitive segments of the dance. The concept of selection is presented with the introduction of a branch into the dance sequence: if the whistle blows then freeze in a pose, else perform the dance.

After the concepts are understood, the teams create dance instructions that include four distinct dance moves, at least one repeat segment, and a pose to hold when the whistle blows. The sequence of dance steps is recorded in a flow chart. And then the action begins as the class dances through each instructional sequence. If classroom space is limited, the dance moves can be restricted to upper-body actions only.

For the peer feedback phase of the project, the dances should be recorded using video cameras. Students compare the flowchart to the performance and award points on the clarity of instruction, sequencing, use of repeats and branching, and quality of dance moves.

The Computing through Dance project then moves into the Scratch programming phase. Students select a dance character and program a dance performance to illustrate their understanding of sequence, repetition, and selection branching.

This project can be extended by adding additional computing concepts such using a variable to determine the number of times the dance sequences are repeated and the use of procedures for the more complex dance moves.

**LEARN MORE:**

*www.digitalschoolhouse.org.uk/algrithms/294*
*www.resources.digitalschoolhouse.org.uk/key-stage-2-ages-7-10/218-scratch-teaching-dance*

## Curriculum in Action

*Programming through Art and Math*

**Sheena Vaidyanathan**

Teaching computer science (CS) through art engages students and showcases the creativity behind programming. Connecting programming to math makes CS relevant to the school curriculum and ensures support and funding.

I combined CS with math and art to create a CS course called CSTEM for the Los Altos School District in California. Because CSTEM is a required class, every one of the over 500 sixth graders learns to code as a medium for creating art.

This year, I used the JavaScript implementation of Processing with the Khan Academy IDE. We began with a 'human computer' exercise in which students acted out three lines of code to draw a house and then coded the process on the computer.

Students were surprised to see how the code on one side of the IDE translated into a design on the other. Students experimented with Processing functions to add windows, fences, chimneys, and other features to their houses. By the end of the first class, students had learned to use functions to create a variety of shapes. Important learning occurred around syntax, sequencing, and parameters.

Students at this age are just learning to graph coordinates. In the Khan Academy IDE, students can guess numbers for the parameters and then use the sliders to experiment. The program output is updated instantly. This form of playing with the x and y values makes the math exercise relevant and fun. In addition to coordinates, students explored other mathematical concepts, including functions, variables, and geometry. Once I had introduced iteration using the draw function, the students were able to create interactive art based on the mouse position variables.

Many students worked outside of class to create complex works of art through code. Using the cloud-based IDE made it easy for students to log in from home to continue working.

To facilitate learning beyond the classroom and to manage

*continued on page 8*

### SHOW ME THE NUMBERS

**PROJECTED STEM JOBS BY STATE IN 2018**

| Rank | State | Projected Number of Jobs |
|---|---|---|
| 1 | California | 1,148,000 |
| 2 | Texas | 758,000 |
| 3 | New York | 477,000 |
| 4 | Florida | 411,000 |
| 5 | Virginia | 404,000 |
| 6 | Illinois | 348,000 |
| 7 | Pennsylvania | 314,000 |

Learn more about STEM jobs projections for your state: *www.stemconnector.org/sites/default/files/store/STEM-Students-STEM-Jobs-Executive-Summary.pdf*

csta Voice

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

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

NATIONAL • SCIENCE • FOUNDATION

**PROGRAMMING THROUGH ART AND MATH**

*continued from page 7*

a large number of classes, I made extensive use of Edmodo, a moderated online classroom, where I post messages and engage in discussions. Students used the environment to share their work, comment on other projects, and ask questions.

I asked the students what they would say to convince others to learn programming; their answers were inspiring.

- *I would tell someone that coding is a type of art. There are no bounds to your creativity and you can create anything you want, it can be easy or hard.*
- *You should try coding! It's really fun, but I'm not promising that it will be easy. It's really useful and teaches patience.*
- *Coding uses all that I have learned in math and applies it to the real world.*

I have seen the value of connecting programming to both art and math; my students are enthusiastic about programming and my school administration is ready to expand the program. A recent, first-ever district coding competition attracted over 100 participants—and 58% were girls! CSTEM has convinced students in our district that CS can be challenging and fun. Required CS education at this age is both valuable and possible.

Sheena Vaidyanathan will discuss these ideas and share successful strategies during a session, *Programming through Art and Math*, at the CSTA Conference on July 16, 2013.

**LEARN MORE:**

CSTEM Los Altos School
*www.lasdschools.org/District/Portal/CSTEM*
Khan Academy *www.khanacademy.org/cs*
Processing class demos *www.computersforcreativity.com*
Edmodo *www.edmodo.com*



## MARK YOUR CALENDAR

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

**Tapestry Project Workshop**
July 10–12, 2013, West Lafayette, Indiana
*www.tapestryworkshop.org*

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

**Tapestry Project Workshop**
July 16–18, 2013, Miami, Florida
*www.tapestryworkshop.org*

**Bootstrap: Videogame Programming with Algebra**
July 17, 2013, Boston, Massachusetts
*www.bootstrapworld.org/workshops/*

**Teaching Java and Graphics Programming Fundamentals through Art and Game Creation**
July 21, 2013, Anaheim, California
*www.buildingsteam.org*

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

**Tapestry Project Workshop**
July 23–25, 2013, Norman, Oklahoma
*www.tapestryworkshop.org*

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

**Tapestry Project Workshop**
July 29–31, 2013, River Grove, Illinois
*www.tapestryworkshop.org*

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

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

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

**Check the most recent CSTA events on the CSTA website**
*csta.acm.org/ProfessionalDevelopment/sub/TeacherWorkshops.html*