Opening Keynote
8:30–9:45 am

Megan Smith – Invited

Concurrent Sessions
10:00–11:00 am

**Computer Science Principles Curricula**
*Owen Astrachan, Dan Garcia, Bennett Brown, Jeff Gray, Ralph Morelli, Nigamanth Sridhar and Baker Franke*
Several recent sessions at CSTA and elsewhere have provided details about the development, piloting, and exam format of the NSF/College Board Computer Science Principles (CSP) project. Other presentations have highlighted individual projects and implementations of the CSP course. These courses differ in many respects – from choice of programming language, to the degree of emphasis on the Internet, and more. As schools move to adopt the CSP framework, develop courses, and deliver them, teachers and administrators look to national models for curricula that can be adopted and adapted to suit local needs. In this session, we highlight six NSF- and private/non-profit-funded projects that have developed, piloted, and made available CSP curricula. The single-session, active format facilitates direct comparisons of the approaches.

**Hands on Computing in K-8 Classrooms**
*Sheena Vaidyanathan*
Introduce K-8 students to the power of computer programming through projects that sense and interact with the external world. Learn successful projects and best practices from a district wide STEM program that integrates physical computing into the curriculum using Bee-Bots in K-2, Lego WeDo and Scratch in grades 3-5 and Arduino in grades 6-8. Students understand abstract ideas in computer science as they move robots through a maze or as they use data from sensors to control motors, LEDs and buzzers. Show elementary school students how to use Scratch code to bring a Lego project to life. Teach middle school students how to build circuits on a breadboard and program them using the Arduino microcontroller. Physical computing projects are challenging and add hands-on excitement to any K-8 computer science curriculum.

**Teaching CS to Students with Learning Differences**
*Sarah Ciras*
Overview of Language Based Learning Differences as they apply to the Computer Science classroom. Strategies and tools will be presented to help you scaffold your classes and reach all kinds of learners.
Out of Your Seat Comp Sci: Coding Using the Kinect  
*Doug Berman*

We will demonstrate and look at code and coding for kinetic movement in the Computer Science classroom using the Microsoft Kinect, and a demonstration of student projects, as well as a brief review of some of the code behind them. We will look at interpreting skeletal data and also facial recognition. We’ll also look at voice recognition. We’ll take a look at some other kinetic movement instruments such as dancepad and/or LEAP motion. We’ve had some incredible projects over the years, and I’ll share how we’ve done it in our high school class. The concept of using your entire body as user input allows for a different type of thinking from your students. This session will include members of the audience interacting with my projects. I am also hoping to generate audience discussion about this type of out-of-your-seat experience.

**Sponsor Spotlight 1: TBD**

---

**Mini Sessions**

**Findings of the CSTA Assessment Task Force**  
*Aman Yadav, Padmaja Bandaru, David Burkhart, Daniel Moix and Eric Snow*

CSTA recently conducted a landscape study of CS assessment in US high schools. This session presents the findings of that study, yielding greater understanding of the current state of K-12 computer science assessment, challenges teachers face when evaluating student understanding of CS concepts, and the role of teachers in assessment development.

**CSTA Standards Crosswalk with ACM IT Learning Outcomes**  
*Elizabeth Hawthorne, Cindy Tucker and Melanie Williamson*

This session will discuss CSTA’s Crosswalk document that maps Level 3B of the 2011 CSTA Computer Science Standards ([http://csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K-12_CSS.pdf](http://csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K-12_CSS.pdf)) to the 2014 "ACM Competency Model of Core Learning Outcomes and Assessment for Associate-Degree Curriculum in Information Technology" that was produced by the ACM Committee for Computing Education in Community Colleges (www.capspace.org). The creation of this Crosswalk document was a collaborative effort between representatives from CSTA and CCECC working together to document the alignment of computing curriculum between high schools courses and community college programs of study. The aim of the presentation is for high school teachers to better understand core student learning outcomes taught in associate-degree IT programs and for community college faculty to better understand the computing concepts and topics taught in high school grades 9 through 12.

**New Life for Old Hardware, A Teaching and Service Opportunity**  
*Neil Plotnick*

Older computer hardware are frequently donated by parents, teachers and community businesses to schools. In return, students and non-profits could benefit from receiving working computer systems. Putting in place a regimented protocol for the acceptance, reconditioning and redistribution of donated computer equipment can have many benefits for the schools and their communities.

**Concurrent Sessions**

11:15 am – 12:15 pm

**Introducing the CSP-POGIL Project**  
*Tammy Pirmann and Brent Gray*

Introducing the CSP-POGIL project! POGIL is a student centered, inquiry based, proven pedagogy. The big ideas of Computer Science Principles can be taught via guided inquiry, regardless of the programming environment chosen. Attendees will have the opportunity to participate in a CSP-POGIL lesson followed by a Q&A with activity authors and beta testers.
Using Minecraft in the MS to Teach Programming
Bob Irving
Minecraft has taken the world by storm. Its quirky, 8-bit graphics and open gameplay have captivated an entire generation. Come to learn how we can harness this energy to teach programming.

Engaging Underrepresented Youth in Computer Science
Alexis Martin, Frieda McAlear, Dan Garcia, Jennifer Arguello, Tiffany Price and Omoju Miller
Given the disparities in computer science participation nationwide among underrepresented student populations, this panel will: (a) highlight the critical need for engaging underrepresented students in computer science and (b) discuss a set of programs, initiatives, and research aiming to address barriers and increase interest, participation, and success in computer science among underrepresented high school students. This panel represents a group of individuals who are leading promising initiatives for broadening participation in computer science (especially among underrepresented groups). This includes the development of a new AP Computer Science principles focused on engaging and inspiring students with computational thinking rather than solely Java programming, integrating culturally relevant pedagogical approaches to computer science instruction and curriculum, providing access to rigorous computer science sequences in out-of-school settings, and investing in organizations which can deliver computer science education broadly through innovative approaches (e.g., online courses).

CS Through Innovative Wearable Technologies
Stephanie Playton
Wait, you’re saying there is a way to integrate fashion design in CS? When observing a loss of interest in CS for girls, we introduced the LilyPad Arduino. What we found was much more than a rekindled interest in CS for this population. In this session, you will learn how integration of e-textiles with an Arduino based microcontroller was used with middle school and high school students through the engineering design process. Project samples will be shared with the group, as well as resources, tutorials, and information on putting student kits together.

Convincing Non CS Teachers to Teach CS Using Bootstrap
Emmanuel Shanzer, Floresa Vaughn and Marisa Brown
Do you want to help your students learn to code, but don’t know how to get started? You do not have to have a computer science background to engage students in learning the art of coding. Bootstrap is an easy way to implement pedagogy that teaches students how to code and develop word problem solving strategies. We will discuss how the Bootstrap curriculum has been used in our urban school setting by non-CS teachers, to teach coding and improve math skills for our students.

Computational Thinking: A Problem-Driven Approach
Gerald Rambally and Rodney Rambally
This presentation will focus on specific activities to foster computational thinking (CT) skills in secondary school computer science. Using example concepts from the secondary school mathematics curriculum; the presentation will demonstrate how key computational thinking skills including algorithmic thinking, problem reformulation, problem reduction, problem transformation, problem representation, exhaustive search, heuristic reasoning, backtracking, and recursion, can be fostered. The presentation presents a minds-on, seamless (i.e., no new content), problem-driven integration of mathematics and computer science to develop computational thinking skills which expose students to habits of mind beyond the standard mathematics curriculum, thereby providing them with a broader, firmer foundation. Furthermore, this approach does not extend the mathematics curriculum. Various types of problems in the existing secondary school mathematics curriculum are used to foster computational thinking. Through these experiences students learn many critical CT principles but more importantly, develop a cognitive model about computational processes.

Sponsor Spotlight 2: TBD
Concurrent Sessions
1:30–2:30 pm

Partners for a Path in K-12 Computer Science
*Pat Youngpradit, Owen Astrachan, Gail Chapman, Joanna Goode, Emmanuel Schnazer, Kiki Profftsman and Irene Lee*

Schools are demanding more direction in developing a path for students moving through successive experiences in computer science. Several programs have begun to scale out across the country, and with this expansion there has been an increase in questions. How do schools understand and navigate the increasing options out there to create a coherent whole? What are the key decision criteria for a K-12 program in computer science? How can multiple courses and programs combine to provide comprehensive teacher preparation and curriculum support across K-12? Code.org has brought these recognized programs together to bring CS to all schools and students in the nation. But just as Code.org can’t do it without their program partners, they can’t do it without the input and support of the community as well. This session will also allow attendees to provide input into Code.org’s overall national effort.

Blocks-based Programming Environments – A Toolbox for Many Occasions
*Josh Sheldon, Judy Perry, Paul Medlock-Walton and Daniel Wendel*

Blocks-based programming environments abound in today’s computer science education ecosystem. This session will examine four such environments developed at the MIT Center for Mobile Learning that share much common heritage and some common features, and yet are specialized for different purposes and learning outcomes. The presenters will compare and contrast MIT App Inventor, GameBlox, TaleBlazer, and StarLogo Nova, four unique blocks-based programming environments that allow: Android app development, web-based game development, mobile device locative augmented reality game building, and 3-D multi-agent simulation or game building, respectively. Each environment is informed by a rich history of computer science education going back to Logo and beyond, in which relative novices to programming are empowered to quickly make personally meaningful artifacts. However, each environment has also been carefully crafted to serve a specific purpose that is distinct from the other environments to be presented.

Recruiting and Retaining Diverse Student Groups in New Mexico Computer Science for all
*Maureen Psaila-Dombrowski, Lauren Curry, Jeremy Jensen and Vernetta Noble*

In this panel, we describe the New Mexico Computer Science for All program (NM-CSforAll) and accompanying classes that were designed to teach computer science and computational thinking through modeling and simulation in a broadly accessible manner. The program has demonstrated effectiveness in attracting and retaining a diverse range of teachers and students from dramatically different teaching environments and communities. The resulting diversity of students exposed to computer science through this program is unprecedented in New Mexico and in the country. The pedagogy, curriculum and flexible approach allowed students who might not normally take a computer science class to participate and succeed. The overall number and diversity of students who complete the class are evidence of its promise. The panel, representative of the different regions and cultural contexts, will discuss how elements of the program were customized to work for them and their students and lessons learned in the process.

From Online Professional Development to Classroom Pedagogy: Working at the Nexus of Research
*Aman Yadav, Sarah Gretter and Phil Sands*

The goal of PD4CS (Professional Development for Computer Science) is to establish an evidence-based professional development (PD) program to improve teachers' knowledge to teach Computer Science (CS) at the high school level. The PD4CS project team has been developing and implementing a high-quality professional development approach that incorporates online just-in-time support, in collaboration with Project Lead The Way (PLTW). This session will present initial observations gathered from the launch of the pilot online professional development website. An overview of the project will be given, as well as a demonstration of selected topics.
The aim of this session is to create a dialogue between CS teachers of various expertise levels and CS researchers by enhancing the teaching-research nexus as well as the advancement of professional development in CS.

**Focusing on Problem-Centered Programming Instruction**  
*Stephen Hughes and J. Philip East*

In 2013, we began discussing what programming instruction might look like if it focused on the problems being programmed rather than the language features being used. As a result, we have found or devised instructional strategies that greatly changed what we do, how we think, how we teach, and (we believe) what our students learn. Those strategies include: Emphasizing the use of natural language algorithm descriptions; Designing solutions that consider a division of labor between the user and program; Stressing the process over the product; Minimizing the introduction of language-based hypotheticals; Working with grounded examples and Actively striving to identify the source of problem-solving barriers. Using this approach has substantially changed our teaching behavior. We will share our experience and rationale for the approach.

**Sponsor Spotlight 3: TBD**

**Concurrent Sessions**  
**2:45–3:45 pm**

**Achieving a Shared Goal with AP Computer Science A and AP Computer Science Principles**  
*Fran Trees, Richard Kick, Andrew Kuemmel, Deepa Muralidhar, Dan Garcia, Lester Wainwright, Glen Martin, Sandy Czajka, Paul Tymann and Lien Diaz*

This session will provide an opportunity for AP CSA Development Committee members to share information regarding the AP CSA course and will discuss plans for collecting feedback from college faculty regarding course content and approaches to programming techniques and languages. AP Computer Science Principles DC members will then discuss the new AP CS Principles course and plans for the development and launch of the course and exam. The session will largely comprise a Q&A session addressing questions regarding both courses.

**Pencil Code: Bridging Between Visual Blocks and Text-Based CS Education**  
*David Bau, Cait Pickens and Matthew Dawson*

This workshop provides an in-depth look at some methods and strategies teachers can use to transition students between visual programming and text-based coding. We will work with the open source-coding environment Pencil Code, a tool with a dual-mode editor that switches between visual block programming and text programming seamlessly. The tool allows students to work with programs with any level of complexity using either blocks or CoffeeScript text code. Attendees will come away with an understanding of best practices when teaching students transitioning to text code for the first time, and they will gain practical experience with lessons involving functions and data. We will discuss our experiences teaching using the tool, and we will share curriculum material appropriate for middle school and high-school classrooms. The tool is open-source and free to use online at http://pencilcode.net/.

**Changing Computing Education Policy in Your State**  
*Renee Fall, W. Richards Adrion, Debra Richardson and Mark Guzdial*

Real change in the distributed US education system is made locally. National groups like Code.org and CSTA are generating enthusiasm for computing education, but making change at the state, district, and school level requires the active participation of individuals and local organizations to engage policy makers, superintendents, and communities. A reform process for improving computing education pathways will take place at multiple stages (from elementary/middle/high schools, community and four-year colleges, and universities) and state by state. The leaders in this session are changing computing education in California, Massachusetts, and Georgia. Their goal is to make computing education accessible to all students. They will present approaches that have worked in each state, and what’s not worked, to show how different states need different strategies. Most of the session is interactive: The audience will caucus by state, and will then pose state-specific questions to the panel.
Learning Computer Science Through Making
Laura Blankenship, Brie Daley and Ryan Barnes
The Maker/DIY movement has become popular everywhere, including in education. There are many overlaps between making and Computer Science. This session will explore ways to connect Computer Science to other disciplines through maker projects. Using example projects that have already been successful, we will share resources and ideas for creating projects that cover a variety of Computer Science topics. Using the CSTA and Common Core standards, we will show how many of these projects can cover multiple standards and be engaging and rewarding.

Sponsor Spotlight 4: TBD

Mini Sessions

TEALS (Technology Education and Literacy in Schools)
Michael Braun and Kevin Wang
You will learn about how TEALS (Technology Education And Literacy in Schools) can support your classroom or school. TEALS is a grassroots program that recruits, trains, mentors, and places high tech professionals from across the country into your classroom.

Effective Blended Learning in Computer Science
Kim Garcia
Engage students and build relationships by implementing blended learning in your computer science class. Using a learning management system, create an online forum where computer science students deepen their understanding through reviewing other students’ code, discussion, and reflection. Learn how to effectively blend face-to-face and online activities in your computer science class.

GenCyber Summer Camps for High School Teachers and Students
Josh Pauli
GenCyber summer camps are funded by the National Science Foundation (NSF), and the National Security Agency (NSA) to provide summer camps for high school teachers and students at locations across the nation. Following the lead from the StarTalk program, GenCyber is spreading cyber security instructional resources to teachers and exploratory sessions to students across the nation.

Closing Keynote
4:00–5:00 pm

GearBox: Randy Pitchford, Aaron Thibault and Jimmy Siebon
Description pending