TEACHING LEARNING
Raleigh, NC
SIGCSE 2012
COLLABORATING

The 43rd Technical Symposium on Computer Science Education

Contents

Message from the Symposium and Program Chairs

Welcome to the proceedings of the 43rd ACM Technical Symposium on Computer Science Education, or SIGCSE 2012, where you will find the archival record of over one hundred papers as well as multiple other session formats that document the latest in computer science education: research, tool building, teaching, curriculum and philosophy. Reflecting our location in North Carolina's renowned Research Triangle, this year's conference features the three themes of "Teaching, Learning, and Collaborating" throughout the sessions. Teaching, learning, and collaborating occur inside and outside of the classroom, and among various combinations of students, academics, industry professionals, and others.

We are pleased to announce the winners of the two annual SIGCSE awards. Professor Harold (Hal) Abelson of Massachusetts Institute of Technology will receive the SIGCSE award for Outstanding Contribution to Computer Science Education, and will provide Friday's keynote address. Jane Prey of Microsoft Research will accept the SIGCSE Award for Lifetime Service to the Computer Science Education Community will and speak at our First Timer's Lunch. In addition, we've invited speakers to deliver two more keynote presentations. Frederick P. Brooks, Jr. from University of North Carolina at Chapel Hill will deliver Thursday's keynote address titled "The Teacher's Job is to Design Learning Experiences; not Primarily to Impart Information." Fernanda Viégas and Martin Wattenberg from Google's "Big Picture" visualization group will jointly address Saturday's Luncheon with "Through the Looking Glass: Talking about the World with Visualization." Symposium statistics are presented on page 4. Many thanks to the authors, reviewers, and Program Committee members whose enormous and vital service generated this program. This year's program includes the usual wide selection of events, including the First-Timer's Lunch and Evening Reception on Thursday and the SIGCSE luncheon on Saturday. Our exhibit hall features a number of exhibitors showcasing the latest in hardware, software tools, textbooks and educational programs and research. We continue to offer accessibility at SIGCSE 2012 for the deaf and hard of hearing.

We are excited about the variety of pre-symposium events that we will be offering. As of the press deadline for this overview, meetings on the following topics will occur on Wednesday: Teaching Open Source, Career Mentoring, SIGCAS Open Meeting, Teaching Ethics in Computer Science, and Computing Accreditation. Near the end of the symposium on Saturday, undergraduate and graduate students will present their work at the ACM SIGCSE Student Research Competition, managed by Ann Sobel. SIGCSE 2012 will also see the return of robots through a new venue, "The SIGCSE Playground: Experience It!" Many thanks go to Jennifer Kay and Douglas Blank for organizing this event.

Our sincere thanks go out to the people who made this Symposium extraordinary. First, our symposium committee: Carl Alphonce, Adrienne Decker, Lynn Degler, John Dooley, Mary Anne Egan, Susan Fox, Michael T. Helmick, Olaf Hall-Holt, John Harrison, Sarah Heckman, Catherine Lang, Steven Huss-Lederman, Cary Laxer, Chuck Leska, Lester I. McCann, Scott McElfresh, Larry Merkle, Brad Miller, Briana Morrison, Kris Nagel, Sarah Monisha Pulimood, Susan Rodger, Kimberly Voll, and Henry Walker. Additional thanks go to our Associate Program Chairs who provided meta-reviews for papers: Joel Adams, Doug Baldwin, Tim Bell, Sheila Castañeda, Tim Fossum, Mark Guzdial, Sherri Goings, Nancy Kinnersley, Lori Pollock, and Sami Rollins. We'd also like to thank all of our student volunteers who help us with all of the small details.

In difficult economic times, we extend a *very* grateful thank you to our supporters, vendors, exhibitors and in-kind donors whose participation literally make the symposium possible. We especially thank Microsoft, Google, and Intel as platinum supporters, SAP University Alliances as a silver supporter, and Oracle for funding an ice cream break. Special thanks go to Dorothea Heck and her team at D. Lawrence Planners for coordinating an outstanding exhibition and to Susan Rodger, our amazing supporter/exhibitor liaison.

We thank SIGCSE President Renée McCauley and the entire SIGCSE Board for their support and guidance, and acknowledge the contributions of SIGCSE Symposium Site Coordinators Bob Beck and Scott Grissom, as well as Ashley Cozzi of ACM. Lisa Tolles of Sheridan Printing brought all materials together. The City of Raleigh provided valuable information and prizes through their excellent Convention and Visitor's Bureau, with special thanks to Julie Brakenbury. We were supported at the Raleigh Convention Center by Tim Greene, Mara Craft, Dave Chapman, Mason Hotaling, Dan Kapps (of Centerplate Catering), and Michael Murphy (of American AV); at the Raleigh Marriott City Center Hotel by Sasha Armour and Thomas Kelville; at the Sheraton Raleigh Hotel by Kevin Johnson; and at the Clarion Raleigh Hotel State Capital by Tricia Nelson.

Special thanks to our home institutions for providing needed resources: College of the Holy Cross, Carleton College, Colorado School of Mines, and Rochester Institute of Technology. We genuinely hope you enjoy the Symposium and find the SIGCSE 2012 Proceedings of use for your work now and your future projects and activities.

Symposium Chairs

Laurie Smith King, *College of the Holy Cross*
David R. Musicant, *Carleton College*

Program Chairs

Tracy Camp, *Colorado School of Mines*
Paul Tymann, *Rochester Institute of Technology*

SIGCSE 2012 Symposium Statistics

Proposal Type	Accepted	Received	Acceptance Rate
Paper	100	289	35%
Panel/Special Session	24	53	45%
Workshop	36	78	46%
Poster	48	109	44%
Birds of a Feather	36	46	78%

Number of Reviewers: 810 (plus 10 assoc program chairs, a.k.a meta-reviewers)
Number of Paper reviews received: 1608, plus 289 meta-reviews
Number of reviews (including meta-reviews) assigned to each paper: 6.6
Number of papers with 6 or more regular reviews: 163
Number of papers with 5 regular reviews: 126

SIGCSE 2012 Award Winners

Award for Outstanding Contribution to Computer Science Education
Harold (Hal) Abelson, *Massachusetts Institute of Technology*
Award for Lifetime Service to the Computer Science Education Community
Jane Prey, *Microsoft Research*

Symposium Evaluations Link

`http://www.sigcse.org/sigcse2012/attendees/evaluations.php`

Convention Center WiFi

Network: SIGCSE Access Code: SIGCSE2012 (all uppercase)

SIGCSE 2012 Symposium Committee

Symposium Chairs	David R. Musicant, *Carleton College*
	Laurie Smith King, *College of the Holy Cross*
Program Chairs	Tracy Camp, *Colorado School of Mines*
	Paul Tymann, *Rochester Institute of Technology*
Associate Program Chairs	Doug Baldwin, *SUNY Geneseo*
	Shiela Casteneda, *Clarke University*
	Sherri Goings, *Carleton College*
	Joel Adams, *Calvin College*
	Tim Fossum, *SUNY Potsdam*
	Nancy Kinnersley, *University of Kansas*
	Lori Pollock, *University of Delaware*
	Tim Bell, *University of Canterbury*
	Chuck Leska, *Randolph-Macon College*
	Sami Rollins, *University of San Francisco*
	Mark Guzdial, *Georgia Institute of Technology*
Workshops	Adrienne Decker, *Rochester Institute of Technology*
	Lester I. McCann, *University of Arizona*
Birds of a Feather	Olaf Hall-Holt, *St Olaf College*
Posters	Kris Nagel, *Georgia Gwinnett College*
Panels and Special Sessions	Chuck Leska, *Randolph Macon University*
Registration	Cary Laxer, *Rose-Hulman Institute of Technology*
	Lynn Degler, *Rose-Hulman Institute of Technology*
	Larry Merkle, *Wright State University*
Publications	Brad Miller, *Luther College*
Treasurer	Scott McElfresh, *Wake Forest University*
Student Volunteers	Mary Anne Egan, *Siena College*
	Steven Huss-Lederman, *Beloit College*
Evaluations	Carl Alphonse, *University at Buffalo, SUNY*
Kids Camp Coordinators	Susan Fox, *Macalester College*
	Sara Monisha Pulimood, *The College of New Jersey*
Database Administrators	Henry Walker, *Grinnell College*
	John Dooley, *Knox College*
International Liason	Catherine Lang, *Swinburne University*
K-12 Liason	John Harrison, *Princess Ann High School*
Local Arrangements	Sarah Heckman, *NC State University*
Publicity / Social Networking	Kimberly Voll, *University of British Columbia*
Student Research Competition	Ann Sobel, *Miami University (Ohio)*
Pre-Conference Event Liason	Briana Morrison, *Southern Polytechnic State University*
Supporter/Exhibitor Liaison	Susan Rodger, *Duke University*
Webmaster	Michael T. Helmick, *Google*

Thursday Schedule

Wednesday, 19:00 to 22:00 Workshops

Thursday, 8:30 to 10:00 Ballroom BC, Keynote: Frederick P. Brooks
Thursday, 10:00 to 10:45 Exhibit Hall A, Break and Exhibits

Thursday, 10:45 to 12:00

301AB	Panel	Computer Curricula 2013: Update
305B	Panel	Scrum Across the CS/SE Curricula
306C	SS	Role of Interdisciplinary Computing in Higher Education, Research and Industry
302A	Papers	Data Structures and Algorithms
302B	Papers	Robots
306A	Papers	K-6 Collaborations
306B	Papers	Tools
302C	Supporter	Intel – Teaching Parallelism Lightning Rounds
305A	Supporter	Microsoft – Empowering Students: Teaching Software Development with Windows Phone

Thursday, 12:00 to 13:45 Marriot State CDEF, First Timer's Lunch

Thursday, 13:45 to 15:00

301AB	Panel	A Stratified View of Programming Language Parallelism for Undergraduate CS Education
305B	SS	Demystifying Computing with Magic
306C	Panel	Community-Based Projects for Computing Majors: Opportunities, Challenges and Best Practices
302A	Papers	Games
302B	Papers	Professional Experiences
306A	Papers	A Session with a View
306B	Papers	Pedagogy: Programming
305A	Supporter	Microsoft – Creative Uses for Kinect in Teaching – with Curriculum Materials

Thursday, 15:00 to 15:45 Exhibit Hall A, Break and Exhibits

Thursday, 15:45 to 17:00

301AB	Panel	Science Fiction in Computer Science Education
305B	Panel	Diversity Initiatives to Support Systemic Change in Undergraduate Computing
306C	SS	Transforming the CS Classroom with Studio-Based Learning
302A	Papers	Broadening Participation
302B	Papers	Online Collaboration
306A	Papers	Middle School Collaborations
306B	Papers	New Tricks for the Classroom
305A	Supporter	Google – All Things Google and Education

Thursday, 17:10 to 18:00 BoF Flock I
Thursday, 18:10 to 19:00 BoF Flock II
Thursday, 19:00 to 20:00 Registration Foyer, SIGCSE Reception

Friday Schedule

Friday, 7:15 to 8:15 302B, Alice Breakfast

Friday, 8:30 to 10:00 Ballroom BC, Keynote: Hal Abelson

Friday, 10:00 to 10:45 Exhibit Hall A, Break and Exhibits

Friday, 10:00 to 12:00 Exhibit Hall A, Poster Session I

Friday, 10:45 to 12:00

301AB	Panel	Teaching Mathematical Reasoning Across the Curriculum
305B	SS	Teaching HS Computer Science as if the Rest of the World Existed
306C	SS	Funding the Challenges in Computing
302A	Papers	CS1: New Ideas
302B	Papers	Team Work
306A	Papers	Summer Experiences
306B	Papers	Software Engineering
305A	Supporter	Intel – Academic Assembly: What are the next imperatives?

Friday, 12:00 to 13:45 Lunch Break

Friday, 12:10 to 13:35

301AB		Snap! Lunch
302A		UPE National Meeting

Friday, 13:45 to 15:00

301AB	SS	CS Principles: Piloting a National Course
305B	SS	Fun, Phone, and the Future - Microsoft XNA Game Studio, Windows Phone, and Kinect SDK
306C	SS	Building an Open, Large-Scale Research Repository of Initial Programming Student Behavior
302A	Papers	Collaborative Learning
302B	Papers	Curriculum Issues
306A	Papers	Active Learning I
306B	Papers	Communication Skills
305A	Supporter	Google – The MIT Center for Mobile Learning and the Future of App Inventor

Friday, 15:00 to 15:45 Exhibit Hall A, Break and Exhibits

Friday, 15:00 to 17:00 Exhibit Hall A, Poster Session II

Friday, 15:45 to 17:00

301AB	SS	Understanding NSF Funding Opportunities
305B	Panel	Teaching Outside the Text
306C	SS	Computer Engineering Review Task Force Report
302A	Papers	Projects
302B	Papers	Alice and Scratch
306A	Papers	Active Learning II
306B	Papers	Non-majors
305A	Supporter	Microsoft – Cloud in a Classroom: Faculty Experiences

Friday, 17:10 to 17:55 302A SIGCSE Business Meeting

Friday, 18:00 to 18:45

302C		Going Greenfoot
305B		CCSC Business Meeting
305A		NCWIT Academic Alliance reception

Friday, 19:00 to 22:00 Workshops

Saturday Schedule

Saturday, 8:30 to 9:45

301AB	SS	Nifty Assignments
305B	SS	Update on the CS Principles Project
306C	Panel	Implementing Evidence-Based Practices makes a Difference in Female Undergraduate Enrollments

Saturday, 8:30 to 10:10

302A	Papers	High School Collaborations
302B	Papers	Parallelism and Concurrency
306A	Papers	Mobile Computing
306B	Papers	Visualization
302C	Poster	Student Research Competition - Graduate
305A	Poster	Student Research Competition - Undergraduate

Saturday, 10:10 to 10:55 Exhibit Hall A, Break and Exhibits

Saturday, 10:55 to 12:10

301AB	Panel	Rediscovering the Passion, Beauty, Joy, and Awe: Making Computing Fun Again
305B	SS	Promoting Student-Centered Learning with POGIL
306C	SS	Teaching Secure Coding - Report from Summit on Education in Secure Software
302A	Papers	Attracting Majors
302B	Papers	OS and Distributed Computing
306A	Papers	Curricular Innovations and Research
306B	Papers	CS Education Research

Saturday, 12:30 to 14:30, Ballroom BC, SIGCSE Luncheon
Keynote: Fernanda Viégas and Martin Wattenberg

Saturday, 15:00 to 18:00 Workshops

General Schedule Notes

Registration is Open:

- Wed 3:00 PM – 9:30 PM
- Thu 7:30 AM – 8:00 PM
- Fri 7:30 AM – 5:00 PM
- Sat 8:00 AM – 12:15 PM and 2:30 PM - 3:00 PM

Registration and the Registration Foyer are in the Main Lobby of the 300 level of the conference center.

Exhibits are Open

- Thu 10:00 AM – 5:00 PM
- Fri 10:00 AM – 5:00 PM
- Sat 9:30 AM – 12:00 PM

NSF Project Showcase

1. Thu 10:00 – 11:30, Exhibit Hall A
2. Thu 3:00 – 4:30, Exhibit Hall A
3. Fri 10:00 – 11:30, Exhibit Hall A
4. Fri 3:00 – 4:30, Exhibit Hall A
5. Sat 10:00 – 11:30, Exhibit Hall A

The ACM Student Research Competition

- Thursday 1:45 – 5:15, Exhibit Hall A
- Saturday 8:30 – 10:10; 302C - grad, 305A - Undergrad

K-12 Teachers room 202 10-5 Thursday, Friday
CS Education Research room 203 10-5 Thursday, Friday
SIGCSE Business Meeting, Friday 5:10-5:55
The SIGCSE Reception, Thursday evening 7:00 – 8:00

Wednesday, 19:00 to 22:00

1. Using Social Networking to Improve Student Learning Through Classroom Salon 201

John Barr, *Ithaca College*; Ananda Gunawardena, *Carnegie Mellon University*

This workshop introduces an innovative social collaboration tool called Classroom Salon (CLS). Developed at Carnegie Mellon University, CLS is a combination of electronic books, social networks, and analytic tools. It enables students to learn by participating in social networks and allows instructors to easily analyze student participation. The workshop covers extant social networks, introduces CLS web-based software (nothing to install) and demonstrates the use of CLS to help students master critical skills such as code review, debugging, and reading documentation. Participants will create Salons, learn how to use them in their courses, and learn how to use the built-in tools to analyze student activities. See http://classroomsalon.org. Laptop (with wifi) required.

2. Challenges and Opportunities in Conducting Educational Research in the Computer Science Classroom 204

Aman Yadav and Tim Korb, *Purdue University*

This workshop will provide CS educators with tools to conduct educational research. Primary objectives of this workshop are: (1) learn basic principles of research design; (2) learn about various types of research designs: qualitative vs. quantitative; experimental vs. quasi-experimental; case studies, survey; (3) to practice designing research. This workshop will help participants make informed decisions when faced with limitations of educational research and collect empirical evidence about what works in the classroom. In addition, we will also discuss how to develop robust student outcome measures, such as surveys and tests. The workshop will be beneficial to participants who have not yet done all of these activities as well as those who have some background in educational research.

3. C++11 in Parallel 205

Joe Hummel, *U. of California, Irvine*

As hardware designers turn to multi-core CPUs and GPUs, software developers must embrace parallel programming to increase performance. No single approach has yet established itself as the “right way” to develop parallel software. However, C++ has long been used for performance-oriented work, and it’s a safe bet that any viable approach involves C++. This position has been strengthened by ratification of the new C++0x standard, officially referred to as “C++11”. This workshop will introduce the new features of C++11 related to parallel programming, including type inference, lambda expressions, closures, multithreading support, and thread-local storage. It will close with discussion of other technologies, including Intel TBB, ArBB, Cilk Plus, and Microsoft PPL, AAL, AMP. Laptop Optional.

4. The Absolute Beginner's Guide to JUnit in the Classroom 206

Stephen Edwards and Manuel Pérez-Quiñones, *Virginia Tech*

Software testing has become popular in introductory courses, but many educators are unfamiliar with how to write software tests or how they might be used in the classroom. This workshop provides a practical introduction to JUnit for educators. JUnit is the Java testing framework that is most commonly used in the classroom. Participants will learn how to write and run JUnit test cases; how-to's for common classroom uses (as a behavioral addition to an assignment specification, as part of manual grading, as part of automated grading, as a student-written activity, etc.); and common solutions to tricky classroom problems (testing standard input/output, randomness, main programs, assignments with lots of design freedom, assertions, and code that calls exit()). Laptop Recommended.

5. Student Scrums 301A

Thomas Reichlmayr, *Rochester Institutue of Technology*

Our students are entering the workforce into an increasing number of companies using Agile processes and practices in the development of their products and services, with Scrum being the most widely used Agile project management framework. Selecting Scrum as the framework for student team projects has the advantage of introducing software process at a level of ceremony that captures foundational software engineering practices and is manageable within the constraints of a class or capstone project. This workshop will introduce participants to the components of the Scrum framework with activities designed to demonstrate the flexibility of Scrum to support a diverse set of course learning outcomes at all levels of the curriculum. Laptop Optional.

6. Reviewing NSF Proposals: Learn about Effective Proposal Writing via the Review Process 301B

Sue Fitzgerald and Guy-Alain Amoussou, *National Science Foundation*

This interactive workshop focuses on the National Science Foundation grant proposal review process. Via close examination of the review process, participants gain an understanding of how to write good reviews and how to improve their own proposal writing. The workshop topics include: the proposal review process from submission to award or decline; elements of a good review; NSF merit criteria (intellectual merit and broader impacts); elements of good proposals; how to volunteer to review. Faculty who wish to understand the NSF review process or seek funding in support of undergraduate education are encouraged to attend. Participants will include novice proposal writers and those with more experience who seek to improve their proposal writing and reviewing skills. Laptop optional.

7. A Hands-On Comparison of iOS vs. Android 302A

Michael Rogers, *Northwest Missouri State University*; Mark Goadrich, *Centenary College of Louisiana*

This workshop is designed for faculty, considering teaching a course in mobile app development, who are unsure as to whether they should use iOS, Android, or both. To help them make an educated decision, in this workshop participants will build one app, to implement the game Pig, in both platforms. By so doing, they will be able to make a head-to-head comparison of the respective development environments, languages, and frameworks, guided by experienced instructors. Participants will need to bring (or share) a recent-vintage MacBook Pro / MacBook Air, with Xcode, Eclipse, and appropriate SDKs, installed prior to the workshop. Details, including installation instructions, may be found at androidios.goadrich.com. Laptop Required.

8. Killing 3 Birds with One Course: Service Learning, Professional Writing, and Project Management 302B

Joseph Mertz, *Carnegie Mellon University*; Scott McElfresh, *Wake Forest University*; Steven Andrianoff and Jennifer Dempsey, *St. Bonaventure University*

Service learning is a great idea, but can be fraught with problems. We present an alternative to the project- course approach. Instead of team-based system- development, we use a student- consultant model. Students individually consult with a nonprofit. Each student leads a small technology project that brings about sustainable change in an organization, while developing analysis, planning, and communication skills. One instructor can manage 30 clients a semester, and we have had nearly 400 to date. Our clients are happy and recruit others. In this session we will share our tricks: managing a large number of partnerships, helping students develop leadership and communication skills, and assessing their performance. A student presenter will describe her consulting experience. Laptop Optional

9. Computer Science Unplugged, Robotics, and Outreach Activities 302C

Tim Bell, *University of Canterbury*; Daniela Marghitu, *Auburn University*; Lynn Lambert, *Christopher Newport University*

You've been asked to talk to an elementary or high school class about Computer Science, but how can you ensure that the talk is engaging? Perhaps you're trying to introduce a concept from Computer Science to a school group, but you want a fun way to get the class engaged. This workshop is a hands-on introduction to Computer Science Unplugged (www.csunplugged.org), a widely used set of kinesthetic, fun activities that cover many core areas of computer science without using high technology. We will explore how to use the activities in a variety of situations, including combining them with robotics activities, and explore some novel applications. Attendees will receive a CD with a copy of a handbook for teachers and a collection of videos demonstrating the activities. Laptop Optional.

10. Introduction to Using FPGAs in the Computer Science Curriculum 307

William Jones and Brian Larkins, *Coastal Carolina University*

One of the challenges in modern curriculum design is balancing between breadth and depth of topics while simultaneously reinforcing the interconnections among topics in the field. We have integrated field-programming gate arrays (FPGA) systems first used in our hardware-based courses into several higher-level systems and applications courses. This allows us to leverage student familiarity with a hands-on, hardware platform and also strengthen the relationships between different subfields within computer science. In this workshop, we present participants with guided hands-on activities for making use of FPGAs in common computer science courses. Laptop Required.

11. Helping Students Become Better Communicators 305A

Janet Burge, Paul Anderson, and Gerald Gannod, *Miami University*

To be successful, CS and SE graduates need strong communication skills (writing, speaking, and teaming), particularly within their discipline. Students exercise these skills during their classes but are not always given explicit domain-specific instruction on these skills, instead relying on instruction provided outside the program. CS and SE faculty are not always comfortable in evaluating these aspects of their assignments and are often unhappy with the results. In this workshop we will lead sessions on teaching writing, speaking, and teaming; situating assignments in workplace- scenarios; and writing communication rubrics that convey faculty expectations to students and support evaluation of student work. For more information, see www.muohio.edu/sigcse_workshop11. Laptop Recommended.

12. ROS for Educators: Teaching with the Robot Operating System and Microsoft Kinect 305B

Michael Ferguson, *Willow Garage, Inc.*; Julian Mason, *Duke University*; Sharon Gower Small, *Siena College*; Zachary Dodds, *Harvey Mudd College*

The Microsoft Kinect and Willow Garage's Robot Operating System (ROS) are changing the way robots are developed. Together, these tools can enable today's CS educators to provide richer and more research-representative experiences with robots and perception. This hands-on workshop will introduce ROS and showcase two pilot courses taught using ROS and the Kinect. Four 20- minute talks will intersperse with participants' hands-on development of Python programs on low-cost Kinect-equipped robots and the ARDrone quadcopter. This workshop is intended for all college-level CS educators interested in robotics or embodied AI. First-time ROS/Kinect users are particularly welcome! Laptops and robots will be provided. See http://www.ros.org/wiki/Courses/sigcse2012. Laptops optional.

13. Board Game Project Ideas for CS 1 and CS 2 306A

Zachary Kurmas, *Grand Valley State University*; James Vanderhyde, *Benedictine College*

Participants will have fun learning and playing relatively unknown board games that are especially suitable for programming projects. We will present games where (1) all players can view the same screen, (2) the board is reasonably simple to program, and (3) there are several elements of the game that relate strongly to a common CS 1, CS 2, or discrete math topic. After we explain the rules and highlight the CS-related elements of the games, participants will have the opportunity to play the games, ask questions, and suggest rule variations that will improve the resulting programming project. See http://www.cis.gvsu.edu/~kurmasz/GamesWorkshop/ for more details and a list of games that may be presented. Laptop Optional.

14. A Taste of Linked Data and the Semantic Web 306B

Marsha Zaidman and David Hyland-Wood, *University of Mary Washington*

The Web has created a global information space of linked documents. The Semantic Web creates an information space of linked data from multiple sources. Information can be mined from the interlinking of available datasets by a distributed query language known as SPARQL, the SQL equivalent for the Semantic Web. Participants will understand and appreciate the role of linked data on the Semantic Web; be able to model, represent, and interpret simple linked data applications; complete exercises that create simple Linked Data models; appreciate the benefits of Linked Data over relational database modeling; be aware of successful commercial applications of linked Data; be directed to resources that facilitate incorporation of this material into their courses. WiFi/Laptop Required.

15. Teaching with Greenfoot and the Kinect – A Novel Way to Engage Beginners 306C

Michael Kölling and Neil Brown, *University of Kent*

The Microsoft Kinect is a sensor module that allows accurate tracking of humans moving in front of it. Greenfoot is an introductory Java programming environment that makes it easy to create animated graphical projects. By combining Greenfoot and the Kinect students can write programs where the user's body is used for input. Users interact with games by waving their hands, jumping, running, dancing, …. These kinds of programs are incredibly good fun and engage target groups who would not normally be interested in programming. The workshop is aimed at teachers of introductory programming courses (high school/university) who have some programming experience and want to incorporate new kinds of projects into their teaching. Laptop recommended but not required. Kinect hardware will be provided.

Thursday, 8:30 to 10:00

Plenary Session and Keynote — Ballroom BC

Welcome

Laurie Smith King, Symposium Co-Chair, *College of the Holy Cross*
Dave Musicant, Symposium Co-Chair, *Carleton College*

2012 SIGCSE Lifetime Service Award

Jane Prey, *Microsoft Research*

2012 SIGCSE Outstanding Contributions to Computer Science Education:

Harold (Hal) Abelson, *Massachusetts Institute of Technology*

Keynote Address: The Teacher's Job is to Design Learning Experiences; not Primarily to Impart Information

Frederick P. Brooks, Jr. *UNC Chapel Hill*

The primary job of the teacher is to make learning happen; that is a design task. Most of us learned most of what we know by what we did, not by what we heard or read. A corollary is that the careful designing of exercises, assignments, projects, even quizzes, makes more difference than the construction of lectures. A second corollary is that project courses that go deeply into narrow aspects of a subject seem to stick longer and deeper than approaches aiming at comprehensive coverage. How to strike a balance? I've taught a first software engineering laboratory course 22 times, and an advanced computer architecture course about ten times. Here are some techniques that work for me.

Thursday, 10:00 to 10:45

Break and Exhibits — Exhibit Hall A

Thursday, 10:00 to 11:30

NSF Showcase #1 — Exhibit Hall A

- **Tools and Best Practices for Improving the Quality of Students' Commenting Skills** Peter DePasquale, Michael Locasto, and Lisa Kaczmarczyk
- **Student Participation in Humanitarian FOSS** Heidi Ellis and Gregory Hislop
- **Computing in the Arts: A Model Curriculum** Bill Manaris and Renee McCauley
- **Integrating Ethics Into Computer Science Courses: Looking for Statistically Significant Effects** Keith W. Miller, Michael Loui, Mary Tracy, and Ken Urban

Thursday, 10:45 to 12:00

PANEL Computer Curricula 2013: Update 301AB

Chair: Mehran Sahami, *Stanford University*

Participants: Steve Roach, *University of Texas at El Paso*; Ernesto Cuadros-Vargas, *San Pablo Catholic University*; David Reed, *Creighton University*

Beginning over 40 years ago with the publication of Curriculum 68, the major professional societies in computing–ACM and IEEE-Computer Society–have sponsored various efforts to establish international curricular guidelines for undergraduate programs in computing. In the Fall of 2010, work on the next volume in the series, Computer Science 2013 (CS2013), began. Considerable work on the new volume has already been completed and a first draft of the CS2013 report (known as the Strawman report) will be complete by the beginning of 2012. This panel seeks to update and engage the SIGCSE community in providing feedback on the Strawman report, which will be available shortly prior to the SIGCSE conference.

PANEL Scrum Across the CS/SE Curricula 305B

Chair: Mark Hoffman, *Quinnipiac University*

Participants: Charles Wallace, *Michigan Technological University*; Douglas Troy, *Miami University*; Sriram Mohan, *Rose-Hulman Institute of Technology*

Scrum is one of the many agile approaches to software development that have been widely adopted over the past decade. Key agile features of Scrum are a flexible, adaptive schedule; democratic, collaborative teams; and frequent, iterative project and process reviews. Scrum is not only a software development strategy but a general learning strategy. Panel participants will describe how their students learn and practice Scrum in a software development, how they use it to manage student projects in non-software development contexts, and how Scrum provides opportunities to integrate communication skills into the CS and SE curricula. As participants in the CPATH II project, panelists have developed Scrum-based assignments that exercise the skills of reading, writing, speaking and teaming.

PANEL Role of Interdisciplinary Computing in Higher Education, Research and Industry 306C

Chair: Ursula Wolz, *Franklin W. Olin College of Engineering*

Participants: Lillian (Boots) Cassel, *Villanova University*

At SIGCSE 2010 NSF directors held a panel on the potential for Interdisciplinary Computing. This session is a direct outgrowth. Via an NSF grant individuals drawn from a range of universities and industry met three times in 2011 to discuss the nature of interdisciplinary computing, its importance both for computing and for other disciplines, obstacles to the further emergence of interdisciplinary computing and ways in which these obstacles might be overcome. This session provides an opportunity for the SIGCSE community to learn about the potential, promise and pitfalls of existing interdisciplinary computing activities. They will be asked to contribute their insights and experiences via structured small-group discussion, and will make connections with others with similar interests.

PAPERS Data Structures and Algorithms 302A

Chair: Ivona Bezakova, *Rochester Institute of Technology*

10:45 Sustainability Themed Problem Solving In Data Structures And Algorithms

Ali Erkan, Tom Pfaff, Jason Hamilton and Michael Rogers, *Ithaca College*

During the past two years, we have been creating curricular material centered around complex problems rooted in sustainability. Since multi-disciplinary learning is one of our primary goals, these projects are most meaningful when they connect students from different disciplines working toward a common understanding. However, strong disciplinary components present in their solutions also allow us to frame these projects from strictly disciplinary perspectives. In this paper, we show how they can be used for increased engagement in the context of data structures and algorithms. We review two new ones to explore (i) the structural characteristics of the western part of the U.S. power-grid, and (ii) the effects of over-harvesting on fish stocks.

11:10 Metaphors and Analogies for Teaching Algorithms

Monika Steinova, *ETH Zurich*; Michal Forisek, *Comenius University*

In this paper we explore the topic of using metaphors and analogies in teaching algorithms. We argue their importance in the teaching process. We present a selection of metaphors we successfully used when teaching algorithms to secondary school students. We also discuss the suitability of several commonly used metaphors, and in several cases we show significant weaknesses of these metaphors.

11:35 Detecting and Understanding Students' Misconceptions Related to Algorithms and Data Structures

Holger Danielsiek, Wolfgang Paul and Jan Vahrenhold, *Technische Universität Dortmund*

We describe the first results of our work towards a concept inventory for Algorithms and Data structures. Based on expert interviews and the analysis of 400 exams we were able to identify several core topics which are prone to error. In a pilot study, we verified misconceptions known from the literature and identified previously unknown misconceptions related to Algorithms and Data Structures. In addition to this, we report on methodological issues and point out the importance of a two-pronged approach to data collection.

PAPERS Robots 302B

Chair: Sherri Goings, *Carleton College*

10:45 **A C-based Introductory Course Using Robots**
Henry Walker, David Cowden, April O'Neill, Erik Opavsky
and Dilan Ustek, *Grinnell College*

Using robots in introductory computer science classes has recently become a popular method of increasing student interest in computer science. This paper describes the development of a new curriculum for a CS 2 course, Imperative Problem Solving and Data Structures, based upon Scribbler 2 robots with standard C. The curriculum contains eight distinct modules with a primary topic theme, readings, labs, and project at the end. Each module resulted from collaboration among former CS 2 students and a faculty member, utilizing an iterative process with revisions. Each lab includes a survey to obtain student feedback that will allow the course to evolve and better fit the needs of future CS 2 students. All materials discussed here are available online for use by others.

11:10 **dLife: A Java Library for Multiplatform Robotics, AI and Vision in Undergraduate CS and Research**
Grant Braught, *Dickinson College*

dLife is a free and open-source Java library that supports undergraduate education and research involving robotics, artificial intelligence, machine learning and computer vision. The design of dLife addresses many concerns raised by experience reports in the CS education literature including a shortened code/test/debug cycle, ready access to robot sensor information and close integration with a robotic simulation system. Support is currently provided for a variety of popular educational and research robots. Easily extensible packages for neural networks, genetic algorithms, reinforcement learning and computer vision support both classroom and research applications.

11:35 **Seven Big Ideas in Robotics, and How To Teach Them**
David S. Touretzky, *Carnegie Mellon University*

Following the curriculum design principles of Wiggins and McTighe (Understanding by Design, 2nd Ed., 2005), I present seven big ideas in robotics that can fit together in a one semester undergraduate course. Each is introduced with an essential question, such as "How do robots see the world?" The answers expose students to deep concepts in computer science in a context where they can be immediately put to the test. Hands-on demonstrations and labs using the Tekkotsu open source software framework and robots costing under $1,000 facilitate mastery of these important ideas. Courses based on parts of an early version of this curriculum are being offered at Carnegie Mellon and several other universities.

PAPERS K-6 Collaborations 306A

Chair: Sheila Castañeda, *Clarke University*

10:45 Design and Evaluation of a Braided Teaching Course in Sixth Grade Computer Science Education

Arno Pasternak, *Fritz-Steinhoff-Gesamtschule Hagen and Technische Universität Dortmund*; Jan Vahrenhold, *Technische Universität Dortmund*

We report on the design and evaluation of the first year of a CS course in lower secondary education that implements the concept of braided teaching. Besides being a proof-of-concept, our study demonstrates that students can indeed be taught CS (as opposed to ICT) as early as in sixth grade while at the same time not falling behind with respect to IT Literacy. We present quantitative and qualitative results and argue that Computer Science can be taught just like any other Science worth full curriculum credit.

11:10 Parallel Programming in Elementary School

Chris Gregg, Luther Tychonievich, Kim Hazelwood and James Cohoon, *University of Virginia*

Traditional introductory programming classes focus on teaching sequential programming skills using conventional programming languages and single-threaded applications. Students rarely learn about parallel programming until much later in their careers. Today, there is a greater need for programmers who are not only proficient in parallel programming, but who are not burdened by previously learned sequential programming habits, with parallelism tacked on as an afterthought. We present an introductory parallel programming course we taught to a group of primary school students using a novel parallel programming language. We provide examples of student-written code and we describe the overall course goal and specific lesson plans geared towards teaching students how to “think parallel.”

11:35 Building Upon and Enriching Grade Four Mathematics Standards with Programming Curriculum

Colleen M. Lewis and Niral Shah, *University of California, Berkeley*

We found that fifth grade students’ performance on Scratch programming quizzes in a summer enrichment course was highly correlated with their scores on a standardized test for Mathematics. We identify ways in which the programming curriculum builds upon target skills from the California state Mathematics standards to help understand opportunities for building upon and enriching Mathematics content through programming curriculum.

PAPERS Tools 306B

Chair: Sage Miller, *Webster Central School District*

10:45 Calico: A Multi-Programming-Language, Multi-Context Framework Designed for Computer Science Education

James Marshall, *Sarah Lawrence College*; Douglas Blank and Mark Russo, *Bryn Mawr College*; Jennifer S. Kay, *Rowan University*; Keith O'Hara, *Bard College*

The Calico project is a multi-language, multi-context programming framework and learning environment for computing education. This environment is designed to support several interoperable programming languages (including Python, Scheme, and a visual programming language), a variety of pedagogical contexts (including scientific visualization, robotics, and art), and an assortment of physical devices (including different educational robotics platforms and a variety of physical sensors). In addition, the environment is designed to support collaboration and modern, interactive learning. In this paper we describe the Calico project, its design and goals, our prototype system, and its current use.

11:10 How a Language-based GUI Generator Can Influence the Teaching of Object-Oriented Programming

Prasun Dewan, *University of North Carolina*

We have built a language-based direct-manipulation user-interface generator that can change, and we argue, improve the lectures and assignments on programming conventions, methods, state, constructors, preconditions, MVC, polymorphism, graphics, structured objects, loops, concurrency, and annotations. Our generator has several novel features for teaching such as interactive instantiation of a class, interactive invocation of methods and constructors that take arbitrary arguments, visualization of objects representing records, sequences, table and graphics, use of preconditions to disable/enable user-interface elements, and automatic generation of model threads.

11:35 CodeWave: A Real-Time, Collaborative IDE for Enhanced Learning in Computer Science

Jason Vandeventer and Benjamin Barbour, *University of North Carolina Wilmington*

Computer science instructors often rely on the final version of a program for assessment and feedback. This ignores the process the student used to arrive at the final program. When the instructor has the ability to observe real-time development progress of each student, they are better equipped to provide appropriate and timely feedback. CodeWave, a software program developed at the University of North Carolina Wilmington looks to alleviate these issues. CodeWave is a real-time, collaborative Integrated Development Environment with traditional features such as syntax highlighting and non-traditional features such as integrated messaging and logged playback. CodeWave enhances productivity by integrating many common tools students and instructors use during the programming process.

SUPPORTER SESSION Intel 302C

Teaching Parallelism: Lightning Rounds

The Teaching Parallelism Lightning Rounds special session is a place to meet and exchange ideas between those beginning the process of incorporating parallelism into their classrooms and curriculum and those already moving down that pathway. This session will build on the popular and exciting Ignite micro-presentations pioneered by O'Reilly (http://ignite.oreilly.com/)

The session will be in two parts: a set of short presentations (5 minutes and 5 slides each) highlighting recent experiences, followed by a community discussion and swap meet about other techniques, tweaks, and opportunities for including parallelism as part of the regular CS curriculum. Each presenter will answer the question "How have I brought parallelism into the computer science classroom?" as well as to provide examples of their content, techniques, tools etc.

SUPPORTER SESSION Microsoft 305A

Empowering Students: Teaching Software Development with Windows Phone

Rob Miles, *University of Hull*

With Windows Phone it is really easy to make publishing applications and games part of the learning experience. Students love being able to share their work with friends, family and even future employers. In this session you'll discover the wealth of Windows Phone based teaching resources available and how they can be used to give students a head start in creating useful applications (including use of Cloud) and entertaining gameplay for the Windows Phone platform, while they are at the same time learning software development techniques.

Thursday, 12:00 to 13:45

First Timer's Lunch Marriott State CDEF

Lunch Speaker: Jane Prey, Winner of 2012 SIGCSE Award for Lifetime Service

Lunch: On Your Own

Thursday, 13:45 to 15:00

PANEL A Stratified View of Programming Language Parallelism for Undergraduate CS Education 301AB

Chair: Richard Brown, *St. Olaf College*

Participants: Joel Adams, *Calvin College*; David Bunde, *Knox College*; Jens Mache, *Lewis & Clark College*; Elizabeth Shoop, *Macalester College*

It is no longer news that undergraduates in computer science need to learn more about parallelism. The range of options for parallel programming is truly staggering, involving hundreds of languages. How can a CS instructor make informed choices among all the options? This panel provides a guided introduction to parallelism in programming languages and their potential for undergraduate CS education, organized into four progressive categories: low-level libraries; higher-level libraries and features; programming languages that incorporate parallelism; and frameworks for productive parallel programming. The four panelists will present representative examples in their categories, then present viewpoints on how those categories relate to coursework, curriculum, and trends in parallelism.

SPECIAL SESSION Demystifying Computing with Magic 305B

Participants: Daniel Garcia, *UC Berkeley* and David Ginat, *Tel-Aviv University*

One man's "magic" is another man's engineering. – Robert A. Heinlein. Many novice students have fuzzy mental models of how the computer works, or worse, sincerely believe that the computer works unpredictably, "by magic". We seek to demystify computing by showing them that even magic itself isn't necessarily mystical; it could just be clever computation. In this session, we will share a variety of magic tricks whose answer is grounded in computer science: modulo arithmetic, permutations, algorithms, binary encoding, etc. For each trick, we will have an interactive discussion of its underlying computing fundamentals, and tips for successful showmanship. Audience participation will be critical, for helping us perform the magic, discussing the solution, and contributing other magic tricks.

PANEL Community-Based Projects for Computing Majors: Opportunities, Challenges and Best Practices 306C

Chair: Jeffrey Stone, *Pennsylvania State University*

Participants: Elinor Madigan, *Pennsylvania State University*; Janice Pearce, *Berea College*; Bonnie MacKellar, *St. John's University*

The use of community-based projects has been recognized as having pedagogical and experiential value for computing majors. Community-based projects can be valuable learning experiences for computing majors as well as for faculty and community partners. However, these types of projects present challenges for faculty and should be aligned with desired course outcomes. This panel will discuss the use of community-based projects from multiple perspectives. The expectation is that the panel will serve as a forum for participants to share the opportunities, challenges, pedagogical motivations, and best practices obtained from prior experience. Exemplar projects will be highlighted, and audience members will have an opportunity to share their own experiences with community-based projects.

PAPERS Games 302A

Chair: Adrienne Decker, *Rochester Institute of Technology*

13:45 The Five Year Evolution of a Game Programming Course

Gillian Smith and Anne Sullivan, *UC Santa Cruz*

This paper presents lessons learned from five years of teaching a game design and programming outreach course. This class is taught over the course of a month to high school students participating in the California Summer School for Mathematics and Science (COSMOS) at the University of California, Santa Cruz. Over these five years we have changed everything in the course, from the overall project structure to the programming language used in the class. In this paper we discuss our successes and failures, and offer recommendations to instructors offering similar courses.

14:10 Programming, PWNed: Using Digital Game Development to Enhance Learners' Competency and Self-Efficacy in a High School Computing Science Course

Katie Seaborn, *York University*; Magy Seif El-Nasr, *Northeastern University*; David Milam, *Simon Fraser University*; Darren Yung, *Frank Hurt Secondary School*

The popularity and inherent engagement of games has caused many educators to start thinking of ways to use game-based techniques to enhance education, particularly to promote STEM (Science Technology, Engineering and Mathematics) concepts to middle and high school students. We report on the design and evaluation of a high school game construction-based curriculum that replaced a traditional computer science class. We collected students' overall impressions, and evaluated students' technical competency and self-efficacy at the start and end of the semester. Our findings show that the curriculum had a positive, statistically significant effect on concept comprehension, which suggests that the curriculum was effective for understanding computer science and game design concepts.

14:35 A Learning Objective Focused Methodology for the Design and Evaluation of Game-based Tutors

Michael Eagle and Tiffany Barnes, *University of North Carolina at Charlotte*

We present a methodology for the design and evaluation of educational games with a focus on well defined learning objectives and empirical verification.Combining practices from educational design, intelligent tutoring systems, classical test-theory, and game design, this methodology guides researchers through the steps of the design process, including identification of specific learning objectives, translation of learning activities to game mechanics, and the empirical evaluation of the final product. This methodology is particularly useful for young researchers and educators are encouraged to promote this methodology for use in student research experiences or serious games courses.

PAPERS Professional Experiences 302B

Chair: Sarah Heckman, *North Carolina State University*

13:45 Course Guides: A Model for Bringing Professionals into the Classroom

Thomas Gibbons, *The College of St. Scholastica*

A new model, professional course guides, describes how practicing professionals can be brought into the classroom as student mentors and integrated into the course material. This new model is compared to existing models for student interactions with practicing professionals including guest speakers, adjunct faculty, and program mentors.

14:10 Towards a Better Capstone Experience

Sriram Mohan, Stephen Chenoweth and Shawn Bohner,
Rose-Hulman Institute of Technology

The capstone experience is designed to bridge the gap from university expectations to those of industry. Yet trying to solve this problem with a single course sequence, even one spanning the senior year, has some shortcomings, in terms of learning outcomes which can be achieved, and also instructional strategies that can be employed. We describe a plan which provides a junior year of practice on a client-based project integrated with learning design and other related topics, followed by a senior year in which students can work more independently to hone these skills on a harder year-long project with another client. This sequence, with scaffolding provided at first that is gradually removed, has proven to be especially effective in preparing undergraduates for a career in the software industry.

14:35 An Open Co-op Model for Global Enterprise Technology Education: Integrating the Internship and Course Work

Jeffrey Saltz, *JP Morgan Chase*; Jae Oh, *Syracuse University*

We present an open co-op program called Global Enterprise Technology Immersion Experience (GET IE). The program provides a global enterprise focus integrated with hands-on experiential work-based learning. GET IE includes a two-semester internship that can be seamlessly incorporated within an existing computer science curriculum. The internship's unique pedagogical innovation is to simultaneously provide the students academic course work that is integrated within a students extended internship and provides relevant problems in global enterprise technology. The curricula is "open" in the sense that other institutions and companies can join the consortium to enrich choices for the students and foster cross-fertilization of curricula activities.

PAPERS A Session with a View 306A

Chair: Don Goelman, *Villanova University*

13:45 Integrating Video Components in CS1

Tamar Vilner, Ela Zur and Ronit Sagi, *The Open University of Israel*

The Open University of Israel (OUI) is a distance learning university. Our CS1 course is taught through video-taped lectures that cover the study material. In addition, students may participate in face-to-face group meetings in study centers located all over the country and taught by tutors. There is a special group called Ofek, in which the tutor is located in a studio and the lesson is broadcast over the internet. Students enrolled in this group participate from their home PCs. The taped Ofek sessions as well as the lectures are stored on the course website, and students can watch them whenever convenient. We conducted a study to investigate students' viewing habits and the relationship between viewing and the success rate in the course.

14:10 Development and Evaluation of Indexed Captioned Searchable Videos for STEM Coursework

Tayfun Tuna, Jaspal Subhlok, Varun Varghese, Olin Johnson and Shishir Shah, *University of Houston*; Lecia Barker, *University of Texas-School of Information*

Videos of classroom lectures have proven to be a popular and versatile learning resource. This paper reports on ICS videos featuring Indexing, Captioning, and Search capability. The goal is to allow a user to rapidly zoom in on a topic of interest, a key shortcoming of the standard video format. A lecture is automatically divided into logical indexed video segments by analyzing video frames. Text is automatically identified with OCR technology and image transformations to drive keyword search. Captions can be added to videos. ICS video player integrates indexing, search, and captioning in video playback and is used by dozens of courses and 1000s of students. The paper reports on development and evaluation of ICS videos framework and assessment of its value as an academic learning resource.

14:35 Metaview: A Tool for Learning About Viewing in 3D

James Miller, *University of Kansas*

Metaview is an interactive tool that helps to teach concepts related to nested 3D coordinate systems, especially in the context of defining and establishing views of 3D scenes in common graphics APIs like OpenGL and Direct3D. We describe the context in which nested coordinate systems arise in the study of graphics programming, then we relate the common conceptual difficulties students typically experience when studying and trying to put this material into practice. We then describe the role that metaview plays in helping to overcome these problems. Metaview is packaged with a set of built-in 3D models used to demonstrate major concepts. In addition, external and/or student-programmed models are easily imported into the tool. Metaview can be run anywhere, anytime using Java Web Start.

PAPERS Pedagogy: Programming 306B

Chair: Saquib Razak, *Carnegie Mellon University*

13:45 Mediated Transfer: Alice 3 to Java

Wanda Dann, Dennis Cosgrove, Don Slater and Dave Culyba, *Carnegie Mellon University*; Steve Cooper, *Stanford University*

In this paper, we describe a pedagogy for an undergraduate programming course using Alice 3 and Java. We applied the educational theory of mediated transfer to develop a new version of the Alice system and accompanying instructional materials. The pedagogy was implemented and tested over two years. Student test scores in experimental, treatment course sections showed a dramatic increase in scores over comparable, non-treatment sections.

14:10 Over-Confidence and Confusion in Using Bloom for Programming Fundamentals Assessment

Richard Gluga, Judy Kay, Sabina Kleitman and Tim Lever, *University of Sydney*; Raymond Lister, *University of Technology Sydney*

A computer science student is required to progress from a novice to an expert through the CS1/2 programming fundamentals sequence. The key contribution is a web-based interactive tutorial that enables computer science educators to practice applying the Bloom Taxonomy in classifying programming exam questions, to define this learning progression. The results of an evaluation with ten participants were analyzed for consistency and accuracy in the application of Bloom. Confidence and self-explanation measures were used to identify problem areas in the application of Bloom to programming fundamentals. The tutorial and findings are valuable contributions to future ACM/IEEE CS curriculum revisions, which are expected to have a continued emphasis on Bloom.

14:35 Modeling How Students Learn to Program

Stephen Cooper, Chris Piech, Mehran Sahami, Daphne Koller and Paulo Blikstein, *Stanford University*

Despite the potential wealth of educational indicators expressed in a student's approach to homework assignments, how students arrive at their final solution is largely overlooked in university courses. In this paper we present a methodology which uses machine learning techniques to autonomously create a graphical model of how students in an introductory programming course progress through a homework assignment. We subsequently show that this model is predictive of which students will struggle with material presented later in the class.

Supporter Session Microsoft 305A

Creative Uses for Kinect in Teaching – with Curriculum Materials

Rob Miles, *University of Hull*

The Kinect sensor is the "Fastest Selling Consumer Electronics Gadget in History". It is a great way to add a new dimension to Xbox 360 gameplay, able to read its environment and track the body movement of players. It is also a great teaching tool and a genuinely creative device. In this session Rob Miles will show how you can harness this creativity and get students enjoying themselves while writing programs that make use of the unique abilities of this amazing sensor and its accompanying Kinect for Windows software. He will also have curriculum materials to share with you that you can use freely in your own classes.

Thursday, 13:45 to 17:15

Student Research Poster Session Exhibit Hall A

Thursday, 15:00 to 15:45

Break and Exhibits Exhibit Hall A

Thursday, 15:00 to 16:30

NSF Showcase #2 Exhibit Hall A

- **Engaging African Americans in Computing through the Collaborative Creation of Musical Remixes,** Brian Magerko and Jason Freeman
- **Building a K-12 Computing Pipeline in Alabama to Address Participation Diversity,** Jeff Gray, Mike Wyss, Shelia Cotten, and Shaundra Daily
- **Computational Thinking in IT - A Scenario-based Approach,** Deborah Boisvert, Paula Velluto, Irene Bruno, and Charles Winer
- **Process-Oriented Guided Inquiry Learning in Computer Science,** Clifton Kussmaul, R. Libby, and Carl Salter

Thursday, 15:45 to 17:00

PANEL — Science Fiction in Computer Science Education — 301AB

Chair: Rebecca Bates, *Minnesota State University, Mankato*

Participants: Judy Goldsmith, *University of Kentucky*; Rosalyn Berne, *University of Virginia*; Valerie Summet, *Emory University*; Nanette Veilleux, *Simmons College*

The use of science fiction to engage students in computer science learning is becoming more popular, with ample material available to help students make connections between technical content and human experience, from Star Trek to The Hitchhiker's Guide to the Galaxy to 2001 to I, Robot to … Fiction can be included in technical courses or used to draw students into the field in introductory classes. The panelists, who represent a range of schools, perspectives and classes, will present brief overviews of how they have used science fiction to engage students in technical topics as well as related ethical and societal issues. After the overviews, there will be plenty of time for discussion of examples and ways to make connections between science fiction and particular classes or topics.

PANEL — Diversity Initiatives to Support Systemic Change in Undergraduate Computing — 305B

Chair: Leisa D. Thompson, *University of Virginia*

Participants: Lecia Barker, *University of Texas*; Rita Manco Powell, *University of Pennsylvania*; Catherine Brawner, *Research Triangle Educational Consultants*; Tom McKlin, *The Findings Group, LLC*

The National Center for Women & Information Technology (NCWIT) Extension Services for Undergraduate Programs (ES-UP) has created a large group of trained consultants (ESCs) and clients who are passionate about women's participation in computing. This panel will describe how our ESCs and clients have worked together to effect change and will show outcomes from our activities over the past three years.

SPECIAL SESSION — Transforming the CS Classroom with Studio-Based Learning — 306C

Participants: Christopher Hundhausen, *Washington State University;* N. Hari Narayanan and Dean Hendrix, *Auburn University;* Martha Crosby *University of Hawaii at Manoa*

The studio-based learning (SBL) model has been the centerpiece of architecture and fine arts education for over a century. Over the past five years, we have been adapting SBL for computing education and empirically evaluating its impact. This effort now involves 26 computing courses at 15 institutions in seven states. To our knowledge, this is the largest implementation and evaluation of a pedagogy for computing education to date. This special session will introduce SBL to a general audience and facilitate a discussion and exchange of ideas. In addition to oral and poster presentations of the SBL model and its evaluation results, the session will feature "war stories" from teachers who have used SBL in their courses, and hands-on activities to help attendees apply SBL to their courses.

PAPERS Broadening Participation 302A

Chair: Kristine Nagel, *Georgia Gwinnett College*

15:45 Making Turing Machines Accessible to Blind Students

Pierluigi Crescenzi, Leonardo Rossi and Gianluca Apollaro,
University of Florence

In this paper we describe how we tried to make the JFLAP Turing machine simulator accessible to blind students. Software accessibility is an important topic for both legal and ethical reasons: in our case, however, we also wanted to make the accessible software usable by blind students in cooperation with the other students, in order to encourage the integration of the blind students within the rest of the class. For this reason, the accessible version of the JFLAP Turing machine simulator that we developed is as much similar as possible to and fully compatible with the original one. In the paper, we also report some very satisfactory preliminary validation results that indicate how the new software can really make Turing machines accessible to blind students.

16:10 Toward an Emergent Theory of Broadening Participation in Computer Science Education

David Webb, Alexander Repenning and Kyu Han Koh,
University of Colorado at Boulder

A fundamental challenge to computer science education is the difficulty of broadening participation of women and underserved communities. The idea of game design and game programming as an activity to introduce children at an early age to computational thinking in a motivational way is quickly gaining momentum. A pedagogical approach called Project First has allowed the Scalable Game Design project to reach over 4,000 middle schools students including a large percentage of female (45%) and underrepresented (48%) students. Our analysis of student motivation data, gender ratios and pedagogical approaches employed by teachers such as mediation and scaffolding suggests strong gender effects based on gender ratios and classroom scaffolding approaches.

16:35 Exploring Formal Learning Groups and their Impact on Recruitment of Women in Undergraduate CS

Julie Krause, Irene Polycarpou and Keith Hellman, *Colorado School of Mines*

As percentages of women in computing jobs and university programs decline, recruiting and retaining women in the field of Computer Science (CS) becomes increasingly important. Undergraduate CS programs, and more specifically, introductory-level CS courses, offer an opportunity to introduce women to CS studies. Furthermore, learning experiences in introductory CS courses can be pivotal in shaping female students' perceptions of CS. Collaborative learning, in various forms, is an instructional construct that has been shown to be effective in recruiting and retaining women in undergraduate CS programs. In this paper we present an exploratory study on formal learning groups and their potential to attract and maintain students' interest in CS studies.

PAPERS Online Collaboration 302B

Chair: Charles Leska, *Randolph-Macon College*

15:45 Perspectives on Active Learning and Collaboration: JavaWIDE in the Classroom

Jam Jenkins, *Valdosta State University*; Evelyn Brannock and Sonal Dekhane, *Georgia Gwinnett College*; Thomas Cooper, *The Walker School*; Mark Hall, *University of Wisconsin – Marathon County*; Michael Nguyen, *Emory University*

JavaWIDE is an innovative environment that promotes active learning and collaboration in programming courses. This paper discusses where and how JavaWIDE has been used to promote active and collaborative learning in both traditional and synchronous distance education courses in four different environments: high school, summer enrichment, and at two- and four-year colleges. After discussing the educational atmosphere and how active learning and collaboration are used in the courses, student responses to the experience are summarized. This collection of case studies illustrates how the concurrent editing, shared environment awareness and other features of JavaWIDE can be used to promote active learning and collaboration within a heterogeneous set of teaching and learning environments.

16:10 How Well Do Online Forums Facilitate Discussion and Collaboration Among Novice Animation Programmers?

Christopher Scaffidi, Aniket Dahotre and Yan Zhang, *Oregon State University*

Animation programming is a widely-respected approach for helping students to learn programming skills, and online forums are a widely-used approach for helping students to interact with one another. But in what ways, if any, does combining animation programming with online forums lead to useful discussion and collaboration among learners? To answer this question, we analyzed online forum discussions among people who were learning to create animation programs using the Scratch programming environment. We discovered that specific kinds of online posts were more likely than others to be followed by discussion, and we found that the ensuing collaboration often involved the exchange of design ideas and feedback within small groups of users.

16:35 Classroom Salon: A Tool for Social Collaboration

John Barr, *Ithaca College*; Ananda Gunawardena, *Carnegie Mellon University*

Classroom Salon is an on-line social collaboration tool that allows instructors to create, manage, and analyze social networks (called Salons) to enhance student learning. Students in a Salon can cooperatively create, comment on, and modify documents. Classroom Salon provides tools that allow the instructor to monitor the social networks and gauge both student participation and individual effectiveness. This paper describes Classroom Salon, provides some use cases that we have developed for introductory computer science classes and presents some preliminary observations of using this tool in several computer science courses at Carnegie Mellon University.

PAPERS Middle School Collaborations 306A

Chair: Catherine Lang, *Swinburne University of Technology*

15:45 Bringing The Breadth of Computer Science to Middle Schools

Elizabeth Carter and Glenn Blank, *Lehigh University*; Jennifer Walz, *Harrison Morton Middle School*

In order to garner more student interest in the pursuit of computer science as both a major and a career path, K-12 students need to be made aware of what computer science is and what it is about earlier in their education. Although students in many high schools can pursue introductory programming, high school is arguably too late to interest students who may have developed ill-informed attitudes about computer science. This paper documents curricular items developed for and taught to an audience of mixed ability 6th through 8th graders taking a local Technology Education class that attempts to showcase some of the more interesting, less stereotypical, aspects of computer science using a breadth approach in an effort to encourage interest in the field.

16:10 Integrating Hard and Soft Skills: Software Engineers Serving Middle School Teachers

Richard Burns, Lori Pollock and Terry Harvey, *University of Delaware*

We have developed and implemented, in four instances, a model for engaging computer science majors in integrating computing into teaching at a K-8 school in an underserved community. This paper describes the design of the service learning course particularly focused on interweaving software engineering practice, service learning, and development of soft skills. CS student teams partner with middle school teacher teams to create learning games, and conduct classroom instruction and observation. We report on our results from evaluating the impact of the course experience on the CS students and middle school teachers through pre-post surveys, evaluator observation of student demo presentations and classroom instruction, focus groups, and student reflective journals.

16:35 The Fairy Performance Assessment: Measuring Computational Thinking in Middle School

Linda Werner, *University of California, Santa Cruz*; Jill Denner and Shannon Campe, *ETR Associates*; Damon Chizuru Kawamoto, *Brown University*

Computational thinking (CT) has been described as an essential capacity to prepare students for computer science, as well as to be productive members of society. But efforts to engage K-12 students in CT are hampered by a lack of definition and assessment tools. In this paper, we describe the first results of a newly created performance assessment tool for measuring CT in middle school. We briefly describe the context for the performance assessment (game-programming courses), the aspects of CT that are measured, the results, and the factors that are associated with performance. We see the development of assessment tools as a critical step in efforts to bring CT to K-12, and to strengthen the use of game programming in middle school. We discuss problems and implications of our results.

PAPERS New Tricks for the Classroom 306B

Chair: Julian Mason, *Duke University*

15:45 Running Students' Software Tests Against Each Others' Code: New Life for an Old "Gimmick"

Stephen Edwards, Zalia Shams, Michael Cogswell and Robert Senkbeil, *Virginia Tech*

At SIGCSE'02, Goldwasser suggested including testing in assignments and then running every student's tests against every other student's program. This provides more insight into the quality of a student's tests as well as her solution. Software testing is more common now, but the all-pairs model of executing tests is still rare. This is because student-written tests, such as JUnit tests, take the form of program code and may depend on any aspect of their author's own solution, and these dependencies can keep tests from compiling against other programs. We discuss this problem and present a Java solution using bytecode rewriting and reflection. Results of applying this technique to two assignments involving 147 students and 240,158 individual test case runs demonstrates feasibility.

16:10 Group Note-Taking in a Large Lecture Class: Design, Implementation, and Evaluation of a Low-Cost Universal Design Practice

Christopher Plaue, Sal LaMarca and Shelby H. Funk, *University of Georgia*

We created a group note-taking system in our large intro computer science course to increase interaction amongst students, promote good note-taking strategies, and provide study resources to all students while also fulfilling the role of accommodating for students with learning disabilities. We show that the section of the course taught with our intervention performed significantly better on their final examination compared to a course taught without the intervention. We report that students enjoyed increased interactions with their peers, and that a third of the class self-reported an increase in their note-taking skills. Furthermore, our intervention only required minimal cost to the institution, and no financial cost to students, and is easily implemented in any size class.

16:35 Following a Thread: Knitting Patterns and Program Tracing

Michelle Craig, *University of Toronto*; Sarah Petersen; Andrew Petersen, *University of Toronto Mississauga*

This paper presents observations about teaching program tracing to novices drawn from a study of knitting patterns. Knitting patterns have evolved from vague, chatty discourse written for experts to precise, line-by-line procedures akin to programs. The knitting community has developed conventions for articulating iteration, conditions, and design decisions. "Executing" one of these patterns is analogous to tracing, so we argue that conventions adopted by knitters to make their patterns comprehensible to non-experts provide insights about teaching tracing to novices. Our observations suggest that using "until" instead of "while" and partially unrolling loops may help beginners understand code and that some structures, like parameters, may be unfamiliar.

SUPPORTER SESSION **Google** **305A**

All Things Google and Education

Margaret Johnson, *Director of University Relations and Education, Google Inc.*

Google believes that all students should have the opportunity to become active creators of tomorrow's technology. Through our diverse set of education efforts, we invest in the next generation of computer scientists and engineers, providing opportunities for all students to engage more directly in technology. Google's mission is to organize the world's information and make it universally accessible and useful. With regard to education, our goal is to leverage Google's strengths and infrastructure to increase access to high-quality, open educational content and technology.

During this session, you will learn about all of Google's education initiatives with a focus on those related to Computer Science.

Thursday, 17:10 to 18:00

Birds of a Feather

BOF **CS Unplugged, Outreach and CS Kinesthetic Activities** **201**

Lynn Lambert, *Christopher Newport University;* Tim Bell, *University of Canterbury;* Daniela Marghitu, *Auburn University*

Outreach activities including Computer Science Unplugged demonstrate computer science concepts at schools and public venues based around kinesthetic activities rather than hands-on computer use. Computer Science Unplugged is a global project with many such activities for children to adults using no technology, including how binary numbers represent words, images and sound, routing and deadlock, public/private key encryption, and others. Effective outreach programs such as this combats the idea that computer science = programming or, worse, keyboarding; and can educate the public, interest students, and recruit majors. Many people have used these activities, and adapted them for their own culture or outreach purposes. Come share your outreach ideas and experiences with such activities.

BOF Infusing Software Assurance and Secure Coding into Introductory CS courses 205

Elizabeth Hawthorne, *Union County College;* Carol Sledge and Nancy Mead, *Carnegie Mellon University;* Mark Ardis, *Stevens Institute of Technology*

Nearly every facet of modern society depends heavily on highly complex software systems. The business, energy, transportation, education, communication, government, and defense communities rely on software to function, and software is an intrinsic part of our personal lives. Software assurance is an important discipline to ensure that software systems and services function dependably and are secure. So, where are the resources to assist computer science educators with this instructional material? Session leaders will share materials from the Software Assurance Curriculum Project at the Software Engineering Institute of Carnegie Mellon University, and will facilitate discussion centered on infusing software assurance into introductory computer science courses at different types of colleges.

BOF Web-CAT User Group 206

Stephen Edwards, *Virginia Tech*

Web-CAT is the most widely used open-source automated grading system, with about 10,000 users at over 65 institutions worldwide. Its plug-in architecture supports extensibility, with plug-ins for Java (including Objectdraw, JTF, Swing, and Android), C++, Python, Haskell, and more. It is also a powerful tool for educational research data collection. It supports a wide variety of assessment strategies, but is famous for "grading students on how well they test their own code." Web-CAT won the 2006 Premier Award, recognizing high-quality, non-commercial courseware for engineering education. This BOF will allow existing users and new adopters to meet, share experiences, and talk about what works and what doesn't. Information on getting started quickly with Web-CAT will also be provided.

BOF Teaching Open Source 301AB

Sebastian Dziallas, *Franklin W. Olin College of Engineering;* Heidi Ellis, *Western New England University;* Mel Chua, *Purdue University;* Steven Huss-Lederman, *Beloit College* and Karl Wurst, *Worcester State University*

Involving students from a wide range of backgrounds in Free and Open Source Software project communities gets them a hands-on, portfolio-building experience in the creation of a real-world project while simultaneously building their institution's public profile. The Teaching Open Source (http://teachingopensource.org) community is an emergent (3 year old) group working on scaffolding to bridge the cultural differences between academic and FOSS communities of practice. Join us to share questions, challenges, and triumphs of incorporating FOSS participation into existing and new curricula as well support resources for doing so. Alumni and current members of the POSSE (Professors' Open Source Summer Experience http://communityleadershipteam.org/posse) will attend in mentorship roles.

BOF AP CS Principles and the 'Beauty and Joy of Computing' Curriculum 302A

Brian Harvey and Luke Segars, *University of California, Berkeley* Tiffany Barnes, *University of North Carolina, Charlotte;*

The College Board's guidelines for the coming AP CS Principles course are broad enough to allow many different interpretations. In particular, different courses have different levels of technical depth. The "Beauty and Joy of Computing" curriculum, used by two of the initial five pilot sites, aims high, with recursion and higher order functions included in the programming half of the course. This session is for high school or college level instructors considering teaching an AP CS Principles course and interested in using the BJC curriculum, and/or the Snap! (formerly BYOB) visual programming language used in the curriculum. See http://bjc.berkeley.edu for the curriculum and http://snap.berkeley.edu for the language.

BOF Teaching Track Faculty in CS 302B

Daniel Garcia, *UC Berkeley;* Jody Paul, *Metropolitan State College at Denver;* Mark S. Sherriff, *University of Virginia*

Many computer science departments have chosen to hire faculty to teach in a teaching-track position that parallels the standard tenure-track position, providing the possibility of promotion, longer-term contracts, and higher pay for excellence in teaching and service. This birds-of-a-feather is designed to gather educators who are currently in such a position to share their experiences as members of the faculty of their departments and schools, and to provide opportunities for schools considering such positions to gather information.

BOF A Town Meeting: SIGCSE Committee on Expanding the Women-in-Computing Community 302C

Gloria Townsend, *DePauw Universtiy*

In January 2004, we organized the second SIGCSE Committee ("Expanding the Women-in-Computing Community"). Our annual Town Meeting provides dissemination of information concerning successful gender issues projects, along with group discussion and brainstorming, in order to create committee goals for the coming year. We select projects to highlight through listserv communication and through our connections with NCWIT, ABI, ACM-W, CRA-W, etc. This year we will highlight the new NSF Broadening Participation in Computing grant – a grant that encompasses projects we presented in previous BOFs and a grant that builds on an alliance among ACM-W, ABI and NCWIT.

BOF Sharing Incremental Approaches for Adding Parallelism to CS Curricula 305A

Richard Brown, *St. Olaf College;* Elizabeth Shoop, *Macalester College;* Joel Adams, *Calvin College;* David Bunde, *Knox College;* Jens Mache, *Lewis & Clark College;* Paul Steinberg and Michael Wrinn, *Intel Corporation;* Matthew Wolf, *Georgia Tech*

Recent industry changes, including multi-core processors, cloud computing, and GPU programming, increase the need to teach parallelism to CS undergraduates. But few CS programs can afford to add new courses or greatly alter syllabi, and the large parallelism body of knowledge relates to many courses. Participants in this BOF will share incremental approaches for adding parallelism to undergraduate CS curricula, where students study parallel computing in brief units. This networking event/brainstorming session/swap meet will bring together: people with sharable parallelism expository readings, hands-on exercises, tech support ideas, etc.; people wishing to include such materials in their courses; and people curious about incremental approaches to teaching parallel computing.

BOF Computer Science: Small Department Initiative 305B

James Jerkofsky, *Walsh University;* Cathy Bareiss, *Olivet Nazarene University*

Faculty in small departments (perhaps 3 FTE, perhaps only 1 or 2,...) face special situations – both challenges and strengths. In this BOF, members will have a chance to talk about both. Challenges include maintaining a well-rounded curriculum and attracting students. Strengths include a close relationship with other members of the department and majors. These and other topics are open for discussion; the specific topics will be based upon the composition and interests of the group assembled.

BOF Teaching with Alice 306A

Donald Slater and Wanda Dann, *Carnegie Mellon University;* Steve Cooper, *Stanford University*

This session is for anyone currently using Alice 2.2 and / or thinking about using Alice 3, or exploring the possibility of using Alice in his or her curriculum. The discussion leaders and experienced Alice instructors will share teaching strategies, tips, and tricks with each other and those new to Alice. The session provides an arena for sharing Alice instructional materials and ideas for courses at all educational levels. This is an opportunity to share assignments and pointers to web sites where collections of instructional materials, such as syllabi, student projects, exams, and other resources are available.

BOF Identifying Effective Pedagogical Practices for Commenting Computer Source Code 306B

Peter DePasquale, *The College of New Jersey;* Michael Locasto, *University of Calgary;* Lisa Kaczmarczyk, *Independent Consultant*

Few, if any, pedagogical practices exist for helping students embrace best practices in writing software documentation, particularly source code comments. Although instructors often stress the importance of good commenting, two problems exist. First, it can be difficult to actually define these best practices, and second, it can be difficult to grade or assess students' application of such methods/practices. This BoF focuses on capturing for dissemination a concrete list of code commenting best practices used by the BoF attendees as they teach their classes.

BOF Design of a Computer Security Teaching and Research Laboratory 306C

Jeffrey Duffany, *Universidad del Turabo;* Alfredo Cruz, *Politechnic University of Puerto Rico*

To engage students and enhance the learning process a certain amout of hands-on experience is desirable to supplement the theory portion of computer security-related courses. This includes courses in information assurance, database security, network security, computer security, computer forensics among others. This BOF will include the opinion of professors that are actually delivering these courses to graduate and undergraduate students. They will tell us what kind of hardware and software is needed to develop a computer security lab or to enhance a classroom environment, with an emphasis on free and open source software, operating systems and the use of virtual machines to perform virus research.

BOF Student ICTD Research and Service Learning Abroad 307

Joseph Mertz, *Carnegie Mellon University;* Ralph Morelli, *Trinity College;* Ruth Anderson, *University of Washington*

This BOF is a chance for information sharing among faculty interested in involving students in ICTD research and/or service learning toward cultural and economic development globally. It takes a lot to get students out into the field. Challenges include developing partnerships, negotiating agreements, vetting the safety of destinations, identifying sources of funding, navigating the logistics of immunizations, visas, accommodations and flights to less-traveled places, reassuring parents as to the wisdom of their child's participation, managing development partner expectations, advising students' activities, and many more. This BOF will provide a venue for sharing experiences, information, and identifying potential new collaborations.

BOF Imaging College Educators Marriott University A

Jerod Weinman, *Grinnell College;* Ellen Walker, *Hiram College*

Within computing, the imaging field includes computer vision, image understanding, and image processing. While much research and teaching is done at the graduate level, the typical imaging educator at an undergraduate institution is the only specialist in his or her department. This BOF brings together educators who currently teach imaging courses or may be interested in expanding curricular offerings. We will emphasize sharing best practices, ideas, and resources as well as building a network for continued cooperation. Discussion topics may include course organization, assignments and projects, and lecture aids or other materials. Our network will include a mailing list for participants to ask questions and share ideas about imaging pedagogy and other means of sharing course materials.

BOF Let's Talk Social Media Marriott University B

Kimberly Voll, *University of British Columbia*

Our students have Facebook, G+, and even Twitter accounts as a matter of course, and are used to rich, highly integrated environments. In contrast, CS education is via themed modalities: lectures, textbooks, labs, discussions, et cetera, that share no active or social connection (you cannot +1 a lecture, for example, share a passage of a text with a classmate, or pull up a view that truly integrates a course and its community). But we now have the technology to create learning environments that share the same rich, multimedia experience as the popular social media sites. What should this look like? How do we start? What have you tried? We'll open with a brief overview of the leading social media tools for those unfamiliar, then proceed straight to an open discussion.

BOF Program by Design: TeachScheme/ReachJava Marriott University C

Viera Proulx, *Northeastern University;* Stephen Bloch, *Adelphi University*

Program by Design is a new name for the comprehensive introduction to programming at all levels that began with TeachScheme/ReachJava. This unconventional introductory computing curriculum covers both functional and the object-oriented program design in a systematic design-based style, enforcing test-first design from the beginning. The Bootstrap curriculum makes programming and algebra exciting for children ages 11-15. Special libraries support the design of interactive graphics-based games, musical explorations, client-server and mobile computing. We invite you to come and meet those who have used the curriculum, learn about new additions, libraries, bring in your experiences with the curriculum, show your projects, or ask questions about how it works and how you can use it.

BOF — CSTA Chapters: Supporting Your Local Computer Science Educators — Marriott Chancellor

Frances P. Trees, *Rutgers, The State University of New Jersey;* Helen Hu, *Westminster College;* Chinma Uche, *Greater Hartford Academy of Mathematics and Science*

As part of its commitment to developing a strong community of computer science educators, the Computer Science Teachers Association (CSTA) supports the development of regional CSTA chapters. A CSTA chapter is a local branch of CSTA designed to facilitate discussion of local issues, provision of member services at the local level, and to promote CSTA membership on the national level. This BOF will provide a platform for the discussion of CSTA chapter formation and for the sharing of successful chapter activities.

BOF — Revitalizing Computing Camp and Outreach: How Do We Engage Teenagers in “Cool” Technology? — Marriott Alumni

Kristine Nagel, *Georgia Gwinnett College;* Evelyn Brannock and Robert Lutz, *Georgia Gwinnett College*

Tech Camps are popular outreach tools to interest teens in computing programs and technology careers. One of the biggest obstacles is how to make Tech Camp “cool” and inviting for teenagers. How do we grab the attention of students to enroll? Once at camp, how do we engage teens with computing as a creative tool with relevancy to their lives? It is summer; subject areas must be entertaining and relevant. Can we stay ahead of the tech-savvy teens with our budget constraints? Robots and storytelling have long been used; how do we innovate and spark interest, throughout the year? The purpose of this BOF is to share ideas, such as App Inventor for Android to create apps, including text messaging, encouraging students to incorporate their own creative graphics, and using tablet devices.

Thursday, 18:10 to 19:00

BOF — Active eTextbooks for CS: What Should They Be? — 201

Cliff Shaffer, *Virginia Tech*

What should the textbook of tomorrow look like in a world of ubiquitous access to computing? Hypertextbooks have proved difficult to create and been fundamentally passive experiences. Commercial eBooks are merely books printed on an electronic screen instead of paper. New technologies such as HTML5 make it feasible to develop interactive applications that integrate with web services to provide a rich, pedagogically effective learning environment compatible with a range of computing platforms. We seek to generate discussion by participants to describe what they hope to see in online textbooks in the near future, and what resources and support would be required for them to adopt such a thing into their own courses.

BOF Enriching Computing Instruction with Studio-Based Learning 205

N. Hari Narayanan, *Auburn University;* Martha Crosby, *University of Hawaii at Manoa;* Dean Hendrix, *Auburn University* and Christopher Hundhausen, *Washington State University*

This BOF is related to the Special Session Transforming the CS Classroom with Studio-Based Learning (SBL). SBL promotes learning in a collaborative context by having students construct, present, review and refine their work with the guidance of peers and teachers. A team of CS educators and education experts have been implementing and evaluating SBL in CS courses over the past five years. The BOF will introduce SBL to the SIGCSE audience, and engage them in a discussion of the potential of, evidence for, and practical advice regarding SBL as an instructional approach that can motivate as well as teach students. Discussions will include “war stories” from teachers who have adopted the approach in their courses and hands-on activities to help participants apply SBL to their courses.

BOF AP CS A - Sharing Teaching Strategies and Curricular Ideas 206

Lester Wainwright, *Charlottesville High School;*Renee Ciezki, *Estrella Mountain Community College;* Robert Glen Martin, *TAG Magnet High School*

This BOF will provide an opportunity for high school and college faculty to discuss the AP CS A curriculum and to explore possibilities for collaborations and outreach activities between high schools and colleges.

BOF Regional Celebrations of Women in Computing (WiC) – Best Practices 301AB

Jodi Tims, *Baldwin-Wallace College;* Ellen Walker, *Hiram College;* Rachelle Kristof Hippler, *Bowling Green State University Firelands College*

Regional celebrations are locally organized, professional conferences modeled after the Grace Hopper Celebration of Women in Computing (GHC). This BOF allows people who have organized or would like to organize such a conference to get together to share successes and challenges. Attendees that have hosted a regional celebration are invited to bring a un-poster (i.e. 8.5 x 11 flyer, 30 copies) that highlights their conference features and/or shares lessons learned. The leaders plan to divide the time between the 5 major areas of conference planning: program, sponsorship, publicity/communications, registration, and site/logistics.

BOF Hacking and the Security Curriculum 302A

Richard Weiss, *The Evergreen State College;* Michael Locasto, *University of Calgary* and Jens Mache, *Lewis & Clark College*

Incorporating information security into the undergraduate curriculum continues to be a topic of interest to SIGCSE attendees. The purpose of this BOF is to help sustain the existing community of educators and researchers interested in bringing ethical hacking skills and an understanding of security into the classroom and relating these topics to the foundations of Computer Science. We would like to bring our colleagues together to share pedagogical practices, stories of hacking and how to use them to inspire our students and communicate complex concepts in computer science and security. We also plan to discuss our own experiences, practices and ongoing efforts (e.g., our infosec teaching experiences, the SISMAT program, EDURange and the dissemination of infosec interactive exercises).

BOF Flipping the Classroom 302B

Barry Brown, *Sierra College*

In a flipped classroom, students watch or listen to the lecture at home and do homework in the classroom. The classroom becomes much more interactive and the educator has ample opportunity to provide individualized guidance when it's most needed. The watch-at-home content can include recorded lectures, demonstration videos, adaptive quizzes, or anything in between. Come share your experiences developing "flip" material, learn from others what's involved, and find out whether it's working to improve success and retention.

BOF Using Social Networks to Engage Computer Science Students 302C

Semmy Purewal, *University of North Carolina at Asheville;* Owen Astrachan, *Duke University;* David Brown, *Pellissippi State Community College* and Jeffrey Forbes, *Duke University*

Social Networking continues to be a popular past-time among high school and college students. In this birds of a feather session, we will share ideas on integrating social networking topics into computer science courses at the introductory and non-major levels. Additionally we will discuss approaches to integrating social network programming into upper level courses. Finally we will attempt to address the following questions: will social networking draw new students into the computing disciplines the way that video games did in the previous generation? Will it attract new types of students with different expectations? Is social networking just a fad that will have no effect on Computer Science programs? Or is social networking a topic that is better left to other academic disciplines?

BOF Digital Humanities: Reaching Out to the Other Culture 305A

Robert Beck, *Villanova University*

This discussion will connect instructors who are reaching out to their colleagues in the humanities to discover areas of collaboration. It focuses on what these disciplines have to contribute to our knowledge of computing and how computational thinking informs these disciplines. One goal is to lay the foundation for a more general program of study in digital humanities that would reach students who would like to see how computing could enhance their work in history, literature, anthropology, or philosophy, for example.

BOF A Multimedia and Liberal Arts Approach to a First Course in Programming and its Crossover Potential for Computer Science and the Arts 305B

Trish Cornez, *University of Redlands;* Richard Cornez, *University of Redlands*

Students are acculturated in a visual, interactive, and interdisciplinary world. This BOF will provide a platform for a discussion on how multimedia can be integrated in a CS1 course. Discussions will focus on attributes of conventional and unconventional first languages and explore a liberal arts approach to integrate disciplines both scientific and artistic. We envision discussions relevant to: Mathematicians visualizing processes using multimedia and algorithms; Physicists using game programming to deconstruct and explore physical environments and re-assembling them as virtual worlds; Computer scientists and behavioral scientists collaborating on responsive systems to explore philosophical underpinnings of media; Musicians and computer scientists creating computational art.

BOF Teaching with App Inventor for Android 306A

Jeff Gray, *University of Alabama;* Harold Abelson, *MIT;* Ralph Morelli, *Trinity College;* Jeff Gray, *University of Alabama* and Chinma Uche, *Greater Hartford Academy of Math and Science*

App Inventor for Android is a visual blocks language for building mobile apps. Like Scratch, the language's drag-and-drop blocks interface significantly lowers the barrier to entry. Beginners can immediately build apps that interface with mobile technology (e.g., GPS, Text-to-speech, SMS Texting) and build apps that have a real-world impact. In this BoF, hosted by App Inventor creator Hal Abelson and experienced teachers and authors, we'll discuss the language, its future in K-12 and university education, and its new home at the MIT Center for Mobile Learning.

BOF Technology that Educators of Computing Hail (TECH): Come, Share your Favorites! 306B

Daniel Garcia and Luke Segars, *UC Berkeley*

The pace of technology for use in computing education is staggering. In the last five years, the following tools/websites have completely transformed our teaching: Piazza, Google Docs, YouTube, Doodle and whenisgood.net, Skype and Google Hangout, and Khan Academy among others. Hardware has also played a part – we love our Zoom H2 digital voice recorder (for recording CD-quality lecture audio), Blue Yeti USB mike (for audio/videoconferences), and iClickers (for engaging students in class). Do you wish you could easily share your favorites? Want to find out what the others know that you don't? Have a tool you've built and want to get some users? Come to this BOF! We'll also show the TECH website we've built that attempts to collect all of these tools in one place.

BOF Motivating CS1/2 Students with the Android Platform 306C

John Lewis, *Virginia Tech;* Anthony Allevato and Stephen H. Edwards, *Virginia Tech*

The use of Android in computing courses is growing. Students find it engaging because they can develop Java apps for mobile devices. Android also offers challenges in the classroom, especially in CS1 and CS2. As a professional- level platform, it uses design idioms that may require students to learn advanced language features earlier. It also adds logistical complications to setting up projects and development tools. Existing approaches to software testing and automated grading need adaptation. This BOF is for sharing assignments, resources, techniques, and experiences with others, focusing on issues that arise when balancing the teaching of fundamental concepts with the complexities required to accomplish basic tasks on the Android platform.

BOF Interdisciplinary Database Collaborations 307

Suzanne Dietrich, *Arizona State University;* Don Goelman, *Villanova University*

Databases play a major role across many disciplines for the storage and retrieval of information. Many database educators are establishing collaborations with colleagues representing a diverse spectrum of interests, for both research and pedagogical purposes. Further, the range of cooperating disciplines is expanding, as evidenced by the emergence of new fields such as computational journalism, as well as by the proliferation of discipline-specific dialects of XML. The goal of this Birds-of-a-Feather session is to bring database educators together to share their experiences on interdisciplinary collaborations in an open dialogue that is fostered by this format.

BOF Google Summer of Code and Google Code-in BoF — Marriott University A

Carol Smith, *Google, Inc.*

Google Summer of Code is the outreach program aimed at getting university students involved in a 3-month online internship working in open source software development.
Google Code-in is the contest aimed at involving 13-18 year olds in open source software development, documentation translation, outreach, research, and more. I will be discussing both programs at this BoF and encouraging students and teachers to get involved.
We'll open the forum for discussion amongst the attendees about how to participate, how to get the word out, and answer any questions they may have.

BOF Building Partnerships Across the CS Education Spectrum — Marriott University B

Chris Stephenson, *Computer Science Teachers Association;* Steve Cooper, *Stanford University;* Don Yanek, *Northside College Prep High School* and Jeff Gray, *University of Alabama*

Over the last five years, CSTA has built a solid outreach and teacher support network through the work of its chapters and Leadership Cohort. This network has also become a major source of active partnerships between K-12 teachers, their schools, and colleagues from colleges, universities, and industry. The goal of this BOF is to provide concrete examples and suggestions for SIGCSE members interested in building these kinds of partnerships.

BOF Engaging The Community With Mobile App Projects — Marriott University C

William Turkett and Paul Pauca, *Wake Forest University* and Joel Hollingsworth, *Elon University*

As the popularity of mobile devices surges, more and more organizations are looking to exploit the novel interaction methods of mobile devices to re-deploy legacy software or to develop innovative new applications. Many organizations are looking to nearby universities for expertise in this area. At the same time, mobile computing has become increasingly integrated within courses in CS departments. Historically, capstone courses and other advanced electives have resulted in the production of non-trivial software artifacts. This BOF will provide a platform for discussion of how the use of mobile app platforms in such courses can allow for the development of meaningful software projects that engage with and give back to the community and provide rich opportunities for service learning.

BOF — Have Class, Will Travel — Marriott Chancellor

Paige Meeker, *Presbyterian College*

At many schools, various disciplines offer travel courses (to other lands or to locations within the USA) to give students an experiential component to their learning. How can we introduce such courses to computer science departments? This BOF will provide a time of sharing ideas for such courses and welcomes discussion of travel courses that have been successfully taught. In addition to normal course preparation, these courses also involve travel arrangements, payment schedules, and careful scheduling to provide maximum benefit to the student. Our group will share ideas for locations of travel, topics of courses, and collaboration with other disciplines, as well as the additional overhead such a course entails, such as cost/payment schedule, insurance, itinerary, safety, etc.

BOF — Integration of Experiential Learning and Teaching: Beyond the Walls of the Classroom, Techniques, Challenges and Merits. — Marriott Alumni

Arshia Khan, Tamara Lichtenberg, Rishika Dhody, Joel Pouale, *The College of St. Scholastica* and John Woosley, *Southeastern Louisiana University*

Integration of experiential learning is critical in the field of computer science. With technology evolving over night, job requirements are extremely volatile. Educators have a challenging task of staying abreast with the technology and market needs while self learning the new technologies. One solution is to rely on the businesses for input on what should be taught and using them to extend the learning into the real world through experiential learning(not just internships). Talking points: Filling gaps between academia and industry; Faculty can share their methods of experiential learning; Applying the practical skills to theoretical knowledge – turning theory into practice; Opportunities to bring real world clients in the classroom

Thursday, 19:00 to 20:00

SIGCSE Reception — Registration Foyer

Friday, 7:15 to 8:15

Alice Breakfast 302B

Friday, 8:30 to 10:00

SIGCSE Keynote Address Ballroom BC

From Computational Thinking to Computational Values

Hal Abelson, *Massachusetts Institute of Technology*

SIGCSE members love the beauty of computational thinking. They know the joy of bringing those ideas to young people. That love for computational thinking entails respect for the computational values that empower people in the digital world. For academics, those values have been central to the flowering of computing as an intellectual endeavor.

Today, those values are increasingly threatened by stresses from both within and outside academia: squabbles over who owns academic work, increasingly stringent and overreaching intellectual property laws, and the replacement of open computing platforms by closed applications and walled-garden application markets.

In this talk I'll describe some things we've done at MIT to support computational values, like open publication of all our course materials, our faculty policy on open publication of academic research, and our recently announced initiative for open online instruction based on non-proprietary software platforms. I'll discuss Creative Commons licensing and Free Software, and the importance of tinkerability for empowering citizens in an information society. And I'll describe App Inventor for Android, a new programming tool motivated by the vision that all of us us can experience mobile computing as creators using tools that we can control and reshape, rather than only as consumers of packaged applications.

Friday, 10:00 to 10:45

Break and Exhibits Exhibit Hall A

Friday, 10:00 to 11:30

NSF Showcase #3 Exhibit Hall A

- **Databases for Many Majors: A Student-Centered Approach,** Suzanne W. Dietrich and Don Goelman
- **Establishing the Exploring Computer Science Course in Silicon Valley High Schools,** Dan Lewis, Ruth Davis, Pedro Hernandez-Ramos, and Craig Blackburn
- **Integrating Security in the Computing Curriculum: Security Injections Towson,** Siddharth Kaza and Blair Taylor
- **How to Run a Successful REU Site Program,** Susan D. Urban, Michael Shin, and Mohan Sridharan

Friday, 10:00 to 12:00

POSTER Poster Session I Exhibit Hall A

Using Reflection to Enhance Feedback for Automated Grading
Carl Alphonce, *University at Buffalo*; Joseph LeGasse, *Meritain Health*

The Reflective Mentor: Charting Undergraduates' Responses to Computer Science Service Learning
Quinn Burke, Yasmin Kafai, Jean Griffin, Rita Powell, Michele Grab, Susan Davidson and Joseph Sun, *University of Pennsylvania*

Merging Healthcare and Technology: A Multi-disciplinary Health Information Technology (HIT) Curriculum
Elizabeth Howard, Donna Evans and Marilyn Anderson, *Miami University - Middletwon*; Jill Courte, *Miami University - Hamilton*

An Integrated Introduction to Network Protocols and Cryptography to High School Students
William Mongan, *Drexel University*

A PC Robot for Learning Computer Vision and Advanced Programming
Xuzhou Chen and Nadimpalli V.R. Mahadev, *Fitchburg State University*

Girls Gather for Computer Science (G2CS)
Shereen Khoja, Juliet Brosing, Camille Wainwright and Jeffrey Barlow, *Pacific University*

Debuggems to Assess Student Learning in E-Textiles
Deborah Fields, Kristin Searle, Yasmin Kafai and Hannah Min, *University of Pennsylvania*

Mediascripting – Teaching Introductory CS by Through Interactive Graphics Scripting
Samuel Rebelsky, Janet Davis and Jerod Weinman, *Grinnell College*

Do Faculty Recognize the Difference Between Computer Science and Information Technology? A Survey of Liberal Arts Faculty
Jaime Spacco and Hannah Fidoten, *Knox College*

Interdisciplinary Travel Courses in Computer Science
Paige Meeker, *Presbyterian College*

User Type Clustering to Refine Search and Browse for Educational Resources
Monika Akbar and Clifford A. Shaffer, *Virginia Tech*

A Comprehensive CS Curriculum Revision, Implementation and Analysis
Steven Huss-Lederman, *Beloit College*

Developing an Interdisciplinary Health Informatics Security and Privacy Program
Xiaohong Yuan, Jinsheng Xu, Hong Wang and Kossi Edoh, *North Carolina A&T State University*

A Team Software Development Course Featuring iPad Programming
Robert England, *Transylvania University*

The Role of Belonging in Computer Science Student Engagement
Nanette Veilleux, *Simmons College*; Rebecca Bates, *Minnesota State University, Mankato*; Cheryl Allendoerfer, Diane Jones and Joy Crawford, *University of Washington*

Streamlining Project Setup in Eclipse for Both Time-Constrained and Large-Scale Assignments
Ellen Boyd and Anthony Allevato, *Virginia Tech*

A Customizable Platform for Classroom Collaboration Using Mobile Devices
Stephen Hughes, Ben Schafer, Aaron Mangel and Sean Fredericksen, *University of Northern Iowa*

Explaining the Dynamic Structure and Behavior of Java Programs using a Visual Debugger
Demian Lessa and Bharat Jayaraman, *SUNY at Buffalo*

Using FPGA Systems Across the Computer Science Curriculum
D. Brian Larkins, H. Erin Rickard and William M. Jones, *Coastal Carolina University*

Maximizing Content Learning for Deaf Students and English as a Second Language Students
Raja Kushalnagar and Joseph Stanislow, *Rochester Institute of Technology*

All-In-One Virtualized Laboratory
Shamsi Moussavi and Giuseppe Sena, *MassBay Community College*

Recursive Thinkers and Doers in CS1
Suzanne Menzel and Joseph Cottam, *Indiana University*

Computing in Context: Video Scenarios for Recognizing and Utilizing Basic Computing Constructs
Madalene Spezialetti, *Trinity College*

Programming Board-game Strategies in the Introductory CS Sequence
Ivona Bezakova, James Heliotis, Sean Strout, Adam Oest and Paul Solt, *Rochester Institute of Technology*

PANEL Teaching Mathematical Reasoning Across the Curriculum 301AB

Chair: Joan Krone, *Denison University*

Participants: Doug Baldwin, *SUNY Geneseo*; Jeff Carver, *University of Alabama*; Joseph Hollingsworth, *Indiana University Southeast*; Amruth Kumar, *Ramapo College of New Jersey;* Murali Sitaraman, *Clemson University*

We discuss ways in which the panel members have incorporated mathematical reasoning into a variety of courses, encouraging and supporting students to apply and enhance their reasoning skills in productive ways across the entire CS curriculum.

SPECIAL SESSION Teaching HS Computer Science as if the Rest of the World Existed 305B

Chair: Scott Portnoff, *Downtown Magnets High School, Los Angeles*

This session discusses the design, implementation and rationale for a HS pre-APCS curriculum of Interdisciplinary Central-Problem-Based (ICPB) units that model real-world applications. In a typical multi-week unit, students begin by solving a problem using a complex software application, such as SDSC Biology Workbench. Students then build a small-scale version of the program, focusing on 1 or 2 algorithms, using Processing, Excel, BYOB or Alice. This approach affords students both context and practical potential for their work. Unit topics come from the fields of Astronomy (Galileo), Bioinformatics (Evolution), Molecular Modeling (DNA Double Helix), Political Science (Women's Suffrage/ Electoral Process), Environmental Science, Music, and Holocaust Studies (Hollerith Machine Technology).

SPECIAL SESSION Funding the Challenges in Computing 306C

Participants: Guy-Alain Amoussou, *NSF* and Scott Grissom, *Grand Valley State University*

What are the perceived challenges by the computing education and research communities? After small groups discuss this question, we will compare and contrast these perceived challenges to the current trend in proposals submitted and funded by the National Science Foundation's (NSF) Transforming Undergraduate Education in STEM (TUES) program. The intention is to create awareness for all involved and to broaden the range of proposals submitted to NSF.

PAPERS CS1: New Ideas 302A

Chair: Lori Carter, *Point Loma Nazarene University*

10:45 Creative Coding and Visual Portfolios for CS1

Dianna Xu and Deepak Kumar, *Bryn Mawr College*; Ira Greenberg, *Southern Methodist University*

In this paper, we present the design and development of a new approach to teaching the college-level introductory computing course (CS1) using the context of art and creative coding. Over the course of a semester, students create a portfolio of aesthetic visual designs that employ basic computing structures typically taught in traditional CS1 courses using the Processing programming language. The goal of this approach is to bring the excitement, creativity, and innovation fostered by the context of creative coding. We also present results from a comparative study involving two offerings of the new course at two different institutions. Additionally, we compare our results with another successful approach that uses personal robots to teach CS1.

11:10 Stepping Up to Integrative Questions on CS1 Exams

Daniel Zingaro and Michelle Craig, *University of Toronto*; Andrew Petersen, *University of Toronto Mississauga*

We explore the use of sequences of small code writing questions (“concept questions”) designed to incrementally evaluate single programming concepts. We report on student performance on a CS1 final examination that included a traditional code-writing question and four corresponding concept questions. We find that the concept questions are significant predictors of performance on both the corresponding code-writing question and the final exam as a whole. We argue that concept questions provide more accurate formative feedback and simplify marking by reducing the number of variants that must be considered. An analysis of the student responses categorized by the students’ previous programming experience suggests inexperienced students have the most to gain from the use of concept questions.

11:35 Using Reflective Blogs for Pedagogical Feedback in CS1

Jeffrey Stone, *Pennsylvania State University*

The use of weekly, reflective student blogs can be one method for collecting ongoing feedback about a CS1 course. Reflective blogs permit a continuous feedback loop that can be used for both formative and summative assessment of pedagogical innovations. This paper reports on a two-year qualitative study involving the use of reflective blogging in six sections of two CS1 courses. Reflective blogs were used as a low stakes, non-intimidating vehicle whereby concerns, requests, and other course-related issues could be voiced by students. The posts were used as an assessment and feedback mechanism for pedagogical transformation of the participating courses. This study demonstrates that reflective blogs in CS1 can be a useful tool for faculty course development.

PAPERS Team Work 302B

Chair: Jody Paul, *Metropolitan State College of Denver*

10:45 Participation Patterns in Student Teams

Vreda Pieterse, Lisa Thompson and Linda Marshall, *University of Pretoria*; Dina Venter, *Olrac-SPS*

We describe a process for teaching teamwork in a Software Engineering module. Our objective was to create opportunities for our students to experience some problems of working in a group before they formed teams in which they had to work for the rest of the year. The process entails expecting students to work on well defined assignments for short periods in teams where risk factors were induced. Through experiencing these short bursts of team tribulation students are prepared to handle difficult events and situations in their teams. We describe the design and implementation of this process. We report on changes in the levels of participation of the students during the process. We offer our explanation of possible factors that may have caused the observed variations.

11:10 Application of Non-programming Focused Treisman-style Workshops in Introductory Computer Science

Lindsay Jamieson, Alan Jamieson and Angela Johnson, *St. Mary's College of Maryland*

In the 60s and 70s, Uri Treisman developed a specific style of workshops to encourage the retention of underrepresented minority students in Calculus courses. Since that time, workshops based on the Treisman model have been successful across the US and have resulted in more underrepresented minority students successfully completing Calculus. Some attempts have been made to translate the Treisman model to CS1, but all previous attempts have been focused on programming skills. However, one of the student assumptions that deter underrepresented minorities from attempting a major or minor in CS is that a computer scientist is a solitary programmer. In this paper, we discuss a specific two year pilot program of non-programming focused Treisman-style workshops in conjunction with a CS1 course.

11:35 Collaboration Across the Curriculum: A Disciplined Approach to Developing Team Skills

Matthew Lang and Ben Coleman, *Moravian College*

Increasing the communication and collaborative skills of computer science students has been a priority in the community for some time. We present our philosophy, collaboration across the curriculum, which moves beyond existing individual courses or course units to teach collaboration skills in a pervasive manner. In our approach, concepts are introduced and skills are developed throughout the computer science curriculum—from CS1 to a capstone experience. Students are provided with opportunities to exercise skills in reflective environments that eventually mirror real-world experiences, and technical course content is not compromised. We argue for this system and provide details about how collaboration across the curriculum is accomplished at a small liberal arts college.

PAPERS Summer Experiences 306A

Chair: Kinnis Gosha, *Morehouse College*

10:45 App Inventor for Android: Report from a Summer Camp

Krishnendu Roy, *Valdosta State University*

Google's App Inventor for Android (AIA) is the newest visual programming language designed to introduce students to programming through creation of mobile apps. AIA opens up the world of mobile apps to novice programmers. Success stories of using AIA to introduce college students to programming exist. We used AIA in computing summer camps for high school students that we offer at our university. This paper is an experience report about using AIA in our camps. We provide a detailed description of designing our camps with AIA including the process of selecting and setting-up an Android device and instructional materials that we designed and made available to everyone. We also share evaluation results of using AIA in our camps and our impression of AIA as a programming-introduction tool.

11:10 Sustainable and Effective Computing Summer Camps

Barbara Ericson, *Georgia Institute of Technology*; Tom McKlin, *The Findings Group*

Summer camps are a popular form of outreach for colleges and universities. But, it isn't enough to offer computing summer camps and hope students like them. The camps should be effective by some measure, such as broadening participation by underrepresented groups and/or increasing learning. Summer camps should also be sustainable, so that institutions can continue to offer them regularly. The summer camps at Georgia Tech have evolved to the point where they are sustainable and effective. This paper presents the rationale for our camps, the business model that makes them sustainable, and the evaluation results that demonstrate positive attitude changes and increases in learning.

11:35 A Summer Science Experience with Computer Graphics for Secondary Students

Timothy Davis, *Clemson University*

This paper describes the principles, implementation, and results of a weeklong summer science course for junior high and high school students interested in computer science. To motivate and foster interest and creativity in students, while providing adequate complexity to introduce programming concepts and techniques, we used programming projects in computer graphics as the main learning tool. Included in our discussion are experiences across three course offerings, as well as detailed course assignments.

PAPERS Software Engineering 306B

Chair: Ariel Ortiz, *Tecnológico de Monterrey, Campus Estado de México*

10:45 Integrating UX with Scrum in an Undergraduate Software Development Project

Janet Davis, Chase Felker and Radka Slamova, *Grinnell College*

We report our experiences using the Scrum agile software development method in an undergraduate user-centered web development project. Our chief contributions are to report on using Scrum in a summer research setting as distinct from academic-year coursework and to consider the integration of Scrum and user experience (UX) development methods in a non-professional, learning environment. Our experience with combining Scrum and UX was positive: this methodology gave our project a clear structure, kept us motivated, and focused us on developing a usable final product. We discuss our adaptations of Scrum to UX development and to the summer research setting, along with challenges we faced and lessons learned, to inform students and faculty who wish to apply such methods in future projects.

11:10 Using WReSTT in SE Courses: An Empirical Study

Peter J. Clarke, Jairo Pava, Debra Davis and Frank Hernandez, *Florida International University*; Tariq M. King, *North Dakota State University*

There continues to be a lack of adequate training for students in software testing techniques and tools at most academic institutions. Several educators and researchers have investigated innovative approaches to integrate testing into programming and software engineering (SE) courses with some success. The main problem is getting other educators to adopt their approaches and getting students to continue to use the techniques. In this paper we present a study that evaluates a non-intrusive approach to integrating software testing techniques and tools in SE courses using a Web-Based Repository of Software Testing Tools (WReSTT). The results of the study show that students who use WReSTT in the classroom can improve their understanding and use of testing techniques and tools.

11:35 Understanding the Tenets of Agile Software Engineering: Lecturing, Exploration and Critical Thinking

Shvetha Soundararajan, Amine Chigani and Arthur James, *Virginia Tech*

In our quest to develop better software products, it is imperative that we strive to learn and understand the application of agile methods to the software development enterprise. Unfortunately, students have only limited exposure to the agile philosophy, principles and practices at the graduate and undergraduate levels of education. In an effort to address this concern, we offered an advanced graduate-level course entitled “Agile Software Engineering” in the Department of Computer Science at Virginia Tech. The primary objectives of the course were to introduce the values and principles and practices underlying the agile philosophy, and to do so in an atmosphere that encourages debate and critical thinking. This paper describes our experiences during the offering of that course.

Supporter Session — Intel — 305A

Intel Academic Assembly: What are the next imperatives?

Lakshmi Talluru: *Director, Technology Strategies*
Burke Walls: *US Intern Program Manager*
Michael Wrinn: *Academic Outreach*
Bob Chesebrough: *Senior Course Architect*
Paul Steinberg: *Academic Community Manager*

There are a number of important currents in computing that will need to be addressed by undergraduate computer science. These include mobility, power and performance, security and cloud computing. How will these trends and concepts be embraced by the curriculum, supported by your administration and comprehended by your students? How can Intel help in this?

You are cordially invited to the Intel Academic Assembly to hear our view and offer your opinions and ideas. This will be your chance to meet members of the Intel Academic Team, learn about opportunities for engagement and propose new directions for collaboration.

For the last five years, Intel has been at the forefront of ongoing efforts to integrate parallelism into undergraduate computer science education. We are now starting to see the fruits of these efforts. Here in the United States, a new ACM Curriculum recommendation is coming out with parallelism throughout; Research work from Berkeley, the University of Illinois and other institutions on design patterns and parallel models is now being widely disseminated; worldwide, many universities and colleges have parallelism in their core curriculum. We look forward to continuing our work collaborating on these and new areas of endeavor.

Friday, 12:00 to 13:45

Lunch Break — On your own

Friday, 12:10 to 13:35

Snap! Lunch — 301AB

UPE National Meeting — 302AB

Keynote Speaker and Recipient of the 2012 UPE: Abacus Award
Alan Kay; *President, Viewpoints Research Institute*

Friday, 13:45 to 15:00

SPECIAL SESSION CS Principles: Piloting a National Course 301AB

Participants: Owen Astrachan, *Duke University*; Ralph Morelli, *Trinity College*; Dwight Barnette, *Virginia Tech*; Jeff Gray, *University of Alabama*; Chinma Uche, *Hartford Academy of Math and Science;* Bill Cowles *Booker T. Washington HS* and Rebecca Dovi *Patrick Henry HS*

The CS Principles course has been designed to be taught nationwide at both the secondary and post-secondary levels. As part of this joint NSF/College Board project, 11 high schools are partnered with 10 colleges to teach the course, be part of a national initiative to test assessment items, and to help validate the curriculum framework that is the basis for the course and project. This special session is a report of the second stage of the pilot that is designed to lead to a national standard and a new, additional AP exam in the next five years. This course will not replace the traditional, CS1-oriented AP exam, but will be a new national introduction to Computer Science.

SPECIAL SESSION Fun, Phone, and the Future - Microsoft XNA Game Studio, Windows Phone, and Kinect SDK 305B

Chair: Pat Yongpradit, *Springbrook High School*

Microsoft XNA Game Studio and C# provide the basis of an advanced high school or introductory post-high school game development computer science course. Game development is serious computer science. The curriculum tools enable students to create games, simulations, and applications for the PC, Xbox 360, Windows Phone, and Kinect that expands students' skills in complex logic, object oriented programming (OOP), advanced algorithms, and data structures. See and participate in demonstrations of student projects from the new Game Development with XNA course curriculum.

SPECIAL SESSION Building an Open, Large-Scale Research Repository of Initial Programming Student Behavior 306C

Participants: Michael Kölling and Ian Utting, *University of Kent*

Many initiatives in improving learning of programming are based on gut instinct or localised experience. Gathering data as a basis for interventions, especially on a large scale, is hard. The BlueJ environment is being instrumented to collect data useful to a variety of educational programming researchers. BlueJ is ideally placed to collect such data: Users number in the millions, situated all over the world. This volume and diversity is unique in the history of such investigations and presents a significant opportunity for researchers. The data will be open to interested research groups, which will enable a wide variety of investigations that were previously impractical. This session presents work to date and solicits input from researchers about the design of the data collection.

PAPERS Collaborative Learning 302A

Chair: Adrian German, *Indiana University School of Informatics and Computing*

13:45 Assigning Student Programming Pairs Based on their Mental Model Consistency: An Initial Investigation

Alex Radermacher, Gursimran Walia and Richard Rummelt, *North Dakota State University*

Pair Programming has been shown to be beneficial to student learning. This paper reports results of research investigating the effectiveness of pairing students based on their mental models. Prior research has found a correlation between mental model consistency and performance in computer programming courses. Students' mental models helps to provide insights into how students approach problem solving and may indicate how to effectively pair students to improve their programming ability and learning. Results indicate that mental model consistency is a predictor of student success in an introductory programming course. Future goals of this research are to fully evaluate all pairing arrangements and to produce tests to evaluate mental model consistency for other computer science concepts.

14:10 Group Whiteboards and Modeler/Customer Teams: Getting Closer to Industrial-Style Collaboration in a Classroom

David Socha, *University of Washington Bothell*

This paper reports on two simple innovations that help create a more authentic and engaging modeling experience in an undergraduate analysis and design course: (a) having each team of students act both as modelers for another team, and as customers for another team, and (b) providing each team with their own whiteboard. The results from their use throughout the course, and for a single use of the whiteboards in a Computing Technology and Public Policy course, were quite positive. They resulted in a qualitatively different experience noticeable both to the instructors and the students. While some students were initially reluctant to use the whiteboards, by the end of the course most students were enthusiastic about their use.

14:35 Is There Service in Computing Service Learning?

Randy Connolly, *Mount Royal University*

While service learning projects in post-secondary computing can achieve important disciplinary outcomes for the students, the benefit of these projects for the service recipients and their community has been under-examined. This paper argues that since these projects are meant to benefit both student donors and community recipients, we must examine more carefully how computing service projects interact with all the social actors affected by the projects. Taking such an approach will require recognizing that ICT by itself will not increase democracy, equality, or any other social good; indeed some service learning projects may actually do more harm than good. The paper concludes with some sample computer learning projects that are oriented towards achieving true service for the recipients.

PAPERS Curriculum Issues 302B

Chair: Colleen Lewis, *University of California, Berkeley*

13:45 Computer Science in NZ High Schools: The first Year of the New Standards

Tim Bell, *University of Canterbury*; Peter Andreae, *Victoria University of Wellington*; Anthony Robins, *University of Otago*

Computer science became available as a nationally assessed topic in NZ schools for the first time in 2011. We review the introduction of computer science as a formal topic, including the level of adoption, issues that have arisen in the process of introducing it, and work that has been undertaken to address those issues.

14:10 Web Science: expanding the notion of Computer Science

Su White, *University of Southampton*; Michalis Vafopoulos, *Aristotle University of Thessaloniki*

Academic disciplines which experience rapid change face problems maintaining teaching programs. Web Science: the science of decentralized information systems is fundamentally interdisciplinary, encompassing the technologies and engineering of the Web alongside associated emerging human, social and organizational practices. As work on the Computer Curricula 2013 is underway, it seems timely to ask what place Web Science may have in the curriculum landscape. This paper discusses the role and place of Web Science in the computing disciplines. It provides an account of work towards defining a curriculum for Web Science utilizing novel methods to support and elaborate curriculum definition and review. The findings of a desk study of existing related curriculum recommendations are presented.

14:35 Educating the Educator Through Computation: What GIS Can Do For Computer Science

John Barr and Ali Erkan, *Ithaca College*

We designed a system where non-computational faculty members (along with undergraduates) enroll in an introductory, multidisciplinary, open source Geographic Information System (GIS) course to experience integrative learning as students. The faculty participants are subsequently required to integrate their newly acquired expertise with their own disciplinary teaching and research; the necessary time commitment is compensated by a three-credit teaching load reallocation. Our hypothesis is that increasing the general faculty's appreciation of computation (in the context of integrative learning) is an indirect yet effective and scalable way to reach a wider group of students to convey our fundamental disciplinary message: computing is more than programming and computing empowers people.

PAPERS Active Learning I 306A

Chair: Robert England, *Transylvania University*

13:45 An Experience Report: On The Use Of Multimedia Pre-Instruction And Just-In-Time Teaching In A CS1 Course

Paul Carter, *University of British Columbia*

We describe an experience using online multimedia instruction and just-in-time teaching in an introductory programming course. Survey data has shown that students are strongly in favour of the approach. A series of screencasts was developed to replace the traditional lecture component of the course. Students were asked to review a small number of screencasts before each class and were assessed on their comprehension at the start of class using a series of "clicker" questions. A just-in-time mini-lecture was provided in response to the initial assessment, on an as-needed basis. The remaining class time was devoted to small-group exercises.

14:10 Using JiTT in a Database Course

Alexandra Martinez, *Universidad de Costa Rica*

This paper describes our experience using the Just-in-Time Teaching (JiTT) technique in an undergraduate database course for computer science majors during two semesters. JiTT was implemented by giving the students reading assignments and asking them to complete web-based reading tests the day before class, so that the instructor could detect weaknesses in students' understanding of the material and adjust the lesson plan just in time for the next day class. Based on surveys as well as on exams and course grades, we found a significant improvement on the students interest in the course and learning of the material.

14:35 Process Oriented Guided Inquiry Learning (POGIL) for Computer Science

Clifton Kussmaul, *Muhlenberg College*

This paper describes an ongoing project to develop activities for computer science (CS) using process oriented guided inquiry learning (POGIL). First, it reviews relevant background on effective learning and POGIL, compares POGIL to other forms of active learning, and describes the potential of POGIL for CS. Second, it describes a sample POGIL activity, including the structure and contents, student and facilitator actions during the activity, and how activities are designed. Third, it summarizes current progress and plans for a NSF TUES project to development POGIL materials for CS. Finally, it discusses student feedback and future directions.

PAPERS Communication Skills 306B

Chair: James Early, *SUNY Oswego*

13:45 Integrating Communication Skills into the Computer Science Curriculum

Katrina Falkner and Nickolas Falkner, *University of Adelaide*

Computer Science majors must be able to communicate effectively. There is considerable work in the area of communication skills development, positioned in terms of curriculum guidelines for effective communication skills development, and example communication skills activities. However, this research is deficient in detailed, contextualised methodologies and frameworks for the development of communication skills within the Computer Science curriculum. We present a new methodology, building upon well established theoretical frameworks, designed to assist academics in the development of communication skills activities integrated with discipline content across the curriculum.

14:10 'Explain in Plain English' Questions: Implications for Teaching

Laurie Murphy, *Pacific Lutheran University*; Renée McCauley, *College of Charleston*; Sue Fitzgerald, *Metropolitan State University*

This paper reports on a replication of work by Corney, Lister and Teague who performed a longitudinal study of novice programmers, looking for relationships between ability to 'explain in plain English' the meaning of a code segment and success in writing code later in the semester. The study extends the work of Corney, Lister and Teague by qualitatively evaluating 'explain in plain English' responses to gain a deeper understanding of student misconceptions. Statistical results from this study are similar to those of Corney, Lister and Teague. Results highlight students' fragile knowledge and suggest the need for assessment and instruction of basic concepts later into the term than instructors are likely to expect.

14:35 The Impact of Question Generation Activities on Performance

Andrew Luxton-Reilly, Daniel Bertinshaw, Paul Denny, Beryl Plimmer and Robert Sheehan, *The University of Auckland*

Recent interest in student-centric pedagogies have resulted in the development of numerous tools that support student-generated questions. Previous evaluations of such tools have reported strong correlations between student participation and exam performance, yet the level of student engagement with other learning activities in the course is a potential confounding factor. We show such correlations may be explained by other factors, and we undertake a deeper analysis that reveals evidence of the positive impact question-generation activities have on student performance.

Supporter Session Google 305A

The MIT Center for Mobile Learning and the Future of App Inventor

Hal Abelson, Professor of Computer Science and Engineering, *MIT*
Mark Friedman, Former App Inventor Project Lead, *Google*

The MIT Media Lab applies an unorthodox research approach to envision the impact of emerging technologies on everyday life—including technologies used in education. Through a generous grant from Google, the Media Lab recently expanded on this work by establishing the MIT Center for Mobile Learning led by Hal Abelson, Mitch Resnick, and Eric Kopfler. The Center's work revolves around the principle that mobile technology can fulfill its potential to enhance education only if teachers and learners can create new mobile technologies, not merely experience them as consumers. This session will discuss the Center's three new initiatives: Scratch, TaleBlazer, and App Inventor.

Friday, 15:00 to 15:45

Break and Exhibits Exhibit Hall A

Friday, 15:00 to 16:30

NSF Showcase #4 Exhibit Hall A

- **FRABJOUS CS - Framing a Rigorous Approach to Beauty and Joy for Outreach to Underrepresented Students in Computing at Scale,** Daniel Garcia, Tiffany Barnes, and Brian Harvey
- **Exploring Computer Science - An Equitable Learning Model for Democratizing K-12 Computer Science Education,** Joanna Goode and Gail Chapman
- **How to Broaden Participation and Scale Up Computational Thinking by Bringing Game Design into Middle Schools,** Alexander Repenning, Kris Gutierrez, Jeffrey Kidder, and David Webb
- **Cybersecurity Laboratory: Enhancing the Hands-on Experience in Cybersecurity Education,** Susanne Wetzel

Friday, 15:00 to 17:00

POSTER Poster Session II Exhibit Hall A

Implementing and Assessing a Blended CS1 Course
John Wright, *Juniata College*

Designing with Projects in Mind: An Approach for Creating Authentic (and Manageable) Programming Projects
Scott Turner, *UNC Pembroke*

Integrating Elementary Computational Modeling and Programming Principles
Jose Garrido, *Kennesaw State University*

RoboLIFT: Simple GUI-Based Unit Testing of Student-Written Android Applications
Anthony Allevato and Stephen Edwards, *Virginia Tech*

OpenDSA: A Creative Commons Active-eBook
Eric Fouh, Maoyuan Sun and Clifford Shaffer, *Virginia Tech*

Active Learning in Computer Science Education Using Meta-Cognition
Murali Mani and Quamrul Mazumder, *University of Michigan, Flint*

Dynamic Programming Across the CS Curriculum
Yana Kortsarts, *Widener University*; Vasily Kolchenko, *New York City College of Technology The City University of New York*

50 Ways to be a FOSSer: Simple Ways to Involve Students & Faculty
Clifton Kussmaul, *Muhlenberg College*; Heidi Ellis, *Western New England University*; Greg Hislop, *Drexel University*

Teaching Computer Science and Programming Concepts Using LEGO NXT and TETRIX Robotics, and Computer Science Unplugged Activities
Daniela Marghitu, Taha Ben Brahim and John Weaver, *Auburn University*

Using POGIL to Teach Students To Be Better Problem Solvers
Helen Hu, *Westminster College*

Developing a Gaming Concentration in the Computer Science Curriculum at an HBCU
Jinghua Zhang and Elva Jones, *Winston-Salem State University*

OSSIE: An Open Source Software Defined Radio (SDR) Toolset for Education and Research
Jason Snyder, *Virginia Tech*

Implementing a Communication Intensive Core Course in the CS Curriculum: A Survey of Methods
Jean French, *Coastal Carolina University*

The Cross-Curriculum Mobile Computing Labware for CS
Liang Hong, *Tennessee State University*; Kai Qian and Dan Lo, *Southern Polytechnic State University*; Yi Pan, Yanqing Zhang and Xiaolin Hu, *Georgia State University*

Teaching Cryptography Using Hands-on Labs
Li Yang and Joseph Kizza *University of Tennessee at Chattanooga*; Andy Wang, *Southern Polytechnic State University*; Chung-Han Chen, *Tuskegee University*

From Drawing to Programming Attracting Middle-School Students to Programming through Self-Disclosing Code
Jennelle Nystrom, Pelle Hall, Andrew Hirakawa and Samuel Rebelsky, *Grinnell College*

Proposed Revisions to the Social and Professional Knowledge Area for CS2013
Carol Spradling, *Northwest Missouri State University*; Florence Appel, *Saint Xavier University*; Elizabeth Hawthorne, *Union County College*

A Better API for Java Reflection
Zalia Shams, *Virginia Tech*

Hands-on Labs for a Mini-Course on Mobile Application Development
Qusay H. Mahmoud, Nicholas Mair, Younis Mohamed and Sunny Dhillon, *University of Guelph*

CEOHP Evaluation, Evolution, and Archival
Vicki Almstrum, *Strayer University*; Barbara Owens, *Southwestern University*; Mary Last, *CEOHP*; Deepa Muralidhar, *North Gwinnett High School*

CodeTrainer Teacher Authoring System: Facilitating User-Created Content in an Intelligent Tutoring System
Christy McGuire, Thomas Harris and Jonathan Steinhart, *Tutor Technologies, Inc*; Leigh Ann Sudol-DeLyser, *Carnegie Mellon University*

Comparing Feature Sets within Visual and Command Line Environments and their effect on Novice Programming
Edward Dillon, Jr., Monica Anderson-Herzog and Marcus Brown, *University of Alabama*
Exploring Connected Worlds
Jeffrey Forbes, *Duke University*

Teaching Parallel Computing with Higher-level Languages and Compelling Examples
Jens Mache, Christopher T. Mitchell and Julian H. Dale, *Lewis & Clark College*; David P. Bunde, Casey Samoore, Sung Joo Lee and Johnathan Ebbers, *Knox College*

SPECIAL SESSION — Understanding NSF Funding Opportunities — 301AB

Participants: Suzanne Westbrook, Victor Piotrowski, Jeff Forbes, Harriet Taylor and Mimi McClure, *National Science Foundation*

This session highlights programs in the National Science Foundation's Division of Undergraduate Education, Office of Cyberinfrastructure and Directorate of Computer and Information Science and Engineering. The focus is on providing descriptions of several programs of interest to college faculty and discussing the requirements and guidelines for programs in these areas. It includes a description of the proposal and review processes as well as strategies for writing competitive proposals. Participants are encouraged to discuss procedural issues with the presenters.

PANEL — Teaching Outside the Text — 305B

Chair: Lester Wainwright, *Charlottesville High School*

Participants: Renee Ciezki, *Estrella Mountain Community College*; Barbara Ericson, *Georgia Institute of Technology*; Glen Martin, *TAG Magnet High School*

We know that students bring diverse experiences and an assortment of learning styles into our classrooms. We greet them and hand out a syllabus listing the required book(s). One size does not fit all when it comes to textbooks. In this session, participants will discover teaching activities that can be used to supplement any text: hands-on, interesting and fun activities that help students understand CS topics. Members of the AP Computer Science-A Development Committee will share these resources and lead a discussion of proven strategies and lesson ideas for teaching outside the textbook.

SPECIAL SESSION — Computer Engineering Review Task Force Report — 306C

Chair: John Impagliazzo, *Hofstra University*

Participants: Susan Conry, *Clarkson University*; Eric Durant, *Milwaukee School of Engineering*; Andrew McGettrick, *University of Strathclyde*; Timothy Wilson, *Embry-Riddle Aeronautical University*; Mitch Thornton, *Southern Methodist University*

The ACM and the IEEE Computer Society created the CE2004 Review Task Force (RTF) and charged it with the task of reviewing and determining the extent to which the CE2004 document required revisions. The RTF recommended keeping the structure and the vast majority of the content of the original CE2004 document. It also recommended that contemporary topics should be strengthened or added while de-emphasizing other topics. Additionally, the RTF recommended that the two societies form a joint special-purpose committee to update and edit the earlier document and to seek input and review from the computer engineering industrial and academic communities. The presentation will provide insights in the RTF findings and thoughts on how a future computer engineering report might evolve.

PAPERS Projects 302A

Chair: Jeff Gray, *University of Alabama*

15:45 Social Sensitivity and Classroom Team Projects: An Empirical Investigation

Lisa Bender, Gursimran Walia, Krishna Kambhampaty and Kendall E. Nygard, *North Dakota State University*; Travis E. Nygard, *Ripon College*

Team work is the norm in major development projects and industry is continually striving to improve team effectiveness. Researchers have established that teams with high levels of social sensitivity tend to perform well when completing a variety of specific collaborative tasks. Our claim is that, the social sensitivity can be a key component in predicting the performance of teams that carry out major projects. This paper reports the results from an empirical study that investigates whether social sensitivity is correlated with the performance of student teams on large semester-long projects. The overall result supports our claim. It suggests, therefore, that educators in computer-related disciplines should take the concept of social sensitivity seriously as an aid to productivity.

16:10 Taming Complexity in Large Scale Systems Projects

Shimon Schocken, *IDC Herzliya*

Engaging students in large software development projects is an important objective, since it exposes design and programming challenges that come to play only with scale. Alas, large scale projects can be monstrously complex – to the extent of being infeasible in academic settings. We describe a framework and a set of principles that enable students to develop large scale systems – e.g. a complete hardware platform or a compiler – in several semester weeks.

16:35 An Approach for Evaluating FOSS Projects for Student Participation

Heidi Ellis, *Western New England University*; Michelle Purcell and Gregory Hislop, *Drexel University*

Free and Open Source Software (FOSS) offers a transparent development environment and community in which to involve students. Students can learn much about software development and professionalism by contributing to an on-going project. However, the number of FOSS projects is very large and there is a wide range of size, complexity, domains, and communities, making selection of an ideal project for students difficult. This paper addresses the need for guidance when selecting a FOSS project for student involvement by presenting an approach for FOSS project selection based on clearly identified criteria. The approach is based on several years of experience involving students in FOSS projects.

PAPERS Alice and Scratch 302B

Chair: Kelly Powers, *Advanced Math & Science Academy Charter School*

15:45 Integrating Computing into Middle School Disciplines Through Projects

Susan Rodger, Melissa Dalis, Peggy Li, Liz Liang and Wenhui Zhang, *Duke University*; Chitra Gadwal, *UMBC*; Francine Wolfe, *Benedictine College*

For four years we have been integrating computing into a variety of middle school disciplines via the Alice programing language. This paper describes our efforts over the past two years in creating model projects for students to build in all disciplines, and our most recent focus on science and mathematics projects. For science we have introduced experiments in Alice and the tools needed for them. In mathematics we have created projects to increase their understanding of programming and to use the projects to increase their understanding of mathematics. We also discuss our efforts in workshops to teach K-12 teachers Alice and an analysis of the teachers' lesson plans and worlds developed in the most recent workshop.

16:10 Children Learning Computer Science Concepts via Alice Game-Programming

Linda Werner, *University of California, Santa Cruz*; Shannon Campe and Jill Denner, *ETR Associates*

Programming environments that incorporate drag-and-drop methods and many predefined objects and operations are being widely used in K-12 settings. But can students as young as those in middle school learn complex computer science concepts using these programming environments when computer science is not the focus of the course? In this paper, we describe a semester-long game-programming course where 325 middle school students used Alice. We report on our analysis of 225 final games where we measured the frequency of successful execution of programming constructs. Our results show that many games exhibit successful uses of high level computer science concepts such as student-created abstractions, concurrent execution, and event handlers.

16:35 The Writers' Workshop for Youth Programmers: Digital Storytelling with Scratch in Middle School Classrooms

Quinn Burke and Yasmin B. Kafai, *University of Pennsylvania*

This study investigates the potential to introduce basic programming concepts to middle school children within the context of a classroom writing-workshop. In this paper we describe how students drafted, revised, and published their own digital stories using the introductory programming language Scratch and in the process learned fundamental CS concepts as well as the wider connection between programming and writing as interrelated processes of composition.

PAPERS Active Learning II 306A

Chair: Douglas Kranch, *North Central State College*

15:45 A Software Craftsman's Approach to Data Structures

Arto Vihavainen, Matti Luukkainen and Thomas Vikberg, *University of Helsinki*

Data Structures (CS2) courses and course books do not usually put much emphasis in the process of how a data structure is engineered or invented. Instead, algorithms are readily given, and the main focus is in the mathematical complexity analysis of the algorithms. We present an alternative approach on presenting data structures using worked examples, i.e., by explicitly displaying the process that leads to the invention and creation of a data stucture and its algorithms. Our approach is heavily backed up by some of the best programming practices advocated by the Agile and Software Craftmanship communities and it brings the often mathematically oriented CS2 course closer to modern software engineering and practical problem solving, without a need for compromise in proofs and analysis.

16:10 Jutge.org: An Educational Programming Judge

Petit Jordi and Roura Salvador, *Universitat Politècnica de Catalunya*; Giménez Omer, *Google*

Jutge.org is an educational programming judge where students can solve more than 800 problems using 22 programming languages. The verdict of their solutions is computed using exhaustive test sets run under time, memory and security restrictions. By contrast to many popular online judges, Jutge.org is designed for students and instructors: On one hand, the problem repository is mainly aimed to beginners, with a clear organization and gradding. On the other hand, the system is designed as a virtual learning environment where instructors can administer their own courses, manage their roster of students and tutors, add problems, attach documents, create lists of problems, assignments, contests and exams. This paper presents Jutge.org and offers some case studies of courses using it.

16:35 Integrating Formal Verification in an Online Judge for e-Learning Logic Circuit Design

Javier De San Pedro, Josep Carmona, Jordi Cortadella and Jordi Petit, *Universitat Politècnica de Catalunya*

This paper investigates the use of formal verification techniques to create online judges that can assist in teaching logic circuit design. Formal verification not only contributes to give an exact assessment about correctness, but also saves the instructor the tedious task of designing test cases. The paper explains how formal verification has been integrated in an online judge. It also describes the courseware created for a course on logic circuits and the successful experience of using it in a one-week summer course with students from secondary and high school.

PAPERS Non-majors 306B

Chair: Derek Schuurman, *Redeemer University College*

15:45 Computing For STEM Majors: Enhancing Non CS Majors' Computing Skills

Joel Adams and Randall Pruim, *Calvin College*

One of the challenges facing the U.S. technological workforce is that fewer college graduates are being prepared for computing careers. Besides trying to attract more CS majors, another approach is to (i) design a computing curriculum that appeals to students and faculty from non-CS disciplines, (ii) use special scholarships to attract students to that curriculum, and (iii) sponsor faculty development workshops for non-CS departments. In this paper, we detail this approach, using a new introductory course oriented to science majors, and scholarships funded by the National Science Foundation Scholarships for Science, Technology, Engineering, and Mathematics (NSF S-STEM) program. We also present several success stories that this approach has produced in its first two years.

16:10 Operations Research: Broadening Computer Science In A Liberal Arts College

Barbara Anthony, *Southwestern University*

Operations research, while not traditionally taught at many small or liberal arts colleges, can be a significant asset to the offerings of a computer science department. Often seen as a discipline at the intersection of mathematics, computer science, business, and engineering, it has great interdisciplinary potential and practical appeal, allowing for recruitment of students who may not consider taking a CS0 or CS1 course. Offering this course not only benefited computer science majors who appreciated the applications and different perspectives, but it provided a means for the department to serve a wider population, increased interdisciplinary education, and resulted in a filled-to-capacity upper-level course in computer science for the first time in recent memory.

16:35 Beyond Competency: A Context-Driven CS0 Course

Jeff Cramer and Bill Toll, *Taylor University*

In the process of revising our general education course we attempted to answer the question "What should a graduate of a liberal arts university understand about computational technology?" University students may know more about narrow areas of technology but the true impact on their lives cannot be understood without an appreciation for the nature and limitations of the technology. This paper presents a set of assumptions about the impact of technology on individuals and society and describes elements of a computing context designed to enable students to critically evaluate the technology that has such an impact on their lives. Assessment of the approach indicates that students are more aware of the impact of technology and the importance of an understanding of the technology.

Supporter Session **Microsoft** **305A**

Cloud in a Classroom: Faculty Experiences

Nilajan Banerjee, *University of Arkansas*
Chia-Chi Teng, *Brigham Young, University*
Alexander Schmidt, *University of Potsdam*
Moderator: Arkady Retik, *Microsoft*

Cloud computing introduces new and exciting opportunities for computing industry. To realize the potential of cloud computing in higher education, one must think about the cloud as a holistic platform for creating new services, new experiences, and new methods for research and teaching. Pursuing these goals in the current set of the CS courses presents a broad range of interesting questions. Come along to hear about the cloud-based teaching resources available and how they have been used in universities world-wide. This panel will provide an opportunity for SIGCSE attendees to hear from faculty who have been teaching CS courses using Windows Azure ask questions and discuss and share their own experiences.

Friday, 17:10 to 17:55

SIGCSE Business Meeting **302A**

Friday, 18:00 to 18:45

Going Greenfoot **302C**
CCSC Business Meeting **305B**

Friday, 18:00 to 18:50

NCWIT Academic Alliance Reception **305A**

Friday, 19:00 to 22:00

16. Intellectual Property Law Basics for Computer Science Instructors 205

David G. Kay, *UC Irvine*

Increasingly the practice of computing involves legal issues. Patenting algorithms, domain name poaching, downloading music, and “re-using” HTML and graphics from web sites all raise questions of intellectual property (IP) law (which includes patents, copyrights, trade secrets, and trademarks). In the classroom, computer science educators often confront questions that have legal ramifications. The presenter, who is both a computer scientist and a lawyer, will introduce the basics of intellectual property law to give instructors a framework for recognizing the issues, answering students’ questions, debunking the most egregious misconceptions about IP, and understanding generally how the law and computing interact. All CS educators are welcome; no computer is required.

17. Teaching and Learning Computing via Social Gaming with Pex4Fun 301A

Nikolai Tillmann, Jonathan de Halleux and Judith Bishop, *Microsoft Research*; Tao Xie, *North Carolina State University*

Pex4Fun (pexforfun.com) is a web-based serious gaming environment for teaching computing at many levels, from high school all the way through graduate courses. Unique to the Pex4Fun experience is a cloud-based program evaluation engine based on dynamic symbolic execution and SMT-solving, which provides customized feedback to the student and automated grading for the teacher. Thus, Pex4Fun connects teachers, curriculum authors, and students in a social experience, tracking and streaming progress updates in real time. This workshop involves creating and teaching course materials at Pex4Fun. Participants should bring a laptop computer. The intended audience includes all levels of CS educators who are interested in integrating educational technology in their teaching environments.

18. Welcome to Makerland: A First Cultural Immersion into Open Source Communities 301B

Mel Chua, *Purdue University*; Sebastian Dziallas, *Franklin W. Olin College of Engineering*; Heidi Ellis, *Western New England University*; Greg Hislop, *Drexel University*; Karl Wurst, *Worcester State University*

Participating in free and open source (FOSS) software communities provides students with authentic learning while supplying instructors with a wide variety of educational opportunities including coding, testing, documentation, professionalism and more. However, instructors may be unfamiliar with how FOSS communities work and therefore may be reluctant to involve students. This workshop is a subset of material used in Red Hat’s Professors’ Open Source Summer Experience (http://communityleadershipteam.org/posse), now in its third year of successfully providing a ramp to FOSS projects for instructors. These instructors have demonstrated success in involving their students in FOSS communities where students have contributed code, interface design, and more. Laptop Required.

19. Computational Art and Creative Coding: Teaching CS1 with Processing 302A

Dianna Xu and Deepak Kumar, *Bryn Mawr College*; Ira Greenberg, *Southern Methodist University*

This workshop showcases a new approach to teaching CS1 using computational art as a context. Participants will be introduced to the Processing programming language and environment, designed for the construction of 2D and 3D visual forms. Its IDE is lightweight, but well-suited for the rapid proto-typing needed for dynamic visual work. We hope to bring the excitement, creativity, and innovation fostered by Processing into the computer science education community. Instructors of all experience levels are welcome. Hands-on portion of the workshop will enable participants to explore Processing and create visual effects on the fly. Course materials and handouts detailing the software and teaching resources will be given out. Laptop Required.

20. AP CS Principles and The Beauty and Joy of Computing Curriculum 302B

Daniel Garcia, Brian Harvey, Nathaniel Titterton and Luke Segars, *UC Berkeley*; Tiffany Barnes, *University of North Carolina, Charlotte;* Eugene Lemon, *Ralph J Bunche High School*; Sean Morris, *Albany High School*; Josh Paley, *Henry M. Gunn High School*

The Beauty and Joy of Computing (BJC) is an introductory computer science curriculum developed at UC Berkeley (and adapted at UNC Charlotte) intended for high school juniors through university non-majors. It was used in two of the five initial pilot programs for the AP CS Principles course being developed by the College Board and the NSF. Our overall goal is to support the CS10K project by preparing instructors to teach the AP CS Principles course through the BJC curriculum. We will share our experiences as instructors of the course at the university and high school level, provide a glimpse into a typical week of the course, and share details of NSF-funded summer professional development opportunities. Laptop Required.

21. Peer Instruction in the CS Classroom: A Hands-On Introduction 302C

Daniel Zingaro, *University of Toronto*; Cynthia Bailey-Lee and Beth Simon, *University of California, San Diego*; John Glick, *University of San Diego*; Leo Porter, *Skidmore College*

We introduce participants to Peer Instruction (PI): an active learning technique applicable to the teaching of many subjects, including CS. In PI, students work together to exchange perspectives and answer challenging conceptual questions, and are supported by short teaching segments. We will introduce and motivate PI, demonstrate its use in combination with a clicker system, and show that PI is much more than the use of clickers. Participants will work in groups to develop new PI questions addressing challenges to their students' learning, and discuss numerous pedagogical benefits conferred through PI. Instructors interested in increasing engagement in any CS course may attend. Participants are encouraged to bring current lecture materials. Laptop Optional.

22. Incorporating Software Architecture in the Computer Science Curriculum 307

Martin Barrett, *East Tennessee State University*; Steve Chenoweth, *Rose-Hulman Institute of Technology*; Larry Jones, *Software Engineering Institute*; Amine Chigani, *Virginia Tech*; Ayse Bener, *Ryerson University*; Mei-Huei Tang, *Gannon University*

This workshop introduces and incorporates software architecture concepts into CS and SE curricula. Participants will learn techniques used in industry to specify quality attributes critical to architecture and use those attributes to drive the system structure using common architectural styles. Exercises will demonstrate pedagogical uses of the techniques in CS and SE classes. Sample computer science curricula with courses that integrate workshop material will be presented. Presenters will lead a brainstorming session to help participants develop practical methods for using the material in their courses. Participants will become part of a community of educators sharing educational resources in software architecture. Laptop Optional.

23. Parallelism and Concurrency for Data-Structures & Algorithms Courses 305A

Robert Chesebrough, *Intel Corporation*; Johnnie Baker, *Kent State University*

This workshop is inspired by Dan Grossman's SIGCSE 2011 workshop on Data Abstractions. We review C/C++ conversions of the original Java-based materials and will include material from the Parallel Algorithms course at Kent State. The workshop will appeal to data-structure and algorithms course instructors. Workshop topics will include divide and conquer approaches, work sharing concepts, and a scoped locking scheme in OpenMP for C++ classes. This material is driven via core data-structure examples (queues, sorting, reductions, etc.) and using a Fork/Join Framework found in OpenMP and Intel® Cilk Plus and Intel® Threading Building Blocks. Participants will write parallel programs and test them on the Intel® Many-core Testing Lab. Laptop Required.

24. ARTSI Robotics Roadshow-in-a-Box: Turnkey Solution for Providing Robotics Workshops to Middle and High School Students 305B

Monica Anderson, *The University of Alabama*; Dave Touretzky, *Carnegie Mellon University*; Chutima Boonthum-Denecke, *Hampton University*

In this half-day tutorial, we will introduce the ARTSI "Robotics Roadshow-in-a-Box (RRIB)", a single point resource for those getting started in robotics outreach. The RRIB is a kit which contains robots, software and prepared materials for providing robotics workshops for middle and high school students that focuses on showing computer scientists as problem solvers and not just programmers through activities with a larger context. The RRIB fills a need for materials that are accessible to those who may have limited knowledge of robotics or limited experience in middle school outreach, whether that is undergraduate students or faculty researchers who might have limited outreach experience or preparation time. Laptop Required.

25. Program by Design: From Animations to Data Structures 306A

Kathi Fisler, *WPI*; Stephen Bloch, *Adelphi University*

We present the Program by Design introductory CS curriculum through the lenses of graphics, animations, algebra, and data structures. Animations programming is popular for CS1, but many such curricula lack clean paths into CS2. Program by Design is different. Using and reinforcing concepts from algebra, students learn to write animations (including standard topics such as model/view separation and event-handling), then move seamlessly into working with structured data, lists, trees, and objects. The curriculum emphasizes design, testing, and writing maintainable programs, without losing the engagement of animations. The workshop uses lectures and hands-on exercises to provide high- school and college teachers an overview of the approach. See www.programbydesign.org. Laptop Optional.

26. CS Outreach with App Inventor 306B

Michelle Friend, *Stanford University*; Jeff Gray, *University of Alabama*

Mobile phone programming can provide teens an authentic and engaging hook into computer science. With App Inventor, developed by Google and moved to MIT, programming Android apps is as easy as clicking blocks together. App Inventor has been used successfully in after school programs, roadshows, summer camps, teacher workshops, and computer science classrooms from middle school through college. Participants will get an overview of App Inventor including project ideas and sample student code, hear outreach planning suggestions, write programs, develop outreach plans, and see how the Java Bridge helps transition from App Inventor to Java. Even the most time-stretched professor or teacher can encourage students in computer science with App Inventor. Laptop Required.

27. Making Mathematical Reasoning Fun: Tool-Assisted, Collaborative Techniques 306C

Jason Hallstrom and Murali Sitaraman, *Clemson University*; Joe Hollingsworth, *Indiana University Southeast*; Joan Krone, *Denison University*

Is it possible to excite students about learning the mathematical principles that underly high-quality software? Can we teach them to apply these principles using modern software tools? Can this be accomplished without displacing existing content? Yes, with the right pedagogical principles, teaching tools, and classroom exercises. This hands-on laboratory will introduce a set of principles, tools, and exercises that have proven to work. By adopting one content module at a time, educators will better prepare students to reason rigorously about the software they develop and maintain. Fees for this workshop will be covered for a limited number of attendees through an NSF award; limited travel support is also available. Laptop Required.

Saturday, 8:30 to 9:45

PANEL Nifty Assignments 301AB

Chair: Nick Parlante and Julie Zelenski *Stanford University*

Participants: Daniel Zingaro, *University of Toronto*; Kevin Wayne, *Princeton University*; Joshua Guerin, *University of Kentucky*; Stephen Davies, *University of Mary Washington*; Dave O'Hallaron, *Carnegie Mellon University;* Debby Keen, *University of Kentucky* and Zachary Kurmas, *Grand Valley State*

I can worry about the strategy of my syllabus, and I can fret over my lectures. Nonetheless, I am always struck that what my students really learn and enjoy in the course depends very much on the assignments. Great assignments are hard to dream up and time-consuming to develop. With that in mind, the Nifty Assignments session is all about promoting and sharing the ideas and concrete materials of successful assignments.

PANEL Update on the CS Principles Project 305B

Chair: Amy Briggs, *Middlebury College*

Participants: Owen Astrachan, *Duke University*; Jan Cuny, *National Science Foundation*; Lien Diaz, *College Board*; Chris Stephenson, *Computer Science Teachers Association*

The CS Principles Project is a collaborative effort to develop a new introductory course in computer science, accessible to all students. Computer Science educators at all levels have worked together on the development of the new curriculum under the direction of the College Board with support from the National Science Foundation. This special session provides an opportunity for the CS Principles project leaders to report on recent updates and new directions, and to engage in discussion on all aspects of the project with SIGCSE participants.

PANEL Implementing Evidence-Based Practices makes a Difference in Female Undergraduate Enrollments 306C

Chair: Wendy DuBow, *University of Colorado*

Participants: Elizabeth Litzler, *University of Washington*; Maureen Biggers, *Indiana University*; Mike Erlinger, *Harvey Mudd College*

While many departments are aware of promising and best practices for recruiting and retaining female students in undergraduate computing majors, there seems to be a drive to try novel approaches instead of evidence-based approaches. Developing a diverse student body requires active recruitment, inclusive pedagogy, meaningful curriculum, evaluation of progress, as well as student, faculty and institutional support. Given the intrinsic challenges of enacting change, departments could make it easier on themselves - and likely achieve better results - if they intentionally and systematically used practices that have been shown to be effective. This panel will present the rationale for implementing evidence-based practices and share the successes some departments have achieved by doing so.

Saturday, 8:30 to 10:10

PAPERS High School Collaborations 302A

Chair: Tim Bell, *University of Canterbury*

8:30 **Life Two Years After a Game Programming Course: Longitudinal Viewpoints on K-12 Outreach**

Antti-Jussi Lakanen, Ville Isomöttönen and Vesa Lappalainen, *University of Jyvaskyla*

In our faculty we have run week-long K-12 game programming courses now for three summers. In this paper we investigate what programming-related activities students do after they take a course, and what factors in the students' background relate to post-course programming. We also investigate a possible change in the students' interest towards higher education science studies. We find that most students continue programming after the course and that their interest towards science studies keeps increasing. In student background we observed some indicative trends, but did not find reliable explaining factors related to post-course programming or increased interest towards science studies.

8:55 **Reflections on Outreach Programs in CS Classes: Learning Objectives for "Unplugged" Activities**

Renate Thies, *Cusanus-Gymnasium Erkelenz and Technische Universität Dortmund*; Jan Vahrenhold, *Technische Universität Dortmund*

To provide a unified view of any scientific field, outreach programs need to realistically portray the subject in question. Consequently, topics and methods actually taught in Computer Science courses should to be touched upon in Computer Science outreach programs or, conversely, elements from successful Computer Science outreach programs should be used to enrich established courses in Computer Science. We follow up on the latter aspect and extract and classify learning objectives from the activities of the well-received Computer Science Unplugged program. Based upon this classification, we comment on where and to which extent these activities can be used to enrich teaching Computer Science in secondary education.

9:20 **Weaving a Tapestry Satellite Workshop to Support HS CS Teachers in Attracting and Engaging Students**

Ambareen Siraj, Martha Kosa and Summer Olmstead, *Tennessee Tech University*

In this paper, we describe the Tennessee Tech University (TTU) Tapestry Workshop for high school (HS) teachers. The Tapestry Workshop initiative, is a collaborative partnership between TTU, the University of Virginia (UVA) and HS teachers to share strategies, practices, and innovative ideas for teaching Computer Science (CS) effectively. This three-day professional development workshop utilized informational, technical, networking, activity-based, and discussion-oriented sessions geared towards attracting and engaging CS students. The workshop was a worthwhile professional development activity for both the organizers and attendees and contributed towards initiation of HS CS program locally.

9:45 **Who AM I? Understanding High School Computer Science Teachers' Professional Identity**
Lijun Ni and Mark Guzdial, *Georgia Institute of Technology*

We need committed, quality CS teachers for quality secondary computing education. Teacher education literature suggests that teachers' sense of commitment and (other aspects of) teaching profession is tightly linked with their teacher identity. However, the current educational system does not provide typical contexts for teachers to build a sense of identity as CS teachers. This study is intended to gain an initial understanding of CS teachers' perceptions about their professional identity and potential factors that might contribute to these perceptions. Our findings indicate that current HS teachers teaching CS courses do not necessarily identify themselves as CS teachers. They have different perceptions related to CS teaching. Four kinds of factors can contribute to these perceptions.

PAPERS Parallelism and Concurrency 302B

Chair: Jodi Tims, *Baldwin-Wallace College*

8:30 **Introducing Parallelism and Concurrency in the Data Structures Course**
Ruth E. Anderson and Dan Grossman, *University of Washington – Seattle*

We report on our experience integrating a 3-week introduction to multithreading in a required data structures course for 2nd-year computer science majors. We emphasize a distinction between parallelism and concurrency that teaches students to use extra processors effectively and enforce mutual exclusion correctly. The material fits naturally in the data structures course by having the same mix of algorithms, programming, and asymptotic analysis as the rest of the course. Our department has used this unit for 1.5 years and we report feedback from students, multiple instructors for the course, and students in a later course that uses threads. We developed a full set of course materials that have been adapted for use by instructors in various courses at five other institutions so far.

8:55 **Exploring Concurrency Using The Parallel Analysis Tool**
Brian Rague, *Weber State University*

One area of investigation that has become increasingly important across all levels of CS instruction is parallel computing. This paper describes the initial version of the Parallel Analysis Tool (PAT), a pedagogical tool designed to assist undergraduate students in visualizing concurrency and effectively connecting parallel processing to coding strategies. The PAT is a complete Java development environment, with an emphasis on (1) helping students to identify appropriate code locations where parallelization can be applied and (2) allowing students to subsequently examine the practical performance tradeoffs of these parallelization decisions in a laboratory setting. The Parallel Quotient supports the analysis of the relative benefits of employing various parallel programming strategies.

9:20 **Virtual Clusters for Parallel and Distributed Education**
Elizabeth Shoop, Eric Biggers, Malcom Kane, Devry Lin and Maura Warner, *Macalester College*; Richard Brown, *St. Olaf College*

The reality of multicore machines as a standard and the prevalence of distributed cloud computing has signaled a need for parallel and distributed computing to become integrated into the computer science curriculum. At the same time, operating system virtualization has become a common technique with open standard tools available to any practitioners. Virtual machines (VMs) installed on available computer lab resources can be used to simulate high-performance cluster computing environments. This paper describes two such virtual clusters in use at small colleges, reports on their effectiveness, and provides information about how to obtain the VMs for use in an educational lab setting.

9:45 **Cross Teaching Parallelism and Ray Tracing: A Project-based Approach to Teaching Applied Parallel Computing**
Chris Lupo and Zoe Wood, *Cal Poly State University*; Christine Victorino, *University of California, Santa Barbara*

This paper describes the integration of two undergraduate computer science courses to enhance student learning in parallel computing concepts. In this cross teaching experience we structured the integration of the two courses such that students studying parallel computing worked with students studying advanced rendering for approximately 30% of the quarter long courses. Working in teams, both groups of students saw the application of parallelization to an existing software project early in the curriculum of both courses. Motivating projects and performance gains are discussed, as well as student survey data on the effectiveness of the learning outcomes. Performance and survey data indicate a positive gain from the cross teaching experience.

PAPERS Mobile Computing 306A

Chair: Cyndi Rader, *Colorado School of Mines*

8:30 **Cabana: A Cross-platform Mobile Development System**
Paul E. Dickson, *Hampshire College*

Mobile application development is a hot topic in computer science education, and debate rages over which platform to develop on and what software to use for development. Cabana is a web-based application designed to enable development on multiple mobile platforms and to make application development easier. It uses an approach to application programming based on a wiring diagram that is supplemented with the ability to program directly using JavaScript. It is an ideal choice for application development in introductory computer science courses and for upper-level courses where the focus is on application design and not application programming. This paper introduces Cabana and describes its use in two different computer science courses.

8:55 Mobile Apps for the Greater Good: A Socially Relevant Approach to Software Engineering

Victor Pauca, *Wake Forest University*; Richard Guy, *University of Toronto*

Socially relevant computing has recently been proposed as a way to reinvigorate interest in computer science. By appealing to students' interest in helping others, it aims to give students life-changing experiential learning not typically achieved in the classroom, while providing software that benefits society at large. For the last two years, we have been using mobile device programming, agile methods, and real-world, socially relevant projects for teaching software engineering in a liberal arts Computer Science curricula. We report on teaching methods, student experiences, and products delivered by this approach. In particular, one of these products, Verbal Victor, is now a commercial and social entrepreneurship success in assistive technology for communication disabilities.

9:20 Using Mobile Phone Programming to Teach Java and Advanced Programming to Computer Scientists

Derek Riley, *University of Wisconsin – Parkside*

In this work the approach employing the Android mobile phone platform in an upper division computer science course to teach Java programming and other advanced computer science topics is presented. Mobile phones are growing influences in the computing market, but their strengths and popularity are rarely exploited in computer science classrooms. The aim of the course is to harness this enthusiasm to improve fluency in the Java language to afford an opportunity to learn how to work on large, complex projects and to enhance the students' preparedness for the job market. The ideas presented in this work could be adapted for improving learning in many courses across the computing curriculum.

9:45 RoboLIFT: Engaging CS2 Students with Testable, Automatically Evaluated Android Applications

Anthony Allevato and Stephen H. Edwards, *Virginia Tech*

Making computer science assignments interesting and relevant is a constant challenge for instructors of introductory courses. Android has become popular in these courses to take advantage of the increasing popularity of smartphones and mobile "apps." This has been shown to increase student engagement but it is only the first step, and we must continue to provide support for teaching methodologies that we have used in the past, such as test-driven development and automated assessment. We have developed RoboLIFT, a library that makes unit testing of Android applications approachable for students. Furthermore, by supporting existing automated grading techniques, we are able to sustain large student enrollments, and we evaluate the effects that using Android has had on student performance.

PAPERS Visualization 306B

Chair: Demian Lessa, *State University of New York at Buffalo*

8:30 **Highway Data and Map Visualizations for Educational Use**

James Teresco, *Siena College*

It is often a challenge to find interesting and appropriate data sets to use as examples to demonstrate graph data structures and algorithms. The data should include examples small enough to work through manually, but some large enough to demonstrate important behaviors of a structure or algorithm. It should be freely available in a convenient format and have some real-world relevance. Visualization of the data and results computed from it is helpful. This paper describes a collection of graph data sets generated from the Clinched Highway Mapping Project's highway data and some examples of their use in courses. The source data, the process used to convert the data into a more useful format, some examples of its use, and a visualization tool using the Google Maps API, are described.

8:55 **Experiments with Algorithm Visualization Tool Development**

Michael Orsega, *University of West Georgia*; Bradley Vander Zanden and Christopher Skinner, *University of Tennessee*

This paper presents the initial stages of a teaching tool named iSketchmate, intended for instructor use during lecture. iSketchmate allows users to create and manipulate splay trees through an animated GUI. It improves upon existing tools by providing (1) the ability to begin with any user-defined tree, (2) a history mechanism so tree operations can be repeated or changed, and (3) finer-grained animation within each operation so instructors may give further descriptions at intermediate steps within any given operation. Experiments showed iSketchmate users could produce significantly more diagrams and these diagrams were significantly more accurate than those made with pencil and paper.

9:20 **CSTutor: A Pen-Based Tool for Visualizing Data Structures**

Sarah Buchanan, Brandon Ochs and Joseph LaViola, *University of Central Florida*

We present CSTutor, a pen-based application for data structure visualization which allows the user to manipulate data structures through the recognition of handwritten symbols and gestures as well as edit the corresponding code. The UI consists of a sketching area where the user can draw a data structure in a way that is as natural as pen and paper. Running in parallel with the visualization is a code view window where the user can make changes to the source code and add functions which manipulate the data structure on the canvas in real time. We also present the results of a perceived usefulness survey. The results of the study indicate that the majority of students would find CSTutor helpful for learning data structures.

9:45 **ECVisual: A Visualization Tool for Elliptic Curve Based Ciphers**
Jean Mayo, Jun Tao, Jun Ma, Melissa Keranen and Ching-Kuang Shene, *Michigan Technological University*

This paper describes a visualization tool ECvisual that helps students understand and instructors teach elliptic curve based ciphers. This tool permits a user to visualize elliptic curves over the real field and over a finite field of prime order, perform arithmetic operations, do encryption and decryption, and convert plaintext to a point. The demo mode of ECvisual can be used for classroom presentation and self-study. With the practice mode, a user may go through steps in finite field computations, encryption, decryption and plaintext conversion. She may compute, and then check, the answer to each operation herself. The opportunity for self-study provides an instructor greater flexibility in selecting a lecture pace for this detail-filled material. Classroom evaluation was positive.

Student Research Competition - Undergraduate 305A

Student Research Competition - Graduate 302C

Saturday, 10:00 to 11:30

NSF Showcase #5 Exhibit Hall A

- **Personal Robots for CS1: Next Steps for an Engaging Pedagogical Framework,** Douglas Blank and Deepak Kumar
- **An Interactive Undergraduate Data Mining Course with Industrial Strength Projects and Hypertextbook,** Ping Chen and Irene Chen
- **Cyberlearning Environments with Emphasis on Social Approach and Collaboration,** Jafar Saniie, Erdal Oruklu, and Tricha Anjali
- **REU Site for Smart UAVs,** Saad Biaz and Wei-Shinn Ku

Saturday, 10:10 to 10:55

Break and Exhibits Exhibit Hall A

Saturday, 10:55 to 12:10

PANEL — Rediscovering the Passion, Beauty, Joy, and Awe: Making Computing Fun Again — 301AB

Chair: Daniel Garcia, *UC Berkeley*

Participants: Barbara Ericson, *Georgia Institute of Technology*; Joanna Goode, *University of Oregon*; Colleen Lewis, *UC Berkeley*

In his SIGCSE 2007 keynote, Grady Booch exhorted us to share the "passion, beauty, joy and awe" (PBJA) of computing. This led to a series of SIGCSE sessions that provided a forum for sharing; What we've done: Highlighting successful PBJA initiatives the presenters have undertaken or seen and wish to celebrate; What we should do (curriculum): Pointing out where our curriculum is lacking in PBJA, and how to fix it; How we should do it (pedagogy): Sharing how a change in attitude / focus / etc. can make strides to improving PBJA. This year we've invited 3 educators who have worked tirelessly to broaden participation of computing to underrepresented groups. The hope with this panel is to be able to explore best practices in outreach, in terms of extolling the PBJA of computing.

SPECIAL SESSION — Promoting Student-Centered Learning with POGIL — 305B

Participants: Helen Hu, *Westminster College* and Clifton Kussmaul, *Muhlenberg College*

POGIL (Process Oriented Guided Inquiry Learning) is a type of learning based on the principle that students learn more when they construct their own understanding. Rather than teaching by telling, POGIL instructors provide activities that guide students to discover concepts on their own. Students work in groups, encouraging them to discuss their findings with their peers. Not only do students learn the material better, but the process of discovery teaches them to be better problem solvers. This special session will provide SIGCSE attendees the opportunity to experience a POGIL activity. The presenters will share guided inquiry activities. We will discuss ways that POGIL may be used to transform computer science classes at all levels, from small schools to large universities.

SPECIAL SESSION Teaching Secure Coding - Report from Summit on Education in Secure Software 306C

Chair: Blair Taylor, *Towson University*

Participants: Matt Bishop, *University of California Davis*; Diana Burley, *George Washington University*; Steve Cooper, *Stanford University*; Ron Dodge, *United States Military Academy*

To help education design curricula that integrates principles and practices of secure programming, the National Science Foundation Directorates of Computer and Information Science and Engineering (CISE) and Education and Human Resources (EHR) jointly sponsored the Summit on Education in Secure Software (SESS), held in Washington, DC in October, 2010. The goal of the summit was to develop roadmaps showing how best to educate students and current professionals on robust, secure programming concepts and practices, and to identify both the resources required and the problems that had to be overcome. The goal of this session is to share some of the key findings and challenges identified by the summit and to actively engage the community in the discussions.

PAPERS Attracting Majors 302A

Chair: Nanette Veilleux, *Simmons College*

10:55 Starting with ubicomp: Using the SenseBoard to Introduce Computing

Marian Petre, Mike Richards and Arosha Bandara, *The Open University*

In this paper, we describe a new undergraduate module for novice students conducted entirely through distance learning: My Digital Life (TU100). The course has been designed to lower the barriers to creating programs that interact with the world; TU100's materials have been designed to excite, encourage, reassure and support learners who explore the novel topic of ubiquitous computing through playful experimentation. It introduces the fundamentals of computing by giving students the capability for programming a device, the SenseBoard, which has built-in input/output and sensors. Programming is done in Sense, an extension of Scratch, which scaffolds programming and reduces the syntax burden.

11:20 Improving First-Year Success and Retention through Interest-Based CS0 Courses

Michael Haungs, Christopher Clark, John Clements and David Janzen, *California Polytechnic State Unviersity, San Luis Obispo*

Many computer science programs suffer from low student retention rates. At Cal Poly, academic performance and retention rates among first-year computer science students are among the lowest on campus. In order to remedy this, we have developed a new CS0 course featuring different "tracks" that students can choose from. This allows students to learn the basics of programming, teamwork, and college-level study in a domain that is of personal interest. In addition, the course relies on classic Project-based Learning (PBL) approaches as well as a focus on both academic and non-academic factors shown to increase student retention. Initial assessment demonstrates positive results in the form of increased academic performance in post CS0 courses and student retention.

11:45 Reshaping The Image Of Computer Science In Only Fifteen Minutes (Of Class) A Week

Sara Sprenkle, *Washington and Lee University*; Shannon Duvall, *Elon University*

Low undergraduate enrollments in computer science will not meet the future demand of employers. Some reasons for the low enrollments are computer science's nerdy image, lack of understanding of the field, and low motivation for learning programming. We propose to change the image of computer science by exposing students to applications of computing and its impact on their lives through reading and discussing recent news articles in 15 minutes of class. We call this component of our courses the Broader Issues in computer science.

PAPERS OS and Distributed Computing 302B

Chair: William Mongan, *Drexel University*

10:55 Experiences Teaching MapReduce in the Cloud

Ariel Rabkin, Charles Reiss, Randy Katz and David Patterson, *UC Berkeley*

We describe our experiences teaching MapReduce in a large undergraduate lecture course using public cloud services. Using the cloud, every student could carry out scalability benchmarking assignments on realistic hardware, which would have been impossible otherwise. Over two semesters, over 500 students took our course. We believe this is the first large-scale demonstration that it is feasible to use pay-as-you-go billing in the Cloud for a large undergraduate course. Modest instructor effort was sufficient to prevent students from overspending. Average per-pupil expenses in the Cloud were under $45, less than half our available grant funding. Students were excited by the assignment: 90% said they thought it should be retained in future course offerings.

11:20 Developing Microlabs Using Google Web Toolkit

Barry Kurtz, James Fenwick and Philip Meznar, *Appalachian State University*

Closed labs provide hands-on experience in a supervised setting. Microlabs extend this approach into lecture with very short hands-on activities occurring in lecture. Programming microlabs were developed for a distributed computing course. This paper describes our logical microlabs where students solve conceptual problems that do not involve programming. These two microlab approaches are integrated into the Microlab Learning Cycle. Microlab activities should be usable with a wide variety of computing devices, including tablets. After experimenting with different development environments we have adopted the Google Web Toolkit. After presenting the current status of our activities, we discuss future directions for microlab development.

11:45 Teaching Operating Systems Using Android

Jeremy Andrus and Jason Nieh, *Columbia University*

The computing landscape is shifting towards mobile and embedded devices. To learn about operating systems, it is increasingly important for students to gain hands-on kernel programming experience in these environments, which are quite different from desktops and servers. We present our work to teach operating systems by leveraging Android, an open, commercially supported software platform increasingly used on mobile and embedded devices. We introduce a series of 5 Android kernel programming projects, and an Android virtual lab which gives students hands-on Android experience with minimal computing infrastructure. We used these projects and virtual lab to teach an introductory operating systems course. Over 80% of students surveyed enjoyed applying operating systems concepts to Android.

Winner: SIGCSE 2012 Best Paper Award

PAPERS Curricular Innovations and Research 306A

Chair: Michael Hewner, *Duke University*

10:55 Open Educational Resources in Computer Science Teaching

Christo Dichev and Darina Dicheva, *Winston Salem State University*

Open content and open access to resources are important factors in the innovation of Computer Science education. This paper presents a study aimed at gaining an understanding of the needs of Computer Science educators in terms of Open Educational Resources (OER): what kind of resources they need, when they need them, how they use them, and what are the barriers and the enablers for using OER. The results of the study are compared and analyzed in the context of the popular OER sites. The work contributes to the research on OER utilization and discovery.

11:20 Emergent Themes in a UI Design Hybrid-Studio Course

Yolanda Reimer, *University of Montana*; Katherine Cennamo, *Virginia Tech*; Sarah Douglas, *University of Oregon*

The goal of our research and teaching collaboration has been to learn more about how key aspects of pedagogy commonly incorporated in architecture and industrial design classes might positively impact the teaching of user interface (UI) design within a standard computer science curriculum. Toward that end, we studied a number of studio design courses, developed a set of curriculum guidelines, and analyzed the effectiveness of these guidelines as implemented in a UI design course. We discovered three emergent themes: 1) students need early and constant reminders that design is an iterative process involving user feedback and testing; 2) instructor modeling is critical; and 3) technology needs to be carefully managed at critical junctures throughout the class.

11:45 A Multilevel, Multidimensional Undergraduate Course and Lab Experience on Embedded Multimedia Systems

Dimitrios Charalampidis, *University of New Orleans*; James Haralambides, *Barry University*

Traditional curricular structures can be fragmented in that course inter-relationships or links between theories, methodologies, and practices, are not immediately recognized by the students. The completion of the course puzzle and the integration of course knowledge usually become evident only after graduation. This paper describes a course/lab implementation that offered students a unique opportunity to experience the full spectrum of course elements, namely, fundamentals of theory, algorithmic/hardware design and simulation, and implementation and testing on FPGAs all within a single framework. The course/lab design is a collaborative effort between the U. of New Orleans and Barry U. to ensure that the course/lab can be implemented successfully in diverse environments.

PAPERS CS Education Research 306B

Chair: Yana Kortsarts, *Widener University, Chester*

10:55 Effective Closed Labs in Early CS Courses: Lessons from Seven Terms of Action Research

Elizabeth Patitsas, *University of Toronto*; Steve Wolfman, *University of British Columbia*

We report on best practices we have established to teach first-year computer science students in closed laboratories, founded on over three years of action research in a large introductory discrete mathematics and digital logic course. Our practices have resulted in statistically significant improvements in student and teaching assistant perception of the labs. Specifically, we discuss our practices of streamlining labs to reduce load on students that is extraneous to the lab's learning goals; establishing a positive first impression for students and TAs in the early weeks of the term; and effectively managing the teaching staff, including weekly preparation meetings for TAs using and a gradual, iterative curriculum development cycle that engages all stakeholders in the course.

11:20 What Do Students Learn About Programming From Game, Music Video, And Storytelling Projects?

Joel Adams and Andrew Webster, *Calvin College*

Drag-and-drop learning environments like Alice and Scratch eliminate syntax errors, making them attractive as ways to introduce programming concepts to students. Having had students create games, music videos, and storytelling projects, we began to wonder: What programming constructs do students actually use and hence learn well enough to be able to apply when creating different kinds of projects? We conducted a quantitative analysis of a collection of over 300 different student projects, and found significant differences in how frequently the students creating those projects used variables, if statements, loops, and dialog constructs.

11:45 Bayesian Network Analysis of Computer Science Grade Distributions

Adam Anthony and Mitchell Raney, *Baldwin-Wallace College*

Time to completion is a major factor in determining the total cost of a college degree. In an effort to reduce the number of students taking more than four years to complete a degree, we propose the use of Bayesian networks to predict student grades, given past performance prerequisite courses. This is an intuitive approach because the necessary structure of any Bayesian network must be a directed acyclic graph, which is also the case for prerequisite graphs. We demonstrate that building a Bayesian network directly from the prerequisite graph results in effective predictions, and demonstrate a few applications of the resulting network in areas of identifying struggling students and deciding upon which courses a department should allocate tutoring resources.

Saturday, 12:30 to 14:30

SIGCSE Luncheon Ballroom BC

Through the Looking Glass: Talking about the World with Visualization

Fernanda Viégas and Martin Wattenberg, *Google*

Data visualization has historically been accessible only to the elite in academia, business, and government. It was "serious" technology, created by experts for experts. In recent years, however, web-based visualizations–ranging from political art projects to news stories–have reached audiences of millions. What will this new era of data transparency look like–and what are the implications for technologists who work with data? To help answer this question, we report on recent research into public data analysis and visualization. Some of our results come from Many Eyes, a "living laboratory" web site where people may upload their own data, create interactive visualizations, and carry on conversations. We'll also show how the art world has embraced visualization. We'll discuss the future of visual literacy and what it means for a world where visualizations are a part of political discussions, citizen activism, religious discussions, game playing, and educational exchanges.

Saturday, 15:00 to 18:00

Workshops

28. Snap! (Build Your Own Blocks) 301A

Brian Harvey, Daniel Garcia and Luke Segars, *University of California, Berkeley*; Josh Paley, *Henry M. Gunn High School*

This workshop is for high school and college teachers of general- interest (”CS 0”) computer science courses. It presents the programming environment used in two of the five initial AP CS Principles pilot courses. Snap! (Build Your Own Blocks) is a free, graphical, drag-and- drop extension to the Scratch programming language. Scratch, designed for 8-14 year olds, models programs as "scripts" without names, arguments, or return values. Snap! supports older learners (14-20) by adding named procedures (thus recursion), procedures as data (thus higher order functions) structured lists, and sprites as first class objects with inheritance. Participants will learn Snap! through discussion, programming exercises, and exploration. See http://snap.berkeley.edu for details. Laptop Required.

29. Circuits and Microcontrollers in Computer Organization Laboratories 301B

Marge Coahran, *Dickinson College*; Janet Davis, *Grinnell College*

This workshop will introduce a set of hands-on laboratory activities appropriate for a first Computer Organization course. Participants will work with real equipment: first implementing elementary digital circuits with TTL logic chips, and then programming AVR microcontrollers in assembly to drive fun accessories such as LEDs and speakers. Participants will not take equipment home afterwards, but will receive parts lists and vendor information. The workshop is intended for educators with little electronics background who are interested in incorporating electronics laboratories into their courses. Laptops (Linux, Mac, or Windows) will provide the programming environment for the AVRs. Free software will be available before the workshop. Participants will work in pairs. Laptop recommended.

30. Web Development with Python and Django 302A

Ariel Ortiz, *Tecnologico de Monterrey, Campus Estado de Mexico*

Many instructors have already discovered the joy of teaching programming using Python. Now it's time to take Python to the next level. This workshop will introduce Django, an open source Python web framework that saves you time and makes web development fun. It's aimed at CS instructors who want to teach how to build elegant web applications with minimal fuss. Django is Python's equivalent to the popular Ruby on Rails framework. Topics that will be covered include: setup and configuration, template language, and database integration through object- relational mapping. Participants should have some familiarity with Python, HTML and SQL. More information at: http://webcem01.cem.itesm.mx :8005/django/ Laptop Required

31. Improving the Accessibility of Computing Enrichment Programs 302B

Richard Ladner, *University of Washington*; Karen Alkoby, *Gallaudet University*; Jeff Bigham, *University of Rochester*; Stephanie Ludi, *Rochester Institute of Technology*; Daniela Marghitu, *Auburn University*; Andreas Stefik, *University of Southern Illinois, Edwardsville*

Many wonderful enrichment programs have been created to introduce young people to computing, but with little attention to making them accessible to students with disabilities. In this workshop participants will learn from practitioners who have introduced computing and programming to young people with disabilities. They will also learn first-hand from students with disabilities about their needs in learning programming. There will be breakout sessions for participants to apply what they have learned to improve existing enrichment programs such as Alice, Arduino, Scratch, Kodu, App Inventor, Greenfoot, Lego Mindstorms, Processing, and Computer Science Unplugged. Laptop Recommended.

32. Enhancing Student Interest by Extending Graphics Applications 302C

Samuel Rebelsky, *Grinnell College*

Computer science teachers strive for new examples and problems to interest millenials. The Media Computing approach has proven successful in attracting students in contexts from community colleges to R1 universities – students are clearly excited by writing programs that make images. In this workshop, we show how to go a step further and have write scripts and plug-ins in Python for open-source graphics programs such as GIMP and Inkscape. Students not only make images, they write filters and features that they can share with others, even non-programmers. E.g., students have written filters that "fractalize" vector graphics or that turn images into stained glass. Further information can be found at http://www.cs.grinnell.edu/~rebelsky/Workshops/SIGCSE2012/
Laptop Required.

33. Engage Your Students by Teaching Programming Using Only Mobile Devices with TouchDevelop 305A

Nikolai Tillmann, Michal Moskal, Jonathan de Halleux and Manuel Fahndrich, *Microsoft Research*; Tao Xie, *North Carolina State University*

The world experiences a technology shift: Powerful and easy-to-use mobile devices like smartphones and tablets are becoming more prevalent than traditional PCs and laptops. We propose to reflect this new reality by adapting how programming is taught. Students should develop software directly on smartphones. In this workshop, we introduce TouchDevelop on Windows Phone 7, a novel application creation environment from Microsoft Research. Its typed, structured programming language is built around the idea of only using a touchscreen as the input device to author code. Easy access to the rich sensor and personal data available on a mobile device results in an engaging programming experience for students who learn programming by creating fun games and applications.
Laptop Optional.

34. CS in Parallel: Modules for Adding Parallel Computing to CS Courses, from CS2 to Theory of Computation 306A

Richard Brown, *St. Olaf College*; Elizabeth Shoop, *Macalester College*

Parallel computing with more and more cores is here to stay. This workshop presents four independent, class-tested, primarily hands-on modules for incrementally adding parallelism in undergraduate CS courses, each requiring 1 to 3 class days and versatile for diverse courses and curricula: parallelizing loops and sharing memory on Intel's Manycore Testing lab (for a second CS course or for computer organization); parallel web crawler in Java or C++ (second CS course); parallel sorting (algorithms); π-calculus theory for communicating sequential processes (theory of computation). Workshop materials provided, drawn from CSinParallel.org. Intended audience: CS instructors. Laptop recommended (Windows, Mac, Linux).

35. Listening to Linked Lists: Using Multimedia to Learn Data Structures 306B

Mark Guzdial and Barbara Ericson, *Georgia Institute of Technology*

Everybody teaches linked lists, with homework like implementing duplicate, weave, and reverse. When those nodes contain strings or numbers, these are pretty boring assignments. When these nodes contain music (MIDI), these operations are composing music, which can then be played. This workshop shows how to use music, images, and sounds to teach the basic data structures, including linked lists, circular linked lists, stacks, queues, and trees. These pieces can then be tied together through the use of simulations to generate animated movies. We will be using Java, though many of the methods can also be used in Python. Laptop Recommended.

36. Puzzle-Based Learning: Introducing Creative Thinking and Problem Solving for Computer Science and Engineering 306C

Raja Sooriamurthi, *Carnegie Mellon University*; Nick Falkner and Zbigniew Michalewicz, *University of Adelaide*

Puzzle-based learning (PBL) is an emerging model of teaching critical thinking and problem solving. Today's market place needs skilled graduates capable of solving real problems of innovation in a changing environment. While solving puzzles is innately fun, companies such as Google and Yahoo also use puzzles to assess the creative problem solving skills of potential employees. In this interactive workshop we will examine a range of puzzles, games, and general problem solving strategies. Participants will emerge with the needed pedagogical foundation to offer a full course on PBL or to include it as part of another course. Currently 40+ institutions around the world are offering courses based on PBL. More details are available at www.PuzzleBasedLearning.edu.au. Laptop Optional.

Pre-Symposium Events

The following events take place Wednesday February 29:

- Teaching Open Source Symposium
- Career Mentoring Workshop
- Open Meeting of the Special Interest Group on Computers and Society (SIGCAS)
- Teaching Ethics in Computer Science: Active Learning
- Computing Accreditation Workshop
- Programming with Alice Workshop (Alice 2.2)
- Programming with Alice Workshop (Alice 3)

More information about these events can be found online at:
`http://www.sigcse.org/sigcse2012/attendees/pre_symposium_events.php`

Upcoming Events

ITiCSE 2012
Technion campus
Haifa, Israel
July 3-5, 2012
http://www.iticse12.org.il/htmls/home.aspx
Conference Chairs
Tami Lapidot – *Technion, Israel*
Judith Gal-Ezer – *The Open University, Israel*

ICER 2012
Auckland, New Zealand
September 10 - 11, 2012
http://icer-conference.org/
General Chair
Alison Clear, *Christchurch Polytechnic Institute of Technology*

SIGCSE 2013
Denver, Colorado, USA
March 6-9, 2013
http://sigcse.org/sigcse2013/
Symposium Co-Chairs
Paul Tymann, *Rochester Institute of Technology*
Tracy Camp, *Colorado School of Mines*