NSF Project Showcase

SIGCSE 2014
March 5-8
Atlanta, Georgia

National Science Foundation

Program at a Glance

Thursday, 10:00 a.m.—11:30 a.m.

Teaching Service-Oriented Programming to CS and SE Undergraduate Students 4
Xumin Liu, Rajendra K. Raj, Thomas Reichlmayr, Alex Pantaleev, and Chunmei Liu

Rethinking Software Testing in Undergraduate Education 5
W. Eric Wong

Problets Anyone? 6
Amruth Kumar

Mobile Computational Thinking with App Inventor 7
Franklyn Turbak, Fred Martin, Mark Sherman, Ralph Morelli, Shaileen Crawford Pokress, and David Wolber

Thursday, 3:00 p.m.—4:30 p.m.

Hands-on Cybersecurity Exercises in the EDURange Framework 8
Richard Weiss, Jens Mache, and Michael Locasto

CS4Alabama: A Model for Statewide Deployment of CS Principles Courses 9
Jeff Gray, Mary Boehm, Carol Crawford, Kathy Haynie, and Sheryl Packman

MyCS: Computer Science for the Middle School Classroom 10
Michael Erlinger

REU Site in Computer Systems 11
Michael Erlinger and Zach Dodds

Friday 10:00 a.m.—11:30 a.m.

Easily-programmable, highly-capable robots for use across the curriculum 12
Zack Butler and Rajendra Raj

Capacity Building through Curriculum and Faculty Development on Mobile Security 13
Li Yang, Joseph Kizza, Kathy Winters, Kai Qian, Prabir Bhattacharya, Fan Wu

Program at a Glance (cont.)

Friday 10:00 a.m.—11:30 a.m. cont.

CE21 Special Project: Principled Assessment of Computational Thinking 14
Eric Snow and Marie Bienkowski

Undergraduate researchers roles in cybersecurity for critical infrastructure and software engineering 15
Joseph Urban

Friday 3:00 p.m.—4:30 p.m.

Research Experience for Teachers: Data Analysis & Mining, Visualization, and Image Processing 16
R. Mitchell Parry & Rahman Tashakkori

Developing a Health Informatics Security and Privacy Program 17
Xiaohong Yuan, Jinsheng Xu, Hong Wang, Kossi Edoh

Cyber Security: Cyber First Responder Program 18
Haydar Sahin

Saturday 10:15 a.m.—11:45 a.m.

Framing a Rigorous Approach to Beauty and Joy for Outreach to Underrepresented Students in Computing at Scale (FRABJOUS) 19
Dan Garcia

CABECT: Collaborating Across Boundaries to Engage Undergraduates in Computational Thinking 20
S. Monisha Pulimood and Kim Pearson

Developing Game-Like Instructional Modules to Enhance Student Learning in Lower Level Core Computer Science Courses 21
Jinghua Zhang, Mustafa Atay, Rebecca Caldwell, Elva J. Jones

Building a serious game to teach secure coding in introductory programming 22
Nicoletta Adamo-Villani

Teaching Service-Oriented Programming to CS and SE Undergraduate Students

Thursday, 10:00 a.m.—11:30 a.m.

Xumin Liu, Rajendra K. Raj, and Thomas Reichlmayr (Rochester Institute of Technology), **Alex Pantaleev** (SUNY Oswego), **Chunmei Liu** (Howard University)

Service-Oriented Programming (SOP) is a relatively new programming paradigm that supports the development of new software applications using existing web services as building blocks. The SOP paradigm has gained significant popularity in industry because it can improve the pace and quality of modern software development via software reuse. The investigators on this multi-institutional project are convinced that SOP must be included into undergraduate computing curricula.

Rethinking Software Testing in Undergraduate Education

Thursday, 10:00 a.m.—11:30 a.m.

W. Eric Wong (University of Texas at Dallas)

Regardless of occupation, software is a fundamental part of every individual's life. Unfortunately, defects are frequently discovered in such software, requiring significant expenditure to maintain and repair. To ensure the quality of the software, testing is the most popular technique used in practice. However, the broad scope of software testing cannot hope to be adequately covered even in a course dedicated to the subject, much less by a more general software engineering course with many competing learning objectives. An important goal is to help students establish the mindset that testing is a critical part of the software development process to be conducted in a rigorous manner, not as an afterthought. To this end, our pedagogical model relies on the following key concepts: many-to-many, minimally intrusive, and non-restrictive. A many-to-many relationship is developed between courses and the instructional modules we created as part of an NSF-sponsored TUES project, where educational materials are selectively applicable to any appropriate courses in a minimally intrusive and non-restrictive way. Rather than briefly covering software testing in a single course, testing skills and topics can be introduced in many different courses, both basic and advanced, at the corresponding skill level of the students. This kind of broad and complete coverage is necessary to ensure that students are prepared to effectively test software as future engineers.

Problets Anyone?

Thursday, 10:00 a.m.—11:30 a.m.

Amruth Kumar (Ramapo College of New Jersey)

Problets are web-based software tutors designed to help students learn programming concepts by solving problems. They present problems, grade the student's answer and provide instant feedback.

Problets present problems such as debugging programs, predicting their output, predicting their state, and evaluating expressions. They are adaptive – they present problems on only the concepts that the student does not already know. The problems are randomized - typically, no two students get the same problem, and no student gets the same problem twice.

Problets provide step-by-step explanation of the correct answer to each problem, a feature unique to problets which has been shown to help improve learning.

Instructors can use problets for closed lab exercises, after-class assignments, and in-class testing. Since problets run in any Java -enabled browser, students can use them 24X7 to learn on their own time, at their own pace, and as often as they please. A typical problet takes 30-40 minutes to complete. Problets are free for educational use.

Problets require no software installation. Adopters direct their students to use problets off a dedicated web site set up for them – they decide which problets to use and when. After their students have used each problet, adopters can request a report, which is provided in Excel format.

Problets are available for expressions (arithmetic, relational, logical, assignment, bit-wise), if-else, switch, while, for, do-while, advanced loop concepts, functions, arrays, classes, recursion and C++ pointers. They are available for C, C++, Java and C#.

Problets have been used continually by numerous 4-year, 2-year and high school instructors since fall 2004. To try out problets, please visit http://www.problets.org. If you are interested in using problets, please email Amruth Kumar, amruth@ramapo.edu

Mobile Computational Thinking with App Inventor

Thursday, 10:00 a.m.—11:30 a.m.

Fred Martin and Mark Sherman (University of Massachusetts Lowell), **Ralph Morelli** (Trinity College), **Shaileen Crawford Pokress** (MIT), **Franklyn Turbak** (Wellesley College), **David Wolber** (University of San Francisco)

Mobile devices are situated computers that enhance our daily lives by integrating computation with location awareness, sensors, social networks, Internet connectivity, and many other features. This situatedness opens up compelling opportunities for computer science education based on engaging young people by empowering them to program these digital Swiss Army Knives.

Since 2009, we have taught introductory undergraduate CS courses based on App Inventor, an environment for creating Android apps. Based on our experiences, we have identified big ideas of mobile computational thinking that augment traditional problem solving and programming concepts. Mobile computing is event-based, leverages device features, emphasizes useful programs embedded in social contexts, takes advantage of a larger informational ecosystem, and involves design, engineering, and entrepreneurship.

Most of our students have no previous programming experience. There are three reasons why App Inventor is a good platform for introducing novices to mobile computational thinking:

- its high-level abstractions for mobile device features promote creating fully functional apps;
- its simple approach to event handling makes it easier to specify app behavior;
- its visual blocks language and cloud-based environment lower barriers to programming.

In our NSF TUES project, we are developing online curricular modules that use App Inventor to teach computational thinking in a mobile context. These modules include web-based tutorials, video lectures, screencasts, programming exercises, and live-coding quizzes. We are also creating and evaluating techniques (including surveys and project rubrics) for assessing students' computational thinking knowledge in the context of our courses and materials.

Hands-on Cybersecurity Exercises in the EDURange Framework

Thursday, 3:00 p.m.—4:30 p.m.

Richard Weiss (Evergreen State College), **Jens Mache** (Lewis & Clark College), **Michael Locasto** (University of Calgary)

This project addresses the need to engage more students at two- and four-year colleges in studying cybersecurity and to provide them with exercises that train and assess them on analytical skills. We provide faculty with interactive exercises that would facilitate adding this topic to the Computer Science curriculum. We are developing and using the EDURange framework for creating these exercises. We have created a reconnaissance exercise for teaching about TCP/IP and how these protocols can be used to map a network structure. We are working on several other exercises related to understanding complex systems; for example, how to recognize anomalous network traffic or understand what the program is doing by looking at the resources it uses.

A fundamental question we are asking is: how to provide a high-level interface for instructors with little experience in cybersecurity that allows them to specify an exercise based on the learning objectives that they want their students to achieve and be assessed on. This interface should be flexible enough that instructors with more experience will be able to specify more variations in the exercises they create. We have developed an intermediate representation and a compiler that generates scripts for Amazon's EC2 and installing and configuring software on the VMs. We have tested our framework several times in classrooms and workshops for students and faculty. One of its advantages is that faculty can use it in their classes with very little set-up. Instructors will be able to access exercises by creating an Amazon account.

CS4Alabama: A Model for Statewide Deployment of CS Principles Courses

Thursday, 3:00 p.m.—4:30 p.m.

Jeff Gray, Mary Boehm, Carol Crawford, Kathy Haynie, and Sheryl Packman (University of Alabama Tuscaloosa)

This presentation describes the first-year experience in developing and evaluating a CS Principles professional development model for training a cohort of teachers across an entire state geography. Over 50 teachers will be trained during a three-year period. The scalable deployment and sustainable persistence of the new CS Principles course adopts the successful practices of a national AP training program developed by the National Math and Science Initiative (NMSI). We apply those practices in a professional development program based on year-long in-person training and distance learning collaboration. A statewide "Teacher Leader" model is being explored where those already teaching more rigorous CS courses assist in training new peer cohorts as they establish CS Principles in their schools. Teachers in these cohorts collaborate together on content and pedagogical learning experiences, fostered by peer leaders. The assessment is uncovering the facets of our model that are most suitable for building a sustainable network of CS Principles teachers. We are concluding the first year of the project, which has introduced CS Principles into our official state curriculum (the CS Principles course was adopted in December 2013 as an official math elective by the Alabama State Department of Education). The presentation will describe the curriculum that is being developed by our teachers, a set of lessons learned from our professional development (PD) experiences, and the details of our evaluation. The lessons learned will be described by relevant literature on PD from other STEM areas.

MyCS: Computer Science for the Middle School Classroom

Thursday, 3:00 p.m.—4:30 p.m.

Michael Erlinger (Harvey Mudd College)

Middle-years Computer Science, or MyCS, is Harvey Mudd College's CS curriculum designed for classrooms ranging from late-elementary to early high school. MyCS seeks to empower K-12 teachers to make creating computation part of as many students' identities as possible. Our goal is to provide engaging, accessible, and easy-to-use content to middle-years teachers, who can implement the content in their classroom using their own pedagogical adaptations of activities and exercises.

Since the project's inception in 2010, the curriculum has sought to reach broader audiences of teachers and their students through regular summer workshops. The curriculum has bdeen prototyped in both Pomona, CA and Lihue, HI school districts, reaching over 2,000 students to date. Our assessments have shown that the MyCS experience reaches young women and men equally effectively, just as it does for students from all of the ethnic and racial subgroups considered. Most recently, we've turned an eye toward reaching a global classroom by developing our curriculum into an online course.

We believe that every middle-years student should be able to confidently say that "CS is something that people like me can do," and that the best way to reach those students is through the instructors who know them well. We will continue to empower teachers of all backgrounds to bring computational thinking to their classroom.

REU Site in Computer Systems

Thursday, 3:00 p.m.—4:30 p.m.

Michael Erlinger and Zach Dodds (Harvey Mudd College)

This Research Experience for Undergraduates continues a summer program at Harvey Mudd College that seeks to capture only the best parts of a graduate-school research experience. Students participate in a variety of projects: developing biological algorithms, designing intelligent music software, and investigating automatic memory management. This presentation will focus on two of 2013's projects, each of which sought to deliver its results to an external audience.

The first project involved the development of CS education exercises that introduce and challenge students' skills in procedural thinking. REU students created a series of 100 carefully graduated mazes that build students' intuition for several Scratch constructs: straight-line code, loops and code factoring, conditionals, color- and sprite-sensing, and the ability to compose those building blocks in order to solve visually-presented tasks. Refined through the summer by deployment in two professional-development workshops, these mazes have become one unit within the MyCS (middle-years computer science) curriculum.

In the second project students investigated the capabilities of inexpensive robot platforms and their potential for creating compelling undergraduate robotics curricula. The result is a new, sub-$300 platform, built atop the Neato vacuum, that provides a foundation for human-scale navigation, localization, and mapping accessible to students at the CS2 level. We present the summer's results along with the robots' current deployment in undergraduate labs at HMC.

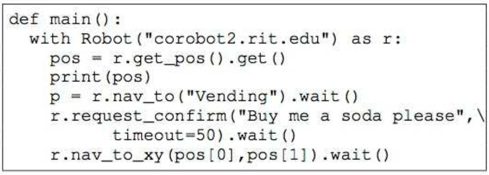
Easily-programmable, highly-capable robots for use across the curriculum

Friday, 10:00 a.m.—11:30 a.m.

Zack Butler and Rajendra Raj (Rochester Institute of Technology)

Robots are typically used at the college level either as a pedagogic platform for introductory programming or for more advanced courses in robotics. With robots becoming cheaper and more plentiful, personal interactions with them will become more commonplace. This project therefore takes the position that undergraduate computing students need the opportunity to explore core computing concepts in a robotics context. Specifically, we will give students the ability to work alongside teams of highly capable and easily programmable corobots, a term used to identify robots that work side by side with humans, rather than being completely autonomous and isolated.

The project has two major components. The first is a robot system that incorporates standard hardware and a fully-developed custom API. This system "solves" the robotics problems so that students not versed in robotics algorithms can still control a robot capable of path planning, navigating the building and avoiding obstacles. The use of a simple API allows robot control to be integrated into any type of programming assignment. The second component of the project is a set of modules to incorporate corobotics into various CS courses such as first-year computing, networking, and data management, thus permitting the students to see these corobots in multiple contexts. The goal is that these assignments will allow students to engage with the material of each course in a domain that is interesting and has a physical and meaningful presence.



```
def main():
  with Robot("corobot2.rit.edu") as r:
    pos = r.get_pos().get()
    print(pos)
    p = r.nav_to("Vending").wait()
    r.request_confirm("Buy me a soda please",\
          timeout=50).wait()
    r.nav_to_xy(pos[0],pos[1]).wait()
```

Capacity Building through Curriculum and Faculty Development on Mobile Security

Friday, 10:00 a.m.—11:30 a.m.

Li Yang, Joseph Kizza, Kathy Winters (University of Tennessee at Chattanooga), **Kai Qian** (Southern Polytechnic State University), **Prabir Bhattacharya** (University of Cincinnati), **Fan Wu** (Tuskegee University)

Mobile security has nowadays become an important topic in security education partly due to the shift in the computing landscape towards mobile devices. The security of mobile devices is vital to the normal functioning in people's lives, and our social, economic and political systems. Mobile security covers a wide area of security including threats and attacks to and defenses of mobile devices and computing protocols including secure coding, cryptography, physical security, secure communication, and policy management. We first compare mobile operating system (OS) security models in terms of traditional access control approaches, application provenance that stamps application with the identity of its author. We then cover encryption that conceals data at rest to address device loss or theft, isolation (Sandboxing), and permission-based access control. Threats and vulnerabilities in mobile applications that include mobile malware, web-based threats, physical threats from lost or stolen devices, and vulnerabilities of mobile applications are also discussed. Mobile malware may exhibit a variety of behaviors as it collects data without a user's knowledge or approval, gathers sensitive or personally identifiable information, or leaves a security hole in the device. Mobile malware includes activity monitoring and data retrieval, system modification, and unauthorized dialing. Web-based and network-based threats take advantages of flaws of web-based applications and networks. They include user interface impersonation, client side attacks, server-side attacks, unauthorized network connectivity, and Wi-Fi sniffing. We have developed extensive teaching materials including lecture materials and hands-on labs related to above topics.

CE21 Special Project: Principled Assessment of Computational Thinking

Friday, 10:00 a.m.—11:30 a.m.

Eric Snow, Marie Bienkowski (SRI International)

SRI International, in the Principled Assessment of Computational Thinking (PACT) project, is developing ways to assess the computational thinking practices—the big ideas that underlie the computer science discipline—that students acquire in the Exploring Computer Science (ECS) curriculum.

SRI is designing and validating the assessments in partnership with the curriculum developers, ECS instructors, assessment experts, and computer scientists. Applying evidence-centered design, we are creating generalized design templates for computational thinking practices. These templates guide the development of assessment tasks for the ECS curriculum, as well as for other CS curricula. The resulting assessments and accompanying scoring rubrics are being piloted and field-tested. SRI will use assessment scores, response processes, and other validity evidence from the field tests to conduct a psychometric evaluation of both unit and summative assessments.

The PACT project represents an important step in filling the computer science pipeline with a more diverse population of students. High-quality assessment tools and resources will lower barriers for adopting and using the ECS curriculum, as well as other CS curricula, and will pave the way to reporting evidence of student progress and readiness to engage in further learning. The PACT project will also contribute to NSF's effort to prepare 10,000 qualified computer science teachers (the CS10K project) by providing both new and experienced CS teachers with accessible and adaptable resources for assessing their students' knowledge and skills in computational thinking.

Undergraduate researchers roles in cyber-security for critical infrastructure & software engineering

Friday, 10:00 a.m.—11:30 a.m.

Joseph Urban (Texas Tech University)

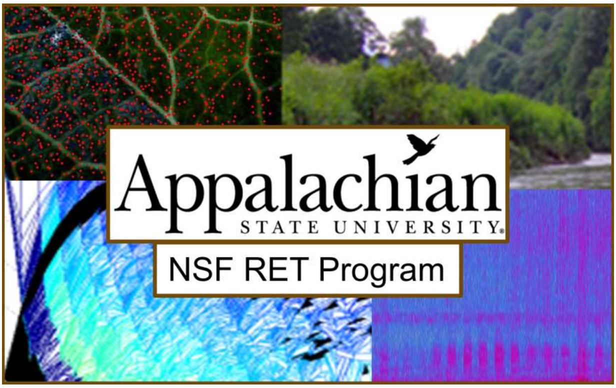
There is a crucial need for professionals capable of addressing cybersecurity for critical infrastructure. The effective development of software systems that have a focus on security is essential. This National Science Foundation Research Experiences for Undergraduates site project provides an opportunity for participants to investigate formal methods in software engineering. This effort is to improve cybersecurity in the development of software systems for critical infrastructure. There is a project for development of a testbed to support classroom exercises for courses in a certificate program in cybersecurity for critical infrastructure under development through an NSF grant. A specification language research effort is in an overall software engineering research program. The Descartes specification language effort is based on a solid foundation of research that has resulted in graduate students who completed degrees and the involvement of four REU site project students and two REU supplement students. This project has a focus on automated software specification generation from expected input and corresponding output that is a direct extension of earlier research on automated test data generation from Descartes specifications. There are extensions to the language for real-time, object-oriented, and intelligent agent software development to support an effort on executable software specifications as formal methods for information assurance. The language and certificate investigations are in the context of grants through the NSF Major Research Instrumentation program and the NSF Federal Cyber Service: Scholarship for Service program in the Capacity Track.

Research Experience for Teachers: Data Analysis & Mining, Visualization, and Image Processing

Friday, 3:00 p.m.—4:30 p.m.

R. Mitchell Parry, Rahman Tashakkori (Appalachian State University)

The NSF supported Research Experiences for Teachers (RET) in Engineering and Computer Science program supports the active involvement of K-12 STEM teachers and community college faculty in engineering and computer science research in order to bring knowledge of engineering, computer science, and technological innovation into their classrooms. The goal of the program is to help build long-term collaborative partnerships between K-12 STEM teachers, community college faculty, and the NSF university research community. Our department received funding to establish a three-year RET site in Data Analysis & Mining, Visualization, and Image Processing. Twelve in-service high school teachers and community college faculty work with faculty mentors and their graduate and undergraduate assistants to conduct research in these fields. During this six-week summer program, participants gain skills that they can utilize to assist their students in solving interdisciplinary problems. In addition, participants design learning modules to teach STEM concepts in their courses. During the school year, teachers bring knowledge of computer science and its application to their classroom exposing their students to computer science. This paper shares some of the activities of this experience.



Developing a Health Informatics Security and Privacy Program

Friday, 3:00 p.m.—4:30 p.m.

Xiaohong Yuan, Jinsheng Xu, Hong Wang, Kossi Edoh (North Carolina A&T State University)

Health informatics is one of the nation's largest growing industries. To protect health information systems, it is extremely important for health informatics professionals to be well educated and trained in information assurance, and to understand the many concerns of security, privacy, integrity and reliability. To meet this demand, we are creating a new, interdisciplinary curriculum model of Bachelor of Science in Computer Science (BSCS) concentration in Health Informatics Security and Privacy (HISP) at North Carolina Agriculture and Technical State University (NC A&T). To establish this BSCS concentration in HISP, we developed a new course on health information systems in the Department of Computer Science, a new course on Mathematics for Health Informatics in the Department of Mathematics, and modified an existing course in the Department of Management to include topics on business practices relating to health information technology. We also developed three course modules on health informatics security and privacy and are integrating these course modules into the existing information assurance courses in the Department of Computer Science.

Currently the BSCS concentration in HISP has been approved by the university. The three new courses and course modules have been taught at least once. We are advertising this program to computer science students, as well as advertising the new courses to students in related disciplines such as mathematics, management, information systems, nursing, biology, etc.

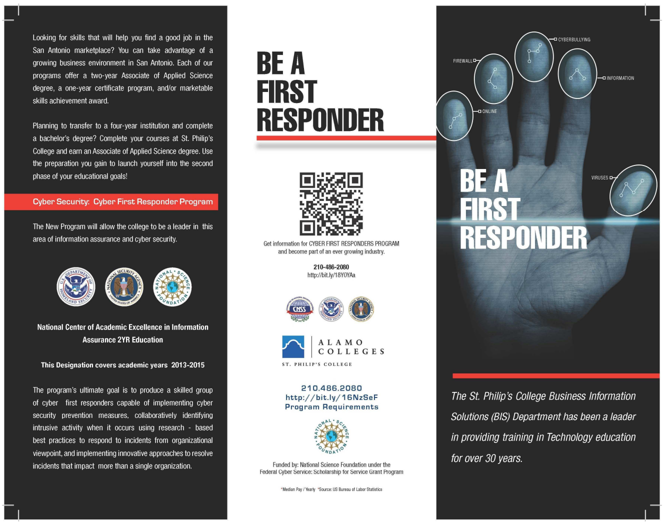
Cyber Security: Cyber First Responder Program

Friday, 3:00 p.m.—4:30 p.m.

Haydar Sahin (St. Philip's College)

The primary intent of the St. Philip's College Cyber Security: Cyber First Responders Program is to develop programs to create the Cyber First Responders that Texas and the nation will need for the fast-growing cyber security workforce. The target population for this project includes Information Technology (IT) Technicians, information security personnel, emergency response personnel, CFR associate degree candidates, college students interested in pursuing a four year degree in CFR, and middle and high school students interested in the CFR career path. St. Philip's will make every effort in its recruitment efforts for the project to include persons from diverse and traditionally underrepresented groups including females, racial minorities, low socioeconomic status, persons with disabilities, and veterans.

The project goals will focus on addressing the identified needs and creating the curriculum and courses that will comprise the larger Cyber First Re-



Looking for skills that will help you find a good job in the San Antonio marketplace? You can take advantage of a growing business environment in San Antonio. Each of our programs offer a two-year Associate of Applied Science degree, a one-year certificate program, and/or marketable skills achievement award.

Planning to transfer to a four-year institution and complete a bachelor's degree? Complete your courses at St. Philip's College and earn an Associate of Applied Science degree. Use the preparation you gain to launch yourself into the second phase of your educational goals!

Cyber Security: Cyber First Responder Program

The New Program will allow the college to be a leader in this area of information assurance and cyber security.

National Center of Academic Excellence in Information Assurance 2YR Education

This Designation covers academic years 2013-2015

The program's ultimate goal is to produce a skilled group of cyber first responders capable of implementing cyber security prevention measures, collaboratively identifying intrusive activity when it occurs using research - based best practices to respond to incidents from organizational viewpoint, and implementing innovative approaches to resolve incidents that impact more than a single organization.

Get information for CYBER FIRST RESPONDERS PROGRAM and become part of an ever growing industry.

210-486-2080
http://bit.ly/18YOYAa

ALAMO COLLEGES
ST. PHILIP'S COLLEGE

210.486.2080
http://bit.ly/16NzSeF
Program Requirements

Funded by: National Science Foundation under the Federal Cyber Service: Scholarship for Service Grant Program

*Median Pay / Yearly *Source: US Bureau of Labor Statistics

The St. Philip's College Business Information Solutions (BIS) Department has been a leader in providing training in Technology education for over 30 years.

Framing a Rigorous Approach to Beauty and Joy for Outreach to Underrepresented Students in Computing at Scale (FRABJOUS)

Saturday, 10:15 a.m.—11:45 a.m.

Dan Garcia (University of California at Berkeley)

The Beauty and Joy of Computing (BJC) is an introductory computer science curriculum developed at UC Berkeley and further refined at UNC Charlotte. It is an intended for non-CS majors at the high school junior through undergraduate level. Together, they formed two of the five initial pilots for the AP CS Principles course being developed by the College Board and the National Science Foundation.

Our BJC course is different from other AP CS:Principles in that we take a more programming-centric approach to the course. We believe the real transformative and empowering experience comes when one learns how to program the computer, to translate ideas into code. Our course teaches students how to do exactly that, using Snap! (based on Scratch), one of the friendliest programming languages ever invented. It's purely graphical, which means programming involves simply dragging blocks around, and building bigger blocks out of smaller blocks. But this course is far more than just learning to program, we talk about the social implications of computing, and the unintended consequences of computing innovations. Throughout the course, relevance is emphasized: relevance to the student and to society.

After the initial pilot, the team embarked on the FRABJOUS CS project (Framing a Rigorous Approach to Beauty and Joy for Outreach to Underrepresented Students in Computing at Scale), whose goal was to prepare 100 teachers across the country to teach our course in their high schools. We are actively recruiting high school teachers to join our PD this year! Sign up at http://bjc.berkeley.edu

CABECT: Collaborating Across Boundaries to Engage Undergraduates in Computational

Saturday, 10:15 a.m.—11:45 a.m.

S. Monisha Pulimood, Kim Pearson (The College of New Jersey)

Our study, “Collaborating Across Boundaries to Engage Undergraduates in Computational Thinking (CABECT)” tests the hypothesis that real-world collaboration between students in computer science, students in complementary disciplines, and a community partner is an effective strategy for infusing computational thinking and community engagement across the curriculum. As proof of concept, we have been piloting a suite of courses in which students in computer science, journalism and interactive multimedia classes have joined forces to design and develop a software system that collects and presents neighborhood-level environmental data. This information is important to community organizations such as Habitat for Humanity in making decisions about acquiring land for affordable housing. We will present a progress report on the implementation of our curricular model as we enter our third semester.

Our approach is deeply rooted in the considerable research base on pedagogy in computer science, the liberal arts and journalism, as well as how students learn. The project is demonstrating how to adapt computer science and non-computer science courses, and is developing a assessment instruments to measure outcomes for participants. The model is replicable for a wide range of computing-dependent disciplines and local community partners. Through multi-disciplinary, collaborative and immersion experiences, undergraduate students better internalize how to take responsibility for engaged citizenship in a complex and diverse society, while advancing their critical, analytic and computational thinking.

Developing Game-Like Instructional Modules to Enhance Student Learning in Lower Level Core Computer Science Courses

Saturday, 10:15 a.m.—11:45 a.m.

Jinghua Zhang, Mustafa Atay, Rebecca Caldwell, Elva J. Jones
(Winston-Salem State University)

The objective of this project is to develop game-like instructional modules to enhance student learning in lower level core Computer Science (CS) courses, namely Programming I, Programming II and Data Structures. Struggling with these three courses is the main barrier preventing students from declaring the CS major and retaining students in the program. Statistics have shown that fewer African Americans are pursuing CS degrees relative to their proportion of the overall population and among those who do major in CS very few are employed as programmers. Many minority students have fears of programming, which is one of the major contributing factors for the above troubling situation. This project will help Winston-Salem State University, an HBCU, to enhance CS education, attract new minority and female students to pursue the CS major, and increase the pool of qualified minority graduates in this field.

To date, five game modules haven been developed and they have been used several times in classrooms. These modules are designed to help students learn difficult concepts in a gaming context. Feedback from students is positive and the evaluation results are very promising. The modules have been refined based on the feedback from faculty, students and the advisory board. None of these modules require prior experience in gaming.

Please stop by the NSF showcase for a demonstration of the game modules.

Building a serious game to teach secure coding in introductory programming

Saturday, 10:15 a.m.—11:45 a.m.

Nicoletta Adamo-Villani (Purdue University), **Stephen Cooper** (Stanford University), **David Whittinghill** (Purdue University)

The purpose of this project is (1) to design and build a serious game prototype for use in helping to develop secure coding abilities among novice programmers, and (2) to develop high-quality introductory computing laboratory exercises that incorporate game activities as part of the laboratory assignments. We have chosen a serious game approach both because the playing of a game allows students to explore topics more sophisticated than they would normally be able to program from scratch as part of an introductory programming class, as well as from the motivational experience – many students enjoy playing video games. The project addresses the need to introduce Information Assurance (IA) education early in the CS curriculum. Security needs to be considered as a design criterion for software development and its introduction cannot be delayed until the students are upperclassmen when they are taught a class in secure coding. Rather than introduce secure coding as a stand-alone topic into an already over-crowded introductory computing class, the objective of the project is to integrate it into the laboratory exercises within the course, treating security as a context within which students learn the traditional programming and problem solving components.

Meet with Program Officers

This year, NSF Program Officers will be available during most presentation sessions in the meeting area outside the NSF Showcase booth. A specific schedule of times and Program Officers is available at the NSF Showcase booth.

While there will be some open time to meet with Program Officers, some will work by appointment.

Please visit http://www.cs.virginia.edu/~sherriff/nsfshowcase to sign up for an appointment time to meet with a Program Officer or come by the NSF Showcase booth.

For more information, please visit:

http://www.cs.virginia.edu/~sherriff/nsfshowcase

Cover Image:
Atlanta Skyline
Credit: Wikimedia Commons