



in cooperation with

The Association for Computing Machinery
Special Interest Group on Computer Science Education

Presents

CCSCNE-2007
THE THIRTEENTH ANNUAL
CONSORTIUM FOR COMPUTING
SCIENCES IN COLLEGES
NORTHEASTERN CONFERENCE
Rochester Institute of technology
April 20 - 21, 2007
Rochester, NY

Papers, Panels, Tutorials,
Pre-Conference Workshops,
Programming Contest, Student Posters

Web Site
<http://www.ccsne.org/2007/>

Statement of Purpose

The CCSCNE brings together faculty, staff, and students from academic institutions throughout the Northeast for exchange of ideas and information concerning undergraduate computing curricula.

Schedule starts on page 3

Descriptions of plenary sessions, workshops, and tutorials start on page 7.

Accommodations and direction information is on page 14 and is also available at <http://www.cs.rit.edu/~ccsne/>.

CCSCNE-2007 Program Committee

Conference Chairs

Paul Tymann, RIT

Keith Whittington, RIT

Papers Chairs

Ali Erkan, Ithaca College

James Heliotis, RIT

Jim Teresco, Williams College

Undergraduate Posters Chairs

Rameen Mohammadi, SUNY Oswego

Evelyn Stiller, Plymouth State University

Panels/Tutorials/Workshops Chairs

Tim Fossum, SUNY Potsdam

Lonnie Fairchild, SUNY-Plattsburgh

Yana Kortsarts, Widener University

Speakers Chair

Ingrid Russell, University of Hartford

Publicity Chairs

Frank Ford, Providence College

Nadia Bolaek, RIT

Vendors Chairs

Hemant Pendharkar, Worcester State College

Adrian Ionescu, Wagner College

Programming Contest Chairs

Frank Ford, Providence College

Zack Butler, RIT

James Kwon, RIT

Registration Chair

Roxanne Canosa, RIT

Local Arrangements Chair

Wendy Benjamin, RIT

Birds of A Feather Coordinators

Stephen Bloch, Adelphi University

Ziya Arnavut, SUNY Fredonia

Friday, April 20, 2007

Registration for Workshop Participants 8:00 a.m. – 9:00 a.m.

Registration for Programming Contest Participants/Coaches 8:00 a.m. – 8:30 a.m.

These early registrations will be in the atrium of B. Thomas Golisano College of Computing and Information Sciences (GCCIS)

General Registration..... 10:00 a.m – 5:00 p.m.

General registration will be in the Student Alumni Union (SAU)

Programming Contest 8:00 a.m. – 12:45 p.m.

Breakfast and registration of teams and team members8:00 a.m. – 8:40 a.m.
GCCIS Atrium

Computers Available for Teams to Practice 8:00 a.m.-8:40 a.m.
GCCIS

Initial Meeting..... 8:40 a.m.-8:55 a.m.
GCCIS

Luncheon for Teams Noon-12:45 p.m.
GCCIS

Pre-conference Workshops 9:00 a.m. – noon

A Hands-on Workshop on Computer Security Tools

Alan Kaminsky, Rochester Institute of Technology
(see description after program)
GCCIS

Assessment Techniques for Curricular Improvement

Roxanne L. Canosa, Rochester Institute of Technology
Rajendra K. Raj, Rochester Institute of Technology
(see description after program; participants are encouraged to bring laptops)
GCCIS

Python as a First Language

Allen B. Downey, Olin College of Engineering
(see description after program)
GCCIS

Vendor Displays10:00 a.m.-6:00 p.m.

SAU

Welcome..... 1:00 p.m.-1:15 p.m.

Jorge Diaz-Herrera, Dean of the B. Thomas Golisano College of Computing and Information Sciences, Rochester Institute of Technology
Paul Tymann, CCSCNE-2007 Co-Chair, Rochester Institute of Technology
SAU

Plenary Session I 1:15 p.m.-2:15 p.m.

TBA

Andries van Dam, Brown University
SAU

(see description after program)

Break 2:15 p.m.-2:45 p.m.

SAU

Concurrent Session 1 2:45 p.m. -4:00 p.m.

Session 1A. Tutorial: Diversity in the Computing Sciences

SAU

Goran Trajkovski, Towson University
(see description after program)

Session 1B. Papers: Assessment

SAU

Leveraging Learning Styles to Improve Student Learning - The Interactive Learning Model and Learning Combination Inventory

Diane Hamilton, Rowan University
Darren Nicholson, Rowan University
Daniel McFarland, Rowan University

Preliminary Results of a Longitudinal Study of Computer Science Students Trends, Behaviors and Preferences

Trudy Howles, Rochester Institute of Technology

Session 1C. Papers: Curricular Issues

SAU

On Startups and Teaching Computer in Four-Year Colleges

Dominique Thiebaut, Smith College

Curricular Response to the Real Time Data and VoIP Tidal Wave

Bruce Hartpence, Rochester Institute of Technology

The N-Body Problem Throughout the Computer Science Curriculum

Lubomir Ivanov, Iona College

Break 4:00 p.m.-4:45 p.m.

SAU

Concurrent Session 2 4:45 p.m.-6:00 p.m.

Session 2A. Panel: Broken or Not? Fixing Undergraduate Computing Education in a Multi-Disciplinary World

SAU

Rajendra Raj, Rochester Institute of technology
Jennifer Kay, Rowan University
Chris Okasaki, United States Military Academy
Herbert Bernstein, Dowling College
Jeffrey Forbes, Duke University

Session 2B. Papers: Introductory Computer Science

SAU

Problem Stereotypes and Solution Frameworks: a Design-First Approach for the Introductory Computer Science Sequence

T.M. Rao, SUNY Brockport
Sandeep Mitra, SUNY Brockport
Roxanne Canosa, Rochester Institute of Technology
Sidney Marshall, Rochester Institute of Technology
Thomas Bullinger, ArchSynergy, Ltd.
Virginia Teller, Hunter College CUNY

RobotStudio: A Universal IDE for Teaching Undergraduate Computer System Courses
Li Xu, UMass-Lowell

Session 2C. Papers: Bioinformatics and Computational Biology

SAU

Developing a Truly Interdisciplinary Bioinformatics Track: Work in Progress

Elizabeth Goode, Towson University

Goran Trajkovski, Towson University

A Multidisciplinary Course in Computational Biology

Brian Tjaden, Wellesley College

Student Posters/Social Hour 6:00 p.m.-7:00 p.m.

SAU

The list of posters and presenters will be distributed at registration

Banquet 7:00 p.m.-9:00 p.m.

SAU

The Programming Contest awards and the Student Poster Session awards
will be announced at the banquet

Saturday, April 21, 2007

Continental Breakfast..... 7:30 a.m.-8:30 a.m.

SAU

Registration8:00 a.m.-10:00 a.m.

SAU

Vendor Displays8:00 a.m.-10:30 a.m.

SAU

Concurrent Session 3 8:30 a.m.-9:45 a.m.

**Session 3A. Tutorial: Teach Scheme, Reach Java: Introducing Object-Oriented
Programming without Drowning in Syntax**

SAU

Stephen Bloch, Adelphi University

Viera Proulx, Northeastern University

(see description after program)

Session 3B. Panel: Undergraduate Research in the Curriculum

SAU

Amruth Kumar, Ramapo College of New Jersey

Brian Ladd, SUNY Potsdam

Ursala Wolz, The College of New Jersey

Ellen Walker, Hiram College

William Joel, Western Connecticut State University

Session 3C. Papers: Projects

SAU

A Software Engineering Project That Looks Like the Real World

Andrew Pletch, State University of New York at New Paltz

Aram Agajanian, State University of New York at New Paltz

Using Formal Software Development Methodologies in a Real-World Student Project: An Experience Report

Sandeep Mitra, SUNY Brockport

Thomas Bullinger, ArchSynergy, Ltd.

Project the Wiki Way: Using Wiki for Computer Science Course Project Management

Li Xu, UMass Lowell

Session 3D. Papers: Collaborative Learning

SAU

ITEAM Integrated Teamwork Enablement and Management in Online Campus

Zhixiong Chen, Mercy College

Delia Marx, Mercy College

Collaborative Programming Projects in an Advanced CS Course

Tamar Benaya, The Open University of Israel

Ela Zur, The Open University of Israel

Plenary Session II.....10:00 a.m.-11:00 a.m.

SAU

Impacts of Moore’s Law: What Every CIS Undergraduate Should Know About the Impacts of Advancing Technology

Mary Jane Irwin, Evan Pugh Professor, Penn State University

Break11:00 a.m.-11:30 a.m.

SAU

Concurrent Session 411:30 a.m.-12:45 p.m.

Session 4A. Panel: The Role of Large Scale Computing in Computer Science Education

SAU

Suk-Chung Yoon, Widener University

Jeff Saltz, JP Morgan Chase

Session 4B. Tutorial: Reading, Writing, and Revising with Wiki Technology

SAU

Cliff Kussmaul, Muhlenberg College

Sharon Albert, Muhlenberg College

Drexel University

(see description after program)

Session 4C. Papers: Simulation and Visualization Tools

SAU

Simulation of a Simple CPU Design and Its Use as an Instructional Tool in a Computer Organization Course

John Phillips, Mansfield University of Pennsylvania

An Animated Learning Tool for Kerberos Authentication Architecture

Xiaohong Yuan, North Carolina AT State Univ.

A Visual Tool for Teaching Multithreading in Java

Yaodong Bi, University of Scranton

John Beidler, University of Scranton

Session 4D. Papers: Courses

SAU

A Simple Framework for Distributed Component-Based Systems

Harold Smith, Penn State New Kensington

Effectiveness of a Language Implementation Project in Building Appreciation for Formal Specification

Doug Baldwin, SUNY Geneseo

Integrating a Real-World Scheduling Problem into the Basic Algorithms Course

Yana Kortsarts, Widener University

Lunch 12:45 p.m.-2:00 p.m.

SAU

Winner of the best paper award will be announced at the luncheon.

Membership Meeting..... 2:00 p.m.-2:30 p.m.

SAU

Board Meeting..... 2:30 p.m.-4:00 p.m.

SAU

Workshop Descriptions and Biographies

Parallel Programming in Java

Alan Kaminsky, Rochester Institute of Technology

Abstract: With the ever-growing demand for solving larger computational problems in all areas of science and engineering, and with the advent of multicore-CPU desktop computers which require parallel programming techniques to obtain full performance from the hardware, it is becoming imperative to teach parallel programming to all computer science students. Since more and more students are learning Java as their principal programming language, it is becoming imperative to teach parallel programming in Java to all computer science students. However, most parallel

programming textbooks are written using the non-object-oriented OpenMP or MPI libraries and using the C or Fortran languages, not Java. Parallel Java (PJ), a library for parallel programming in 100% Java, was developed in response to these trends. Including capabilities patterned after the OpenMP and MPI standards, PJ is a unified API for writing object oriented parallel programs that run on shared memory multiprocessor (SMP) machines such as multicore PCs, on cluster parallel computers, and on hybrid SMP cluster parallel computers. In this workshop you'll get an introduction to parallel computing, get an introduction to PJ, write small parallel programs using PJ, run your programs on shared memory and cluster parallel computers, and measure your programs' performance. You'll also receive an advance draft copy of a new textbook the presenter is writing on parallel programming in Java.

Biography: In his 30 years of industrial and academic computing experience, Alan Kaminsky has developed telephone switching system software at Bell Laboratories, developed real-time embedded control software and fuzzy genetic algorithms at Harris Corporation, taught graduate software engineering as an Assistant Professor at the Rochester Institute of Technology, and worked on printer system architectures at Xerox Corporation. While at Xerox, Alan got involved with Sun Microsystems' Jini Network Technology, led the Jini Printing Working Group industry consortium that defined a draft specification for the Jini Print Service, and was part of the expert group that developed the Java Print Service API released as package javax.print in the standard Java platform. Alan was also one of the original members of the Jini Community Technical Oversight Committee. Now an Associate Professor in the Department of Computer Science at the Rochester Institute of Technology, Alan teaches and conducts research in parallel programming, distributed systems, cryptography, security, small mobile wireless devices, wireless networking, and ad hoc networking. Alan invented Parallel Java (PJ), an API and middleware for parallel programming in 100% Java on shared memory multiprocessor (SMP) parallel computers, cluster parallel computers, and hybrid SMP cluster parallel computers. Alan also invented the Tuple Board, a new paradigm and middleware for distributed collaborative applications running on ad hoc networks of mobile wireless computing devices. Alan has a B.S. in Electrical Engineering from Lehigh University and an M.S. in Computer Engineering from the University of Michigan.

Assessment Techniques for Curricular Improvement

Roxanne L. Canosa, Rochester Institute of Technology

Rajendra K. Raj, Rochester Institute of Technology

(participants are encouraged to bring laptops)

Abstract: Professional and regional accreditation bodies increasingly require programs to maintain a three-part assess/evaluate/improve process. This three-part process must be clearly documented, regularly used, and should be inclusive of all faculty. Adhering to this process should enable a program to determine the extent to which students are achieving state educational objectives and outcomes. In recent years program assessment has become more rigorous; it is no longer sufficient to collect data and evaluate outcomes once every accreditation cycle. Instead, assessment is now viewed as an ongoing activity that will serve to effect change

and should lead to continuous improvement of the curriculum. Program evaluation should also include other key constituents such as alumni, students, and employers. Outcomes-based assessment presents a new set of challenges for departments to grapple with. It is sometimes difficult to identify the types of data to collect, and once collected, how to use the data to evaluate individual courses and the overall program. Moreover, it is challenging to interpret the data to identify specific strengths and weaknesses within the program.

The workshop will begin with an overview of the current state of outcomes-based assessment, and will continue with an explanation of the terminology currently in use. A survey of effective data collection techniques will be presented, including direct and indirect assessment techniques. The presenters' own assessment process will be presented and critiqued, and participants will have an opportunity to apply similar techniques for their own assessment and accreditation requirements. The goal of the workshop is to help participants understand how to collect, interpret, and use assessment data for the accreditation process to improve the quality of their courses and the overall degree program. Participants may bring samples of their own course syllabi and program objectives to make the workshop more relevant to their own needs.

Biographies:

Roxanne L. Canosa is an Assistant Professor of Computer Science at the Rochester Institute of Technology. She teaches introductory computer science courses, as well as advanced courses in algorithm analysis, artificial intelligence, operating systems, and professional communications. She has been involved in the assessment and accreditation process for the past three years, and is currently the departmental assessment coordinator at RIT. Dr. Canosa is also actively involved in the department-level and college-level curriculum committees. She previously worked at Eastman Kodak Company, and earned her Ph.D. in Imaging Science in 2003.

Rajendra K. Raj is an Associate Professor of Computer Science at the Rochester Institute of Technology. He has taught introductory and advanced computer science courses at the undergraduate and graduate levels in programming languages, security, and database systems. He has been actively involved with curricular and assessment issues, has conducted external program reviews, and currently advises the departmental assessment committee at RIT. Dr. Raj was previously a Vice President in information technology at Morgan Stanley & Co., and an Assistant Professor at SUNY Oswego. He earned his Ph.D. in Computer Science from the University of Washington, Seattle.

Python as a First Language

Allen B. Downey, Olin College of Engineering

Abstract: Python is one of the best languages for beginning programmers. It is syntactically and semantically clean, which spares students from the distractions that plague other languages. It provides lists and dictionaries (hashtables) as built-in types, so students can focus more on algorithms and less on the implementation of data structures. It is dynamically typed, which allows students to explore alternative designs more easily (with the drawback of providing less compile-time error checking). It is also free and well-documented.

Python's implementation of object-oriented features lends itself to either an "object-first" or "object-later" approach, allowing students to get from basic programming concepts to data structures and object-oriented design quickly and (relatively) painlessly. In addition, Python has standard modules for graphics, GUIs, network programming, database programming, and more. In one semester, students who have never

programmed before can take on engaging projects like networked games and web applications.

This workshop is for faculty who are considering using Python in an introductory programming class. I will provide a brief introduction to the language, and participants will have the opportunity to work on exercises similar to what might be used in an intro class.

I will also introduce Swampy, a suite of programs I use in my classes. Swampy includes TurtleWorld, a Logo-like graphics environment implemented in Python, and Lumpy, a program that generates UML diagrams of a running program. Information about Swampy is available from allendowney.com/swampy.

The Python version of the book is available in electronic form at thinkpython.com.

Participants in this workshop will receive a bound hardcopy.

Biography: Allen Downey got his Ph.D. in Computer Science from Berkeley in 1997, and has taught introductory programming (among other things) at U.C. Berkeley, Colby College, Wellesley College, and Olin College. He is the author of "How To Think Like a Computer Scientist," a free, open-source textbook available in Python, Java and C++.

Plenary Sessions Abstracts and Biographies

Van Dam

Biography: Andries van Dam (Andy) has been on Brown's faculty since 1965, and was one of the Computer Science Department's co-founders and its first Chairman, from 1979 to 1985. He was a Principal Investigator and was the Director from 1996-1998, in the NSF Science and Technology Center for Graphics and Visualization, a research consortium including Brown, Caltech, Cornell, North Carolina (Chapel Hill), and the University of Utah. Professor van Dam received the B.S. degree with Honors in Engineering Sciences from Swarthmore College in 1960 and the M.S. and Ph.D. from the University of Pennsylvania in 1963 and 1966, respectively

Impacts of Moore's Law: What Every CIS Undergraduate Should Know About the Impacts of Advancing Technology

Mary Jane Irwin, Evan Pugh Professor, Penn State University

Abstract: Moore's Law postulated by Gordon Moore, then CEO of Intel, in 1965 – that the number of transistors that can be integrated on a single integrated circuit will double every 24 months – has been projected to hold for at least the next decade. Motivated by the semiconductor industry's technology roadmap, this talk will present the impacts of advancing technology that will determine what our students will need to know in order to use these systems of the future most effectively. For example, in order to contain the energy costs yet stay on the performance curve, manufacturers are moving to multi-core systems rather than continuing to increase single-core clock rates. Thus, multi-threaded, multi-core applications programming will be the norm, not the exception. Power will be the most important computing resource: battery power for mobile systems, heat dissipation for large systems, and power generation/delivery for large server/compute

farms. Systems will become less reliable, not more, due to inevitable increases in both transient and permanent faults in the underlying hardware. The usual computing resources (time, space, communication) and the emerging computing resources (power, resiliency, security) will have to be cooperatively managed by the user, by the programmer, and by the run-time system.

Biography: Dr. Irwin has been on the faculty at Penn State since 1977 where she currently holds the title of Evan Pugh Professor and A. Robert Noll Chair in Engineering in the Department of Computer Science and Engineering. Her research and teaching interests include computer architecture, embedded and mobile computing systems design, power and reliability aware design, and emerging technologies in computing systems. Dr. Irwin received her Ph.D. from the University of Illinois. She received an Honorary Doctorate from Chalmers University, Sweden, in 1997, was named a Fellow of The Institute of Electrical and Electronic Engineers (IEEE) in 1995, a Fellow of The Association for Computing Machinery (ACM) in 1996, and was elected to the National Academy of Engineering in 2003. Dr. Irwin is currently serving as Co-chair of ACM's Publications Board and as a member of the CRA-W Steering Committee. In the past she has served as a founding co-Editor-in-Chief of ACM's Journal on Emerging Technologies in Computing Systems and as Editor-in-Chief of ACM's Transactions on the Design Automation of Electronic Systems, as an elected member of the CRA's (Computing Research Association) Board of Directors, of the IEEE Computer Society's Board of Governors, of ACM's Council, and as Vice President of ACM. Dr. Irwin's complete curriculum vitae is available online via [vitae.pdf](#). More details on her research (including publications) are available at the Microsystems Design Lab (mdl) group's webpage. She and her husband celebrated their 40th wedding anniversary in 2006 and (finally!) became grandparents on March 19, 2002.

Tutorials Descriptions

Tutorial: Teach Scheme, Reach Java: Introducing Objecy-Oriented Programming without Drowning in Syntax

Stephen Bloch, Adelphi University

Viera Proulx, Northeastern University

Abstract: A first programming course should be about not this week's "hot" language, but fundamental programming concepts and practices. Yet beginning students often mistake language as the subject. How can we teach students to develop real programs without overemphasizing language?

We start in a simple, consistent language: a pedagogical subset of Scheme. The IDE enforces this subset and gives level-appropriate error messages, but allows students to outgrow each successive subset. Students learn fundamental programming concepts in this sheltered environment before encountering them in the more bewildering world of Java or C++.

Students learn a step-by-step **design recipe** for software development: a series of concrete questions, with concrete products at each stage. Test cases are developed early, as part of requirements specification. This design recipe provides not only a framework for Socratic teaching, but also a grading rubric showing that that *every* step matters, not only code.

We emphasize data types as both concept and tool: to every data type we develop correspond a *coding pattern* and a *testing pattern*. For example, when we introduce self-referential data types, the already-learned patterns for polymorphism and fields automatically yield a safe form of recursion.

For non-majors, we stop with Scheme. For CS majors, the second course is in Java; it builds on the programming concepts and methodology learned in Scheme, explicitly discussing how to apply the same concepts and design recipe. The resulting student, after a year, can approach programming problems with care (not “hack until it works”). This tutorial will demonstrate the design recipe on classroom problems, and how to teach the same concepts in Scheme and Java. You’re all invited to a week-long summer workshop, covering in more depth the technical, classroom, and curriculum issues.

Biographies:

Stephen Bloch (Adelphi University) has taught beginning programming for twelve years, including both CS1/CS2 for majors and a first-and- last programming course for non-majors; in the last eight years, he’s applied this approach variously in Scheme, Java, and C++. He’s led several week-long summer workshops to teach high school faculty the approach, and has NSF funding to lead similar workshops in 2007-2009 for college faculty.

Viera K. Proulx (Northeastern University) has a long-standing interest in computer science education. Since 2001 she’s been developing ways to teach class-based OOP with the aid of the data- type-driven Design Recipe. She’s presented her work at numerous workshops and conferences. As a Co-PI on the aforementioned NSF grant, she will also lead workshops for college faculty in 2007-2009.

Tutorial: Diversity in the Computing Sciences

Goran Trajkovski, Towson University

Abstract: The Computer Sciences have been traditionally considered to be a “diversity-unfriendly” ground. It is true that topics of national origin, culture, ethnicity, gender etc. cannot be incorporated in separate courses/units in the amount in which they are in the humanities, for example. However, despite the general attitude, infusion of such topics happens in our classrooms on daily basis. It is that many of us just do not label it as such. Via small interventions, reworking of the examples, projects, assignments etc., we can and we are infusing diversity topics in our daily teaching activities.

The goal of this workshop is to raise awareness in the community on the importance of the topic, overview methodologies of infusion of diversity in the curriculum on a micro and macro level. With the introduction of *My First Diversity Workbook*, a tool the tutorial facilitator uses for purposes of training faculty, the participants will enter the subject discussed hands-on. By creating a collaborative online community, the discussions

started at this workshop can continue and grow, thus providing the participants a forum for exchange of ideas and experiences from their own classrooms.

Biography: Dr. Goran Trajkovski is an assistant professor of computer and information sciences at Towson University, USA. Throughout his 12 years in higher education, he has been developing modules and provided training in diversity infusion in the curriculum for faculty in the “diversity-unfriendly” disciplines of computer science, IT, natural sciences, and mathematics. He has co-chaired the 10th and is co-chairing the 13th Towson University multicultural conference (2004 and 2007 respectively). He is the editor of the book *Diversity in Information Technology Education: Issues and Controversies* (InfoSys, 2006).

Tutorial: Reading, Writing, and Revising with Wiki Technology
Cliff Kussmaul, Muhlenberg College

Abstract: Appropriate use of wikis in teaching can enhance development of reading, writing, and critical thinking skills. The best known wiki is probably Wikipedia, a free-content encyclopedia with over five million articles in over 250 languages, but wiki technology presents possibilities well beyond its Wikipedian use. A wiki allows users very easily to collaborate to create and edit documents, and links between documents -- effectively making a website -- without needing to acquire a new set of skills. Wikis also enable users to track changes to documents, viewing any previous version of a page to see how it evolved and who made which changes. This support for collaboration and revision tracking encourages innovations that extend and expand upon common teaching methods.

Our tutorial gives participants opportunities to work with wiki-based exercises involving close reading and peer review. It alternates between presentation of key ideas and techniques, hands-on activities, and discussion. We also present survey results that assess the exercises' effectiveness of exercises we have already implement for use in classes. This tutorial is primarily targeted at faculty who want to learn about wikis and ways to use wikis in their courses to support reading, writing, and revising. It should also be of value to students interested in these topics.

Between them, the facilitators have experience using wikis in first year seminars, for team project courses, and to support collaborations across and between institutions. Development and presentation of this tutorial is supported by a Hewlett Packard Technology for Teaching grant.

Biographies:

Clifton Kussmaul is Assistant Professor of Computer Science at Muhlenberg College, and Chief Technology Officer for Elegance Technologies, Inc. which develops software products and provides software product development services to clients. Clif has a PhD from the University of California, Davis, an MS and MA from Dartmouth College, and a BS and BA from Swarthmore College. His interests include agile development, virtual teams, entrepreneurship education, and cognitive neuroscience, particularly auditory processing.

Sharon L. Albert is a Visiting Lecturer in the Religion Studies Department at Muhlenberg College. She actively experiments with and implements into her teaching a variety of instructional technologies, and has worked as an Instructional Technologies Specialist at the University of Chicago where she is completing a Ph.D. in Religion. Sharon has an M.Phil. from University College Dublin and a B.A. from McGill University.

Accommodations and Directions

Radisson Airport Hotel

A block of rooms has been reserved at Radisson Airport Hotel located adjacent to campus. Their address, telephone, and web address area below.

Address: 175 Jefferson Road , Rochester New York 14623
Reservations: (888) 201-1718 US
Telephone: (585) 475-1910 Fax: (585) 475-9633
Web site: http://www.radisson.com/rochesterny_airport

When making your reservation, mention the code RITCCSCNE. Rates at the Radisson are \$89 for a single and \$99 for a double.

Other hotels in the Rochester area are listed at the RIT site:
<http://www.rit.edu/%7E960www/contact/areahotels.php3>

Directions to Rochester Institute of Technology

Directions and campus maps are available at <http://inside.rit.edu/maps/>.

A quick summary of directions from that site is:

From the Airport: Turn right onto Brooks Avenue, then a quick right onto I-390 South. From 390, take the next exit (Scottsville Road) and turn right at the end of the ramp. Drive for approximately three miles, then turn left onto Jefferson Road (Route 252). Proceed east a short distance to campus, RIT's main entrance will be on your right.

From the NYS Thruway: Take exit 46 and proceed north on I-390 to exit 13 (Hylan Drive). Take a left onto Hylan Dr. and continue north to Jefferson Road (Route 252), and take a left at the light. Proceed west a short distance to the main campus.

Once entering campus at the main entrance (indicated with a VERY large sign: "Rochester Institute of Technology, Founded 1829"), follow the signs to the Information Booth. The Campus Safety officer will issue a parking pass and provide directions to your desired destination.

