# Table of Contents

# A Word of Appreciation

We thank our supporters, without whom this conference would not have been possible.

Technion Israel Institute of Technology

THE OPEN UNIVERSITY OF ISRAEL

מחשבה
מרכז המורים הארצי למדעי המחשב

csta Computer Science Teachers Association

Microsoft®

## Exhibitors

www. PROBOOK .co.il
The eBook Store

Taylor & Francis Online

PRINCETON UNIVERSITY PRESS

# 1. Welcome to ITiCSE 2012!

We are pleased to present the Final Program of the 17th Annual Conference on Innovation and Technology in Computer Science Education—ITiCSE 2012. This year, ITiCSE takes place in Haifa, the largest city in northern Israel. We hope you will have a chance to visit the city's sites as well as other Israeli attractions and experience the amazing variety, rich history, varied geography, ancient background and modern present of Israel.

The conference program offers interesting elements for every attendee. All three keynotes of ITiCSE 2012 are in conjunction with the Turing Centenary events of Alan Turing Year. We are looking forward to hearing keynotes by Prof. Michael Rabin, Prof. Lenore Blum, and Prof. David Harel. Apart from the keynotes, regular paper sessions, poster sessions, and tips, techniques & courseware sessions, we have four interesting working groups, and four excursions. The conference dinner will be held at the Palm Beach Club in Acre, where we will enjoy excellent food, lively music, and the breathtaking scenery of Haifa Bay and Acre's old city walls.

We accepted 60 excellent papers from among the 134 submitted—an acceptance rate of 45%. 14 of these papers were assembled in two short presentation sessions. In addition, 3 invited panels, 23 posters and 13 tips, techniques & courseware presentations were accepted. All submissions were peer-reviewed by four reviewers at least, and we thank our 145 reviewers for their help in reviewing the submissions.

ITiCSE is located in Europe, but has a worldwide outreach. We received papers from 32 different countries covering every continent excluding Antarctica. Submissions were accepted from 21 countries, including nine non-European ones. We tried to balance the program between quality and diversity, given the limited amount of time available. We believe that you will enjoy the program.

The organization of any conference is challenging, especially one so large and international in scope. We are very grateful to the many who helped make this conference possible. First we would like to thank our most excellent committee: Liz, Noa, Ronit, Bruce, Michal, John Impagliazzo and John Dooley, Henry, Arnold, Tamar Vilner, Cary, Michael Goldweber, Don, Mats, Ela, Bruria, Cecile, Dan, Niv, Tamar Paz, and Bella Furman. We also want to thank all the authors, reviewers, presenters, session chairs, working group leaders and participants. It was a pleasure to work with such motivated and experienced scientists.

We also wish to thank our contact people at ACM headquarters, Lisa Tolles and her outstanding staff at Sheridan Printing, and the members of the SIGCSE Board for supporting us all the way. Thank you also to the conference exhibitors and sponsors.

*Tami, Judith, Michael and Orit*

# 2. Conference Committee

| | |
|---|---|
| **Conference Co-Chairs**: | Tami Lapidot – Technion, Israel<br>Judith Gal-Ezer – The Open University, Israel |
| **Program Co-Chairs**: | Michael E. Caspersen – Aarhus University, Denmark<br>Orit Hazzan – Technion, Israel |
| **Working Groups Coordinators:** | Elizabeth S. Adams – James Madison University, USA<br>Noa Ragonis – Beit Berl College, Israel |
| **Panels:** | Ronit Ben-Bassat Levy – The Weizmann Institute of Science, Israel<br>Bruce J. Klein – Grand Valley State University, USA |
| **Posters:** | Michal Armoni – The Weizmann Institute of Science, Israel<br>John Impagliazzo – Emeritus, Hofstra University, USA |
| **Tips, Techniques and Courseware:** | Arnold N. Pears – Uppsala University, Sweden<br>Tamar Vilner – The Open University, Israel |
| **Treasurer and Registrar**: | Cary Laxer – Rose-Hulman Institute of Technology, USA |
| **Proceedings and Publicity:** | Michael Goldweber – Xavier University, USA<br>Don Goelman – Villanova University, USA |
| **Evaluations**: | Mats Daniels – Uppsala University, Sweden<br>Ela Zur – The Open University, Israel |
| **System Administrators:** | John Dooley – Knox College, USA<br>Henry M. Walker – Grinnell College, USA |
| **High-school Projects:** | Bruria Haberman – The Weizmann Institute of Science, Israel<br>Cecile Yehezkel – The Weizmann Institute of Science, Israel |
| **Local Committee:** | Dan Aharoni – Kinneret College on the Sea of Galilee, Israel<br>Bella Furman – Technion, Israel<br>Niv Kfir – Technion, Israel<br>Tamar Paz – Oranim College, Israel |

# 3. Conference Program

## Tuesday, July 3, 2012

| Tuesday, July 3, 2012 | |
|---|---|
| | Shuttles to the Technion from the conference hotels (see page 52) |
| **16:00** | **Tuesday Plenary session (Segoe Architecture Building Auditorium)** |
| 16:00 | Welcome and Opening Session<br>Dr. Avital Stein, Executive Vice President and Director General, Technion |
| 16:30 | Keynote: Never Too Early to Begin: Computer Science for High-School Students<br>*Michael Rabin (see bio on page 16)* |
| **17:30** | **Coffee break & demonstration of Israeli high-school students' projects (near the Segoe Auditorium)**<br>(see details on page 49**)** |
| **18:15** | **Panel I: The New CSTA K–12 Computer Science Standards (Segoe Auditorium)**<br>*Chris Stephenson; Steven Cooper; Judith Gal-Ezer; Barbara Boucher Owens* |
| **18:15** | **Papers 1: Active Learning (The Student Union, Cinema hall)**<br>**Chair: J. Mark Pullen** |
| 18:15 | Activities, affordances and attitude --- how student-generated questions assist learning<br>*Andrew Luxton-Reilly; Paul Denny; Beryl Plimmer; Robert Sheehan* |
| 18:40 | Maximizing Learning and Guiding Behavior in Free Play User Generated Content Environments<br>*Acey Boyce; Antoine Campbell; Shaun Pickford; Dustin Culler; Tiffany Barnes* |
| 19:05 | Lectures Abandoned: Active Learning by Active Seminars<br>*Henrik Bærbak Christensen; Aino Corry* |

| Tuesday, July 3, 2012 | |
|---|---|
| **18:15** | **Papers 2: K-12 I (Butler Auditorium, Samuel Neaman Institute)**<br>**Chair: Karina Assiter** |
| 18:15 | Infusing Computational Thinking into the Middle- and High-School Curriculum<br>*Amber Settle; Baker Franke; Ruth Hansen; Frances Spaltro; Cynthia Jurisson; Colin Rennert-May; Brian Wildeman* |
| 18:40 | Pseudo Abstract Composition: The Case of Language Concatenation<br>*Ronnie Alankry; David Ginat* |
| 19:05 | Teaching Graph Algorithms to Children of All Ages<br>*Paul Gibson* |
| 19:30 | Shuttles to the conference hotels |

## Wednesday, July 4, 2012

| Wednesday, July 4, 2012 | |
|---|---|
| | Shuttles to the Technion from the conference hotels (see page 52) |
| **9:00** | **Wednesday Plenary Session (Segoe Auditorium)** |
| 9:00 | Keynote: Alan Turing and the Other Theory of Computation<br>*Lenore Blum (see bio on page 17)* |
| **10:00** | **Coffee break & Posters I (near the Segoe Auditorium)** |
| **10:45** | **Panel II: Computer Science as a Community Involvement Activity (Segoe Auditorium)**<br>*Assaf Zaritsky; Ohad Barzilay* |
| **10:45** | **Papers 3: Algorithms (The Student Union, Cinema hall)**<br>**Chair: Debra Goldberg** |
| 10:45 | Forming Project Groups while Learning about Matching and Network Flows in Algorithms<br>*Dinesh Mehta; Tina Kouri; Irene Polycarpou* |

| Wednesday, July 4, 2012 | |
|---|---|
| 11:10 | Evolution of an Experimental Approach to Computer-Based, Active Learning of Greedy Algorithms<br>*J. Ángel Velázquez-Iturbide* |
| 11:35 | Digging for Algorithmic Nuggets in the Land of Polyominoes<br>*Anany Levitin* |
| **10:45** | **Papers 4: Systems (Computer Science department, room 337)**<br>**Chair: Barry Fagin** |
| 10:45 | The Empirically Refined Competence Structure Model for Embedded Micro- and Nanosystems<br>*Andre Schäfer ; Steffen Büchner; Steffen Jaschke; Harald Schmidt; Bruno Kleinert; Rainer Brück; Sigrid Schubert; Dietmar Fey* |
| 11:10 | Supporting Operating Systems Projects using the uMPS2 Hardware Simulator<br>*Michael Goldweber; Renzo Davoli; Tomislav Jonjic* |
| 11:35 | Integrating Data-Intensive Cloud Computing with Multicores and Clusters in an HPC Course<br>*Atanas Radenski* |
| **12:00** | **Short break between sessions** |
| **12:15** | **Working groups report (Segoe Auditorium)**<br>**Chairs: Elizabeth Adams, Noa Ragonis** |
| **12:15** | **Papers 5: CS1/2 I (The Student Union, Cinema hall)**<br>**Chair: Waleed Khalifa** |
| 12:15 | All Syntax Errors Are Not Equal<br>*Paul Denny; Andrew Luxton-Reilly; Ewan Tempero* |
| 12:40 | Code Comprehension Problems as Learning Events<br>*Leigh Ann Sudol-DeLyser; Mark Stehlik; Sharon Carver* |
| 13:05 | An Open-Ended Environment for Teaching Java in Context<br>*André Santos* |

| Wednesday, July 4, 2012 | |
|---|---|
| **12:15** | **Papers 6: Testing (Computer Science department, room 337)**<br>**Chair: Amber Settle** |
| 12:15 | On Teaching Arrays with Test-Driven Learning in WebIDE<br>*Michael Hilton; David Janzen* |
| 12:40 | JUG: A JUnit Generation, Time Complexity Analysis and Reporting Tool to Streamline Grading<br>*Christopher Brown; Robert Pastel; Bill Siever; John Earnest* |
| 13:05 | Exploring Influences on Student Adherence to Test-Driven Development<br>*Kevin Buffardi; Stephen H. Edwards* |
| **13:30** | **Lunch**<br>Served in the Student Union building, the Transparent Hall ("Ha-Ulam Ha-Shakuf" in Hebrew). **Please bring the Wednesday lunch voucher with you.** |
| **14:30** | **Panel III: Assessing the Benefits of Integrating Social Issues Components in the Computing Curriculum (Segoe Auditorium)**<br>*Paul Leidig; Michael Goldweber; Barbara Owens* |
| **14:30** | **Tips, Techniques and Courseware I (The Student Union, Cinema hall)**<br>**Chair: Vivienne Farrell** |
| 14:30 | Programming Studio: Advances and Lessons Learned<br>*Charlie Meyer; Michael Woodley* |
| 14:40 | Sample Courseware for Introductory OO Programming<br>*Rikki Fletcher; Rocio Guillen* |
| 14:50 | A Web-Based Problem Solving Tool for Introductory Computer Science<br>*Petr Jarušek; Radek Pelánek* |
| 15:00 | Techniques at the Intersection of Computing and Music<br>*Jesse M. Heines; Gena R. Greher; S. Alex Ruthmann* |

| Wednesday, July 4, 2012 | |
|---|---|
| 15:10 | Nintendo® DS Projects to Learn Computer Input-Output<br>*Edurne Larraza-Mendiluze; Nestor Garay-Vitoria; Jose Martín; Javier Muguerza; Txelo Ruiz-Vazquez; Iratxe Soraluze; Jose Francisco Lukas; Karlos Santiago* |
| 15:20 | Using Professional and Ethical Themes<br>*John Impagliazzo* |
| 15:30 | Breadth First Search (animation and obstacle avoidance)<br>*Arnold Rosenbloom* |
| **14:30** | **Papers 7: Assessment I (Computer Science department, room 337)**<br>**Chair: Paul Denny** |
| 14:30 | Statistical Evidence of the Correlation between Mental Ability to Compute and Student Performance in Undergraduate Courses<br>*Osvaldo Oliveira* |
| 14:55 | Grade Inflation, What Students Value, and the Necessity of Suffering<br>*Taly Sharon; Paul Kingsley* |
| 15:20 | Instructor-Centric Source Code Plagiarism Detection and Plagiarism Corpus<br>*Jonathan Y. H. Poon; Kazunari Sugiyama; Yee Fan Tan; Min-Yen Kan* |
| **15:45** | **Coffee break & Posters II (near the Segoe Auditorium)** |
| 16:30 | Shuttles to the conference hotels (see page 52) |
| **17:00** | **Break at hotels** |
| | Transportation to the Banquet in Acre (see details on page 52)<br>**Please bring the Banquet tickets with you!** |
| **19:00** | **ITiCSE 2012 Banquet (Palm Beach hotel, Acre)** |
| 23:00 | Transportation back to the conference hotels |

## Thursday, July 5, 2012

| Thursday, July 5, 2012 | |
|---|---|
| | Shuttles to the Technion from the conference hotels (see page 52) |
| **9:00** | **Thursday Plenary Session (Segoe Auditorium)** |
| 9:00 | Keynote: Standing on the Shoulders of a Giant: One Person's Experience of Turing's Impact<br>*David Harel(see bio on page 17)* |
| **10:00** | **Short break between sessions** |
| **10:15** | **Tips, Techniques and Courseware II (Segoe Auditorium)**<br>**Chair: Khaled Asad** |
| 10:15 | Teaching Labs on Pseudorandom Number Generation<br>*Elizabeth Patitsas* |
| 10:25 | Best Practices for Time-Management of Student Groups with Heterogeneous Effort<br>*André Schäfer; Matthias Mielke; Rainer Brück* |
| 10:35 | Developing Contexts for Teaching Java Using AGUIA/J<br>*André Santos* |
| 10:45 | The Presenter First Design Approach<br>*Zachary Kurmas* |
| 10:55 | Hardware Simulator for Teaching CPU Design<br>*Michael Black* |
| 11:05 | Introvert Educators : Techniques to be Effective in the Traditional Face-to-face CS classroom<br>*Karina Assiter* |
| **10:15** | **Papers 8: CS1/2 II (The Student Union, Cinema hall)**<br>**Chair: Ricardo Queiros** |
| 10:15 | Enriching Introductory Programming Courses with Non-Intuitive Probability Experiments Component<br>*Yana Kortsarts; Yulia Kempner* |

| Thursday, July 5, 2012 | |
|---|---|
| 10:40 | A Study on Students' Behaviors and Attitudes towards Learning to Program<br>*Anabela Gomes; Álvaro Santos; António Mendes* |
| 11:05 | Initial Results of Using an Intelligent Tutoring System with Alice<br>*Stephen Cooper; Yoon-Jae Nam; Luo Si* |
| **10:15** | **Papers 9: Technologies in CSE (Butler Auditorium, Samuel Neaman Institute)**<br>**Chair: Kazunari Sugiyama** |
| 10:15 | Student Reactions to Classroom Lecture Capture<br>*Paul Dickson; David Warshow; Alec Goebel; Colin Roache; W. Richards Adrion* |
| 10:40 | Beyond PDF and ePub: Toward an Interactive Textbook<br>*Bradley Miller; David Ranum* |
| 11:05 | The Future of Teaching Programming is on Mobile Devices<br>*Nikolai Tillmann; Michal Moskal; Jonathan de Halleux; Manuel Fahndrich; Judith Bishop; Arjmand Samuel; Tao Xie* |
| **11:30** | **Short break between sessions** |
| **11:45** | **Papers 10: Assessment II (Segoe Auditorium)**<br>**Chair: Maria del Carmen Calatrava Moreno** |
| 11:45 | Evaluation of a Collaborative Instructional Framework for Programming Learning<br>*Luis Miguel Serrano Cámara; Maximiliano Paredes Velasco; Jesús Ángel Velázquez Iturbide* |
| 12:10 | Capstone Project: Fair, Just and Accountable Assessment<br>*Vivienne Farrell; Gil Ravalli; Graham Farrell; Paul Kindler; David Hall* |
| 12:35 | Comparing the Effectiveness of Different Educational Uses of Program Animations<br>*Jaime Urquiza-Fuentes; J. Ángel Velázquez-Iturbide* |

| Thursday, July 5, 2012 | |
|---|---|
| **11:45** | **Papers 11: Classroom Management (The Student Union, Cinema hall)**<br>**Chair: Arkady Retik** |
| 11:45 | Pros and Cons for Teaching Classroom and Online Simultaneously<br>*J. Mark Pullen* |
| 12:10 | SpecCheck: Automated Generation of Tests for Interface Conformance<br>*Chris Johnson* |
| 12:35 | PETCHA – A Programming Exercises Teaching Assistant<br>*Ricardo Queirós; José Paulo Leal* |
| **11:45** | **Papers 12: K-12 II (Butler Auditorium, Samuel Neaman Institute)**<br>**Chair: Dinesh Mehta** |
| 11:45 | Spaghetti for the Main Course? Observations on Naturalness of Scenario-Based Programming<br>*Michal Gordon; Assaf Marron; Orni Meerbaum-Salant* |
| 12:10 | A New Curriculum for Junior-High in Computer Science<br>*Iris Zur* |
| 12:35 | Outreach for Improved Student Performance: A Game Design and Development Curriculum<br>*Katelyn Doran; Acey Boyce; Samantha Finkelstein; Tiffany Barnes* |
| **13:00** | **Lunch**<br>Served in the Student Union building, the Transparent hall ("Ha-Ulam Ha-Shakuf" in Hebrew). **Please bring the Wednesday lunch ticket with you.** |
| **14:00** | **Papers 13: Short Presentations I (Segoe Auditorium)**<br>**Chair: Assaf Marron** |
| 14:00 | CS1001.py: A Topic-Based Introduction to Computer Science<br>*Benny Chor; Rani Hod* |
| 14:10 | Are Children Capable of Learning Image Processing Concepts? Cognitive and Affective Aspects<br>*Khaled Asad; Moshe Barak* |

| Thursday, July 5, 2012 | |
|---|---|
| 14:20 | OpenIRS-UCM: an Open-Source Multi-platform for Interactive Response Systems<br>*Carlos García Sánchez; Fernando Castro Rodríguez; José Ignacio Gómez Pérez; Christian Tenllado Tenllado; Daniel Ángel Chaver Martínez; José Antonio López Orozco* |
| 14:30 | A Method to Construct Counterexamples for Greedy Algorithms<br>*Midthala Jagadish; Sridhar Iyer* |
| 14:40 | Integrating AI and Machine Learning in Software Engineering Course for High School Students<br>*Ahuva Sperling; Dorit Lickerman* |
| **14:00** | **Papers 14: Pedagogical Tools (The Student Union, Cinema hall)**<br>**Chair: Edurne Larraza-Mendiluze** |
| 14:00 | An Interactive Functional Programming Tutor<br>*Alex Gerdes; Johan Jeuring; Bastiaan Heeren* |
| 14:25 | V-Lab: A Cloud-based Virtual Laboratory Platform for Hands-On Networking Courses<br>*Le Xu; Dijiang Huang; Wei-Tek Tsai* |
| 14:50 | Serious Toys: Teaching the Binary Number System<br>*Yvon Feaster; Farha Ali; Jason Hallstrom* |
| **14:00** | **Papers 15: Computers and Society (Butler Auditorium, Samuel Neaman Institute)**<br>**Chair: Elizabeth Patitsas** |
| 14:00 | Bio1 as CS1: Evaluating a Crossdisciplinary CS Context<br>*Zachary Dodds; Ran Libeskind-Hadas; Eliot Bush* |
| 14:25 | A Study of Stereotype Threat in Computer Science<br>*Amruth Kumar* |
| 14:50 | What Do Computer Scientists Do? A Survey of CS and Non-CS Liberal Arts Faculty<br>*Hannah Fidoten; Jaime Spacco* |

| Thursday, July 5, 2012 | |
|---|---|
| **15:15** | **Coffee break (near the Segoe Auditorium)** |
| **15:35** | **Papers 16: Short Presentation II (Segoe Auditorium)**<br>**Chair: Carlos Garcia Sanchez** |
| 15:35 | Novices' Perceptions and Experiences of a Mobile Social Learning Environment for Learning of Programming<br>*Mercy Maleko; Margaret Hamilton; Daryl D'Souza* |
| 15:45 | Choosing a Study Mode in Blended Learning<br>*Mikko Myllymäki; Ismo Hakala* |
| 15:55 | cs4fn: A Flexible Model for Computer Science Outreach<br>*Chrystie Myketiak; Paul Curzon; Jonathan Black; Peter W. McOwan; Laura R. Meagher* |
| 16:05 | Anatomy, Dissection, and Mechanics of an Introductory Cyber-Security Class's Curriculum at the United States Naval Academy<br>*Christopher Brown; Frederick Crabbe; Rita Doerr; Raymond Greenlaw; Chris Hoffmeister; Justin Monroe; Donald Needham; Andrew Phillips; Anthony Pollman; Stephen Schall; John Schultz; Steven Simon; David Stahl; Sarah Standard* |
| 16:15 | Fuzzy OOP: Expanded and Reduced Term Interpretations<br>*Ronit Shmallo; Noa Ragonis; David Ginat* |
| 16:25 | Formal Learning Groups in an Introductory CS Course: A Qualitative Exploration<br>*Julie Krause; Irene Polycarpou; Cyndi Rader* |
| **15:35** | **Papers 17: Interface-Related Tools (The Student Union, Cinema hall)**<br>**Chair: Andre Santos** |
| 15:35 | Competitive Evaluation in a Video Game Development Course<br>*Manuel Palomo-Duarte; Juan Manuel Dodero; Jose Tomas Tocino; Antonio Garcia-Dominguez; Antonio Balderas-Alberico* |
| 16:00 | User Interface Evaluation by Novices<br>*Dennis Bouvier; Tzu-Yi Chen; Gary Lewandowski; Robert McCartney; Kate Sanders; Tammy VanDeGrift* |

| Thursday, July 5, 2012 | |
|---|---|
| 16:25 | MyTuringTable: A Teaching Tool to Accompany Turing's Original Paper on Computability<br>*Barry Fagin; Dino Schweitzer* |
| **15:35** | **Papers 18: Curriculum Issues (Butler Auditorium, Samuel Neaman Institute)**<br>**Chair: Michael Black** |
| 15:35 | Integrating the Teaching of Algorithmic Patterns into Computer Science Teacher Preparation Programs<br>*Noa Ragonis* |
| 16:00 | microPython: Non-Majors Programming from the Very First Lecture<br>*John Aycock* |
| 16:25 | Engaging Computer Science in Traditional Education: The ECSITE Project<br>*Debra Goldberg; Dirk Grunwald; Clayton Lewis; Jessica Feld; Sarah Hug* |
| 16:50 | A Systematic Approach to Teaching Abstraction and Mathematical Modeling<br>*Chuck Cook; Svetlana Drachova; Jason Hallstrom; Joe Hollingsworth; David Jacobs; Joan Krone; Murali Sitaraman* |
| **17:15** | **Short break between sessions** |
| **17:30** | **Closing Session (Segoe Auditorium)** |
| 18:30 | Shuttles to the conference hotels (see page 52) |

## 3.1 Keynote Speakers

**Michael Rabin**

**The Hebrew University**
**and Harvard University**

Rabin is Professor of Computer Science at Harvard SEAS. He received his Ph.D. in Mathematics at Princeton University. His contributions include innovation of non-deterministic computations, probabilistic automata, the first paper on complexity of computations, non-deterministic automata on infinite trees applicable to decision problems in logic and program verification, randomized algorithms including randomized test for primality, PKE system provably as safe as factorization, oblivious transfer, asynchronous parallel computing, a practical provably unbreakable encryption system recently implemented as a Masters Thesis Project by a student jointly supervised by Rabin and Ron Rivest Currently Rabin is working on a practically efficient zero-knowledge proof system with application to privacy and financial processes such as secure auctions, matching, and regulation compliance.

Rabin's awards include the ACM Turing Award, the Dan David Prize in Computation and Communication, the Israel Prize in Computer Science, the Harvey Prize in Science and Technology, the ACM Paris Kanellakis Prize for Theory and Practice, and the Rothschild Prize in Mathematics. His academy memberships include the US National Academy of Science, the American Philosophical Society, the French Academy of Sciences, the Royal Society and the Israel Academy of Sciences. He holds six honorary degrees. He was Albert Einstein Professor at the Hebrew University where he was also Rector (academic head). He was Visiting Professor at Berkeley, Columbia, Yale, Paris University, University College London, MIT, Cal Tech, ETH, NYU Courant Institute. He was Saville Fellow at Merton College Oxford and Steward Fellow at Gonville and Caius College Cambridge. During Spring 2009 he was Visiting Researcher at Google lab New York.

**Lenore Blum**
**Carnegie Mellon University**

Blum received her Ph.D. in mathematics from M.I.T. in 1968. She then went to UC Berkeley as a Postdoctoral Fellow and Lecturer in Mathematics. In 1973 she joined Mills College where she founded the Dept. of Mathematics and Computer Science. In 1979 she was awarded the first Letts-Villard Chair at Mills. In the fall of 1999, Blum joined Carnegie Mellon, where she is Distinguished Career Professor of Computer Science, as well as co-Director on the NSF-ITR ALADDIN Center for Algorithm Adaptation Dissemination and Integration. Blum's research, from her early work in model theory and differential fields to her more recent work in developing a theory of computation and complexity over the real numbers has focused on merging seemingly unrelated areas. Blum is well known for her work in increasing participation of girls and women in STEM fields. She was instrumental in founding the Assoc. for Women in Mathematics (serving as its President from 1975 to 1978), the Math/Science Network and its Expanding Your Horizons conferences for high school girls, and served as co-PI for the Mills Summer Mathematics Institute for undergraduate women. At CMU she has been faculty advisor to the Women@SCS and a member of the President's Diversity Advisory Council. Furthermore, Blum has served on the Council and as Vice President of the AMS (1990-1992). After representing the AMS at the Pan African Congress of Mathematicians in 1991, she became committed to building links between the American and African mathematics communities.

**David Harel**
**The Weizmann Institute of Science, Israel**

David Harel has been at the Weizmann Institute of Science in Israel since 1980. He was Department Head from 1989 to 1995, and was Dean of the Faculty of Mathematics and Computer Science between 1998 and 2004. He was also co-founder of I-Logix, Inc. He received his PhD from MIT in 1978, and has spent time at IBM Yorktown Heights, and sabbaticals at Carnegie-Mellon, Cornell, and the University of Edinburgh. In the past he worked mainly in theoretical computer science (logic, computability, automata, database theory), and he now works mainly on software and systems engineering and on modeling biological systems. He is the inventor of Statecharts and co-inventor of Live Sequence Charts (LSCs), and co-designed Statemate, Rhapsody, the Play-Engine and PlayGo. Among his books are "Algorithmics: The Spirit of Computing" and "Computers Ltd.: What They Really Can't Do", and his awards include the ACM Karlstrom Outstanding Educator Award (1992), the Israel Prize (2004), the ACM Software System Award (2007), the Emet Prize (2010), and four honorary degrees. He is a Fellow of the ACM, the IEEE and the AAAS, and a member of the Academia Europaea and Israel Academy of Sciences.

## 3.2 Faculty and Student Posters (near the Segoe auditorium)

### Posters I: Wednesday, 10:00-10:45 (near the Segoe auditorium)

| |
|---|
| The Effect of Mathematical vs. Verbal Formulation for Finite Automata<br>*Tali Dror; Dafna Levi Rashti* |
| A Model of CS Teachers' Knowledge Growth<br>*Neomi Liberman; Yifat Ben-David Kolikant; Catriel Beeri* |
| Cryptography for the Million<br>*Yonatan Chen; Eran London; Moshe Munk* |
| The Scientific Method and Software Testing Integrated into the Same Lesson<br>*Hanania T. Salzer; Bruria Haberman; Cecile Yehezkel* |
| An Animation as an illustrate tool for learning concepts in oop<br>*Yael Mussai; Neomi Liberman* |
| How Innovative Technology Tools Can Be Used to Create New Methodology for Teaching Knowledge<br>*Eti Hershkovich; Bruria Haberman* |
| Mobile Game Development Projects in CS 1<br>*Stan Kurkovsky* |
| Applying Advanced Technology Tools in Distance Learning: Case Study: Traffic Data and Road Safety<br>*Muna Abutair-Baghdadi; Khaled Asad; Jamal Raiyn* |
| Junior High School Students Performing Image Smoothening and Noise Filtering by Applying Mathematical Operations<br>*Khaled Asad* |
| Intuitive Thinking while Dealing With Abstract Data Types<br>*Waleed Khalifa* |
| Similarities in CSE and Gemara Education<br>*Galit Shriki; Bracha Daum-Reiter* |
| Teachers' Perception of Teaching Problem-solving Strategies to Novices<br>*Lavy Bunimovich* |

## Posters II: Wednesday, 15:45-16:30 (near the Segoe auditorium)

| |
|---|
| Are Students Learning Object Oriented Programming in an Object Oriented Programming Course? Student Voices<br>*Desmond Govender; Irene Govender* |
| Problem Presentation in CS1 Courses<br>*Carmen Morgado; Fernanda Barbosa* |
| A Learning Tool for MP3 Audio Compression<br>*Mohamed Hamada; Hayato Namae* |
| A Qualitative Framework for Comparison and Evaluation of Computer Science Doctoral Programs<br>*Maria del Carmen Calatrava Moreno* |
| A Structured Approach to Problem Solving in CS1<br>*Carmen Morgado; Fernanda Barbosa* |
| DSS for the Group Estimating of the Graduation Papers<br>*Kuzmitskiy Vsevolod; Davydov Boris* |
| Using Quick Response Codes for Student Interaction During Lectures<br>*Robert Law* |
| Zawilinski: Helping Beginning Programmers Conduct MediaWiki-Based Research<br>*Zachary Kurmas* |
| Kielce: Configurable HTML Course Documents<br>*Zachary Kurmas* |
| Teaching Programming on a Mobile Device<br>*Nikolai Tillmann; Judith Bishop* |
| Visual Search with Deep Zoom to Explore Curriculum Resources Interactively<br>*Arkady Retik* |

# 4. Abstracts

## 4.1 Tuesday Keynote 16:30-17:30

**Never Too Early to Begin: Computer Science for High-School Students**
*Prof. Michael O. Rabin, the Hebrew University and Harvard University*

Computer science and technology innovated over the past sixty years, have revolutionized science, the economy and societal interactions.

Inherently CS constitutes a new science combining mathematics, logic, information theory and electronics, on par with physics, chemistry and the life sciences. It is appropriate to educate students in the fundamentals of this science. The curriculum should emphasize the scientific content rather than provide mere training in some programming language.

## 4.2 Tuesday 18:15-19:30 Paper Sessions 1, 2 & Panel I

**Panel: The New CSTA K–12 Computer Science Standards**
*Chris Stephenson (Computer Science Teachers Association), Steven Cooper (Stanford University), Judith Gal-Ezer (The Open University of Israel), Barbara Boucher Owens (Southwestern University)*

In this panel, we describe the new CSTA K-12 Computer Science Standards, their organization, and their hoped-for impact.

**Activities, affordances and attitude --- how student-generated questions assist learning**
*Andrew Luxton-Reilly (The University of Auckland), Paul Denny (The University of Auckland), Beryl Plimmer (The University of Auckland), Robert Sheehan (The University of Auckland)*

Recent research has focused on tools that support the creation, review and sharing of student-generated content for peer learning. However, we know little about the student perspective of such activities. In this paper, we identify what students believe is most helpful for their learning by analyzing open-ended comments from students engaged in creating, answering and reviewing exam-style questions generated by their peers. Students report learning about content and appropriate standards of work, both individually and through interaction with peer generated resources.

**Maximizing Learning and Guiding Behavior in Free Play User Generated Content Environments**
*Acey Boyce (University of North Carolina Charlotte), Antoine Campbell (University of North Carolina Charlotte), Shaun Pickford (University of North Carolina Charlotte), Dustin Culler (University of North Carolina Charlotte), Tiffany Barnes (University of North Carolina Charlotte)*

Providing users the ability to create their own unique content in educational software and games can be highly effective at motivating the users to use and reuse the system. It is especially popular with students who self-identify as creative or wanting to do their own thing rather than a prescribed activity. Due to the popularity of user generated content modes, some users may ignore other modes the software has to offer and only create new original content. Therefore it is important to maximize the learning potential and effectively guide user behavior in a constructivist free play environment. However, in doing so it is vital that we do not hamper the creative freedom of the user, the very reason users enjoy content creation. Here we present effective strategies for meeting these goals, provide an example implementation, and present results of a study using the example.

**Lectures Abandoned: Active Learning by Active Seminars**

*Henrik Bærbak Christensen (Aarhus University), Aino Corry (Aarhus University)*

Traditional lecture-based courses are widely criticised for being less effective in teaching. The question is of course what should replace the lectures and various active learning techniques have been suggested and studied. In this paper, we report on our experiences of redesigning a software engineering course in software architecture to employ a range of active learning techniques. The course was constrained by the study administration's traditional assignment of a three hour lecturing slot and as such we had to make techniques that fitted this imposed constraint. Our contribution is active seminars as a replacement of traditional lectures, an activity template for the contents of active seminars, an account on how storytelling supported the seminars, as well as reports on our and the students' experiences.

**Infusing Computational Thinking into the Middle- and High-School Curriculum**

*Amber Settle (DePaul University), Baker Franke (University of Chicago Lab Schools), Ruth Hansen (University of Chicago Lab Schools), Frances Spaltro (University of Chicago Lab Schools), Cynthia Jurisson (University of Chicago Lab Schools), Colin Rennert-May (University of Chicago Lab Schools), Brian Wildeman (University of Chicago Lab Schools)*

In recent years there have been significant efforts to revamp undergraduate and K-12 curricula to emphasize computational thinking, a term coined by Jeannette Wing in 2006. We describe work introducing and enhancing computational thinking activities and assessments in the middle- and high-school curriculum at the University of Chicago Lab Schools. In total six courses were altered as a part of the Computational Thinking across the Curriculum Project: middle-school and high-school computer science, and high-school Latin, graphic arts, English, and history. We detail the modifications to the curriculum and discuss the successes and challenges of the project.

**Pseudo Abstract Composition: The Case of Language Concatenation**

*Ronnie Alankry (Tel-Aviv University), David Ginat (Tel-Aviv University)*

Composition is a fundamental problem solving heuristic. In computer science, it primarily appears in program design with concrete objects such as language constructs. It also appears in more abstract forms in higher-level courses. One such form is that of language concatenation in the Computational Models course. This concatenation involves the composition of two specifications of infinite sets (source languages) into a third one, and requires both abstraction and non-deterministic conception. In this paper, we illuminate behaviors of advanced high school students, with such composition. Students who encountered difficulties offered pseudo solutions, which enclosed only "surface" features and observations. We orderly display their solutions, discuss them, and offer suggestions for tutors to cope with this phenomenon.

**Teaching Graph Algorithms To Children Of All Ages**
*Paul Gibson (Telecom Sud Paris)*

We report on our experiences in teaching graph theory and algorithms to school children, aged 4 to 17. Our objectives were to demonstrate that children can discover quite complex mathematical concepts, and are able to work with abstractions and use computation reasoning from quite an early age. We provide details of our incremental approach, which can be used with students of a wide range of abilities. Also, we comment on the importance of problem based learning where the algorithms are presented as possible solutions to games or puzzles. Finally, we conclude with a number of important observations with regard to the introduction of computer science into schools.

## 4.3 Wednesday Keynote 9:00-10:00

**Alan Turing and the Other Theory of Computation**
*Lenore Blum (Carnegie Mellon University)*

The two major traditions of the Theory of Computation, each asking claim to similar motivations and aspirations, have for the most part run a parallel non-intersecting course. On one hand, we have the tradition arising from logic and computer science addressing problems with more recent origins, using tools of combinatorics and discrete mathematics. On the other hand, we have numerical analysis and scientific computation emanating from the classical tradition of equation solving and the continuous mathematics of calculus. Both traditions are motivated by a desire to understand the essence of computation, of algorithm; both aspire to discover useful, even profound, consequences.
While the logic and computer science communities are keenly aware of Alan Turing's seminal role in the former (discrete) tradition of the theory of computation, most remain unaware of Alan Turing's role in the latter (continuous) tradition, this notwithstanding the many references to Turing in the modern numerical analysis/computational mathematics literature.
In this talk I recognize Turing's work in the foundations of numerical computation. I also indicate its role in complexity theory today, and how it provides a unifying concept for the two major traditions in the Theory of Computation.

## 4.4 Wednesday 10:45-12:00 Paper Sessions 3, 4 & Panel II

**Panel: Computer Science as a Community Involvement Activity**

*Assaf Zaritsky (Tel Aviv University), Ohad Barzilay (Tel Aviv University)*

In the proposed panel we discuss the efforts made by the academy and industry to bring computer science (CS) to audience that is not exposed to CS education, not represented in the CS workforce or not accessible due to lack in technological solutions.

**Forming Project Groups while Learning about Matching and Network Flows in Algorithms**

*Dinesh Mehta (Colorado School of Mines), Tina Kouri (Colorado School of Mines), Irene Polycarpou (Colorado School of Mines)*

The matching problem in bipartite graphs is the basis for many applications that have become especially prominent with the advent of online markets that connect two entities (e.g., job-seekers and employers). Its algorithmic basis is the max-flow problem in networks, a topic that is often covered in introductory algorithms texts and courses. Separately, the Computer Science education literature is abundant with examples which indicate that the quality of the experience in the implementation of programming tasks is enhanced when done in groups. In this paper, we describe the application of a network-flow-based matching algorithm in bipartite graphs to form project groups in the algorithms course at our university. This activity simultaneously provides students with an immersive experience in a bipartite matching application. We present a small exploratory study on the effectiveness of the activity.

**Evolution of an Experimental Approach to Computer-Based, Active Learning of Greedy Algorithms**

*J. Ángel Velázquez-Iturbide (Universidad Rey Juan Carlos)*

Some years ago we presented a novel approach to the active learning of greedy algorithms. The approach was two-fold: an experimental method and the interactive assistant GreedEx that supports it. In this paper we present the evolution of the different elements of our approach, based on our experience of 5 years using and evaluating it in a course on algorithms. Firstly, usability evaluations were conducted to check the adequacy of GreedEx to its intended aims, so they guided the evolution of GreedEx. Secondly, the analysis of students' reports allowed us to identify unexpected misconceptions, which convinced us of the necessity of several didactic interventions. Our findings suggest that we succeeded both in obtaining an attractive and effective tool and in removing severe students' misconceptions. We consider that the paper is interesting to CS education researchers because of its specific contributions to the teaching of algorithms, and also as an example of a multifaceted, middle-term CS education research.

**Digging for Algorithmic Nuggets in the Land of Polyominoes**
*Anany Levitin (Villanvoa University)*

This paper seeks to demonstrate that tiling with Polyominoes — almost entirely neglected by textbooks on algorithms — can provide valuable examples of applications of algorithm design techniques, some of which are not easy to illustrate by examples from other domains.

**The empirically refined competence structure model for embedded micro- and nanosystems**
*Andre Schäfer (University of Siegen), Steffen Büchner (University of Siegen), Steffen Jaschke (University of Siegen), Harald Schmidt (University of Erlangen-Nürnberg), Bruno Kleinert (University of Erlangen-Nürnberg), Rainer Brück (University of Siegen), Sigrid Schubert (University of Siegen), Dietmar Fey (University of Erlangen-Nürnberg)*

Teaching the development of embedded micro- and nanosystems needs a well structured foundation. In this paper we show how we refined our normative competence structure model (NCSM) to an empirical competence structure model (ECSM) by incorporating the results of a survey of German experts in the field of embedded systems. In addition we introduce a course concept for a new internship which takes the results of the ECSM into account. The project is promoted by the German Research Foundation (DFG).

**Supporting Operating Systems Projects using the uMPS2 Hardware Simulator**
*Michael Goldweber (Xavier University), Renzo Davoli (University of Bologna), Tomislav Jonjic (University of Bologna)*

We live in a multicore world. In spite of this observation, none of the available system emulators designed for undergraduate education and for use in operating systems courses support multiprocessors. This paper presents \umps2, a pedagogically undergraduate-appropriate multiprocessor system emulator/architecture. Using uMPS2 educators now have the ability to structure realistic operating system projects to maximize student exposure to the ubiquitous parallelism and concurrency present in current computing devices.

**Integrating Data-Intensive Cloud Computing with Multicores and Clusters in an HPC Course**
*Atanas Radenski (Chapman University)*

This paper presents the design and implementation of a new High-Performance Computing (HPC) course. This course amalgamates the emerging trend of data-intensive cloud computing with the dominant innovation of multicore computing and the important legacy of cluster computing. While others have reported the development of novel HPC courses, we are the first, to our knowledge, to report an upper-level HPC course that introduces the increasingly popular data-intensive cloud computing paradigm - integrally with the better-established shared memory and message-passing models. This course is entirely based on free digital reading resources and open source software. The course's design rationale and implementation - including the use of open source software tools and reading resources -

may be beneficial for educators who need to design HPC courses or modules, and especially for those who would wish to cover data-intensive cloud computing.

## 4.5 Wednesday 12:15-13:30 Paper Sessions 5,6 & WG report

***Note: the information about the Working Group Session is on page 46.***

**All Syntax Errors Are Not Equal**

*Paul Denny (The University of Auckland), Andrew Luxton-Reilly (The University of Auckland), Ewan Tempero (The University of Auckland)*

Identifying and correcting syntax errors is a challenge all novice programmers confront. As educators, the more we understand about the nature of these errors and how students respond to them, the more effective our teaching can be. It is well known that just a few types of errors are far more frequently encountered by students learning to program than most. In this paper, we examine how long students spend resolving the most common syntax errors, and discover that certain types of errors are not solved any more quickly by the higher ability students. Moreover, we note that these errors consume a large amount of student time, suggesting that targeted teaching interventions may yield a significant payoff in terms of increasing student productivity.

**Code Comprehension Problems as Learning Events**

*Leigh Ann Sudol-DeLyser (Carnegie Mellon University), Mark Stehlik (Carnegie Mellon University), Sharon Carver (Carnegie Mellon University)*

Code comprehension problems have been shown to be effective assessment items in computer science education. In this paper we present qualitative and quantitative results of a study evaluating the effectiveness of code comprehension questions with feedback as learning events. Students taking an introductory programming course as a part of a university requirement interacted with an online tutoring system as a part of a homework assignment. Students answered the problems in their own words first, before selecting a multiple-choice option from the system. Both the open-ended and multiple-choice responses were collected and analyzed. Results indicate that code comprehension questions with appropriate feedback can be learning events, and asking students to self-explain the code can also help them choose the correct answer within a single problem. The use of open-ended and multiple choice responses to the same question is also shown to be useful in refining distracter items for future assessment. Recommendations from this study can be applied not only to tutoring systems, but also to the type of interactions used in worked examples in class lecture and textbook production.

**An Open-Ended Environment for Teaching Java in Context**

*André Santos (ISCTE-IUL)*

Teaching programming in context, i.e. having students learning how to program by manipulating artifacts of a familiar domain (e.g. images, card games), has demonstrated convincing results in terms of raising student retention and interest in CS. This paper

presents a pedagogical environment for teaching Java in context that is extensible with respect to the domains it supports. Instructors model domains and develop visualization widgets for rendering their objects. In turn, students use the environment to visualize and manipulate objects of the domain when solving exercises. The advantage and originality of the proposed environment is that it embodies an open-ended platform for creating diverse contexts at a low cost, standing as an enabler to widen the spectrum of abstractions for programming in context.

**On Teaching Arrays with Test-Driven Learning in WebIDE**

*Michael Hilton (California Polytechnic State University - San Luis Obispo), David Janzen (California Polytechnic State University - San Luis Obispo)*

Test-driven development (TDD) has been shown to reduce defects and to lead to better code, but can it help beginning students learn basic programming topics, specifically arrays? We performed a controlled experiment where we taught two CS0 classes about arrays, one using WebIDE, an intelligent tutoring system that enforced the use of Test-Driven Learning (TDL) methods, and one using more traditional static methods and development environment that instructed, but did not enforce the use of TDD. Students who used the TDL approach with WebIDE performed significantly better in assessments and had significantly higher opinions of their experiences than students who used traditional methods and tools.

**JUG: A JUnit Generation, Time Complexity Analysis and Reporting Tool to Streamline Grading**

*Christopher Brown (Michigan Technological University), Robert Pastel (Michigan Technological University), Bill Siever (Northwest Missouri State University), John Earnest (Michigan Technological University)*

The JUnit Generation (JUG) system provides fast, semi-automated feedback to students. It uses a Java-like script to generate unit tests and time complexity tests, then runs those tests to generate reports. The goals for JUG are improved feedback for students, and decreased preparation and grading time for instructors and grading assistants.

**Exploring Influences on Student Adherence to Test-Driven Development**

*Kevin Buffardi (Virginia Tech), Stephen H. Edwards (Virginia Tech)*

Test-Driven Development (TDD) is a software development process with a test-first approach that shows promise for improving code quality. Our research addresses concerns raised in both academia and industry about a lack of motivation or acceptance in adopting TDD. In a CS2 class, we used an automated testing tool and post-class surveys to observe patterns of behavior in testing as well as changes in attitudes. We found significant positive outcomes for students following TDD. We also identified obstacles deterring students from adhering to TDD and discuss reasons and possible remedies.

## 4.6 Wednesday 14:30-15:45 Paper Session 7 & Panel III & TT&C I

**Panel: Assessing the Benefits of Integrating Social Issues Components in the Computing Curriculum**

*Paul Leidig (Grand Valley State University), Michael Goldweber (Xavier University), Barbara Owens (Southwestern University)*

The inclusion of social issues, including ethical and professional topics, in computing curricula has become commonplace two decades after being incorporated into the ACM Computing Curricula. However, authors of academic papers and conference presentations often concentrate on integrating the broader issues of societal impact and best practices into computing curricula, while neglecting the assessment of their benefits. This panel explores how the institutions of the panelists include social issues in projects and the curriculum as a whole, and additionally how they assess the benefits of doing so. Special attention is given to an appreciation of the social good emanating from the use of community-based and non-profit organizations in student projects. Additionally, ways to assess the effectiveness of these approaches are presented in an effort to help meet model curriculum guidelines and accreditation requirements.

**TT&C: Programming Studio: Advances and Lessons Learned**

*Charlie Meyer (University of Illinois at Urbana-Champaign), Michael Woodley (University of Illinois at Urbana-Champaign)*

Previous work described an innovative approach to teaching programming fundamentals to undergraduate students in a course called Programming Studio. This required program is designed to be given shortly after CS2. It teaches topics such as high quality code design, proper problem decomposition and methods of code documentation, as well as practical subjects such as correct usage of various types of version control and a variety of programming languages that are not usually introduced in other CS courses. Here we build on what was previously written with the new methods of assessment, types of assignments, and the lessons learned from expanding the course from less than 75 students to nearly 150 students.

**TT&C: Sample Courseware for Introductory OO Programming**

*Rikki Fletcher (California State University, San Marcos), Rocio Guillen (California State University, San Marcos)*

In this paper, we describe a programming assignment for first semester computer science students that requires them to apply object-oriented (OO) programming and design skills in order to successfully complete their work. The students work in an interactive 3D environment that gives them immediate visual feedback.

**TT&C: A Web-Based Problem Solving Tool for Introductory Computer Science**

*Petr Jarušek (Faculty of Informatics), Radek Pelánek (Faculty of informatics)*

We present a "Problem Solving Tutor" -- a web-based educational tool for learning through problem solving. The tool contains more than 1 400 problems, mainly introductory programming problems, math and logic puzzles. All problems are interactive and the system gives students immediate feedback on their performance. The tool makes individual predictions of problem solving times and therefore is able to recommend each student a problem of suitable difficulty.

**TT&C: Techniques at the Intersection of Computing and Music**

*Jesse M. Heines (Univ. of Massachusetts Lowell), Gena R. Greher (Univ. of Massachusetts Lowell), S. Alex Ruthmann (Univ. of Massachusetts Lowell)*

Our work on Performamatics aims to enhance students' computational thinking (CT) by engaging them in fundamental concepts that unite computing and music. Our approach leverages students' near universal interest in music as a context for rich CT experiences. The techniques we share are used in a General Education course open to students in any major called Sound Thinking, which is now being offered for the fourth time. Our presentation will feature videos, recordings, and live examples of student work to demonstrate the types of results that students produce for the assignments we describe.

**TT&C: Nintendo® DS Projects to learn Computer Input-Output**

*Edurne Larraza-Mendiluze (UPV/EHU), Nestor Garay-Vitoria (UPV/EHU), Jose Martín (UPV/EHU), Javier Muguerza (UPV/EHU), Txelo Ruiz-Vazquez (UPV/EHU), Iratxe Soraluze (UPV/EHU), Jose Francisco Lukas (UPV/EHU), Karlos Santiago (UPV/EHU)*

Being the Input/Output subsystem an important topic in the area of Computer Architecture [1], in this work we propose a Project-Based Learning (PBL) methodology enhanced by the use of a Nintendo®1 DS machine. The use of both, methodology and platform, is endorsed by students' comments, the projects themselves and the scores obtained in the subject.

**TT&C: Using Professional and Ethical Themes**

*John Impagliazzo (Hofstra University)*

This presentation describes ways in which teachers could use computing professionalism and ethics to enrich topics in their courses. These enrichment techniques could become a very effective extension to teaching computing courses. Using professionalism as a pedagogical technique could contribute to the understanding of course material and to the foundations of lifelong learning experiences of students. Professional extensions and ethics also encourage students to appreciate the subject at hand and foster a sense of natural inquiry surrounding their area of study that includes the human dimension.

**TT&C: Breadth First Search (animation and obstacle avoidance)**

*Arnold Rosenbloom (University of Toronto)*

A simple, motivating first year assignment which animates a Breadth First Search solution to a graphical obstacle avoidance problem is presented.

**Statistical Evidence of the Correlation between Mental Ability to Compute and Student Performance in Undergraduate Courses**

*Osvaldo Oliveira (Faculty of Campo Limpo Paulista)*

In recent years, many studies have suggested the hypothesis that mental ability to compute is essential for many activities and is as fundamental as reading, writing and arithmetic. This work provides mathematical arguments to verification of this hypothesis. Assuming a precise statement of what is to compute, based on the model of computation "Turing Machine", we found experimentally the existence of a correlation, statistically significant, between mental ability to compute and student performance in four different university courses.

**Grade Inflation, What Students Value, and the Necessity of Suffering**

*Taly Sharon (University of Liverpool), Paul Kingsley (University of Liverpool)*

This paper questions whether there is a conflict between the academic and business interests of a university in combating grade inflation. A survey of online Master's degree students in Computing at a British university was carried out. It examined variations in the perceived value of a degree or the university's perceived reputation as a result of a number of possible changes aimed at reducing the sacrifices involved in gaining a degree; making it less likely that students would fail modules or their degree; and reducing the degree dropout rate to almost zero. The conclusion is that students saw such changes as reducing the value of their degree. The research suggests that in certain circumstances, academic and business interests can be aligned in opposing grade inflation.

**Instructor-Centric Source Code Plagiarism Detection and Plagiarism Corpus**

*Jonathan Y. H. Poon (National University of Singapore), Kazunari Sugiyama (National University of Singapore), Yee Fan Tan (KAI Square), MIn-Yen Kan (National University of Singapore)*

Existing source code plagiarism systems focus on the problem of identifying plagiarism between pairs of submissions. The task of detection, while essential, is only a small part of managing plagiarism in an instructional setting. Holistic plagiarism detection and management requires coordination and sharing of assignment similarity -- elevating plagiarism detection from pairwise similarity to cluster-based similarity; from a single assignment to a sequence of assignments in the same course, and even among instructors of different courses. To address these shortcomings, we have developed our Student Submissions Integrity Diagnosis (SSID), an open-source system that provides holistic plagiarism detection in an instructor-centric way. SSID's visuals show overviews of plagiarism clusters throughout all assignments in a course as well as highlighting most-similar submissions on any specific student. SSID supports plagiarism detection workflows; e.g., allowing student assistants to flag suspicious assignments for later review and confirmation by an instructor with proper authority. Evidence is automatically entered into SSID's logs and shared among instructors. We have additionally collected a source code plagiarism corpus, which we employ to identify and correct shortcomings of previous plagiarism detection engines and to optimize parameter tuning for SSID deployment. Since its deployment, SSID's workflow enhancements have made plagiarism detection in our faculty less tedious and more successful.

## 4.7 Thursday Keynote 9:00-10:00

**Standing on the Shoulders of a Giant: One Person's Experience of Turing's Impact**
*David Harel (The Weizmann Institute of Science, Israel)*

The talk will briefly describe three of Turing's major achievements, in three different fields: computability, biological modeling and artificial intelligence. Interspersed with this, I will explain how each of them directly motivated and inspired me to carry out a variety of research projects over a period of 30 years, the results of which can all be viewed humbly as extensions and generalizations of Turing's pioneering and ingenious insights.

## 4.8 Thursday 10:15-11:30 Paper Session 8,9 & TT&C II

**TT&C: Teaching Labs on Pseudorandom Number Generation**
*Elizabeth Patitsas (University of Toronto)*

This presentation describes our work teaching labs on pseudorandom number generation (PRNG) at the Universities of British Columbia and Toronto. We use PRNG as an example of an application of sequential circuitry in our digital logic courses. This gives us an example to discuss ``what is randomness?", security issues relating to seeding and encryption, and why and how we use randomness in computing.

**TT&C: Best Practices for Time-Management of Student Groups with Heterogeneous Effort**
*André Schäfer (University of Siegen), Matthias Mielke (University of Siegen), Rainer Brück (University of Siegen)*

The daily work of an engineer in industry is affected by project work; often more than one project at a time. Consequently, time and self-management is an important part of the education of future engineers and scientists. Students of electrical engineering and computer science at the University of Siegen have a student project work in their curricula to deepen hard skills and to learn soft skills, like teamwork, organization and time-management. Often these groups consist of students with heterogeneous requirements in terms of working hours. In this contribution, the authors present a good practice for time-management of groups with heterogeneous working-time.

**TT&C: Developing Contexts for Teaching Java Using AGUIA/J**
*André Santos (ISCTE-IUL)*

Teaching programming in context, i.e. having exercises dealing with artifacts of a familiar domain (e.g. images, card games), has demonstrated convincing results in terms of raising student retention and interest in CS. This session addresses the development of contexts for Java by extending the AGUIA/J environment.

**TT&C: The Presenter First Design Approach**

*Zachary Kurmas (Grand Valley State University)*

Presenter First is a design technique that combines the Model-View-Presenter design pattern with the concept of "user stories." We have found that Presenter First helps students decompose their programming projects into manageable tasks.

**TT&C: Hardware Simulator for Teaching CPU Design**

*Michael Black (American University)*

This presentation describes a GUI-based tool for teaching CPU design to computer architecture students. A datapath builder allows microarchitecture building blocks, such as registers, ALUs, and multiplexors, to be laid out and wired together. A control builder allows a control state machine to be developed for the datapath. Because the processor design is simulated within a full PC emulator, student-designed processors can use emulated devices, such as drives, video, and I/O ports. A tutorial teaches students to use the simulator to build and simulate a pipelined RISC processor.

**TT&C: Introvert Educators : Techniques to be Effective in the Traditional Face-to-face CS classroom**

*Karina Assiter (Wentworth Institute of Technology)*

This purpose of this session is two-fold: first, to consider an initial set of strategies that introvert educators could adopt (and then adapt) in order to be effective in the traditional face-to-face computer science classroom; and, then, to initiate research (and dialogue) about educator personality types, in general, and their associated best teaching practice, since, just as preferred learning style depends on the personality of the learner, best teaching practices should be determined based (among other things) upon the personality type of the instructor.

**Enriching Introductory Programming Courses with Non-Intuitive Probability Experiments Component**

*Yana Kortsarts (Widener University), Yulia Kempner (Holon Institute of Technology)*

Probability theory is branch of mathematics that plays one of the central roles, not only in computer science, but also in science at large. We present a way to integrate non-intuitive probability experiments into introductory programming courses. We consider a set of the probability problems in which the intuitive approach leads to the wrong solution. These problems are based on interesting scenarios, attractive to students, and could be successfully "translated" into programming assignments. We present and discuss numerical simulations and verification of the correct solution through the computational approach. The proposed enrichment provides opportunity to engage students in experimental problem solving. Surprising computational results enhance students' curiosity and interest. This component promotes active involvement in the course. In addition, these problems provide an opportunity to make a connection between mathematics and computer science topics. Such non-intuitive answers may be remembered by students and may promote better understanding in the basic probability course they take later.

**A study on students' behaviors and attitudes towards learning to program**

*Anabela Gomes (University of Coimbra and Engineering Institute of Coimbra), Álvaro Santos (University of Coimbra and Engineering Institute of Coimbra), António Mendes (University of Coimbra)*

Learning difficulties in introductory programming courses are well known to teachers and students. Although several types of causes for those difficulties can be pointed out, in this work we focused on student related issues, namely their study methods and attitudes towards learning to program. We found a strong correlation between students' results and their personal perceptions of competence during the course. This result raises the need for teachers to consider this issue when devising the pedagogical approaches to use in introductory programming courses.

**Initial Results of Using an Intelligent Tutoring System with Alice**

*Stephen Cooper (Stanford University), Yoon-Jae Nam (Stanford University), Luo Si (Purdue University)*

This paper describes the initial steps taken towards incorporating an intelligent tutoring system (ITS) into Alice. After initially describing an ITS, the paper focuses on the development of several tutorials for teaching specific introductory programming concepts that have been created using stencils. Initial results concerning usability and effectiveness of these stencil-based tutorials are provided.

**Student Reactions to Classroom Lecture Capture**

*Paul Dickson (Hampshire College), David Warshow (Hampshire College), Alec Goebel (Hampshire College), Colin Roache (Hampshire College), W. Richards Adrion (University of Massachusetts)*

This paper evaluates the benefits and drawbacks of lecture recording, which aspects of lectures and lecture capture systems are most used, and what additional features and functions would make the experience more effective. We evaluated 4 computer science courses recorded during spring 2011 using our comprehensive lecture capture system PAOL and presented with webMANIC. We discuss the results of student surveys and focus groups and compare these with prior surveys that investigated how students reacted to the availability of online lecture content and how they used these resources in large- and small-scale deployments with both home-grown and commercial lecture capture technologies. The primary motivation for this study was to analyze how lecture capture fits in the context of computer science curricula and pedagogy and about how we can enhance our systems to be more educationally effective.

**Beyond PDF and ePub: Toward an Interactive Textbook**

*Bradley Miller (Luther College), David Ranum (Luther College)*

This paper describes a new and unique vision for electronic textbooks. It incorporates a number of active components such as video, code editing and execution, and code visualization as a way to enhance the typical static electronic book format. In addition, the

textbook is created with an open source authoring system that has been developed to allow the instructor to customize the content of the active and passive parts of the text. Initial results of a semester long trial are presented as well.

**The Future of Teaching Programming is on Mobile Devices**

*Nikolai Tillmann (Microsoft Research), Michal Moskal (Microsoft Research), Jonathan de Halleux (Microsoft Research), Manuel Fahndrich (Microsoft Research), Judith Bishop (Microsoft Research), Arjmand Samuel (Microsoft Research), Tao Xie (North Carolina State University)*

From paper to computers, the way that we have been writing down thoughts and performing symbolic computations has been constantly evolving. Teaching methods closely follow this trend, leveraging existing technology to make teaching more effective, and preparing students for their later careers with the available technology. Right now, in 2012, we are in the middle of another technology shift: instead of using PCs and laptops, mobile devices are becoming more prevalent for most everyday computing tasks. In fact, never before in human history were incredibly powerful and versatile computing devices such as smartphones available and adopted so broadly. We propose that computer programming, and thus the teaching of programming, can and should be done directly on the mobile devices themselves, without the need for a separate PC or laptop to write code. Programming on mobile devices engages students in new ways, allowing them to access and manipulate programmatically their most personal digital data such as pictures, videos, and music in an easy and intuitive way. Programming on smartphones that we carry around with us at all times means instant gratification for students, as they can show their games and applications to their friends, and it means that students can do their homework or additional practicing at all times. We describe such a mobile programming environment and call out challenges that need to be overcome and opportunities that it creates.

## 4.9 Thursday 11:45-13:00 Paper Session 10, 11, 12

**Evaluation of a Collaborative Instructional Framework for Programming Learning**

*Luis Miguel Serrano Cámara (Universidad Rey Juan Carlos), Maximiliano Paredes Velasco (Universidad Rey Juan Carlos), Jesús Ángel Velázquez Iturbide (Universidad Rey Juan Carlos)*

ur work lies within the collaborative approach to learning CS1. We developed an instructional framework for collaborative learning (named CIF) that is oriented by educational objectives at the analysis level of Bloom's taxonomy. Moreover, the framework is supported by a platform (named MoCAS) of collaborative tools delivered with mobile devices. In this paper an experiment is presented to evaluate the educational effectiveness of CIF and MoCAS vs. three more different learning approaches. We selected a representative topic of analysis tasks, namely scope of identifiers. The experiment was scheduled with four groups, instructed using the classical lecture format, a collaborative approach, the CIF framework, and CIF plus the MoCAS tool. The experiment obtains statistically significant results showing enhanced learning for the joint use of CIF and MoCAS.

**Capstone Project: Fair, Just and Accountable Assessment**

*Vivienne Farrell (Swinburne University), Gil Ravalli (Swinburne University), Graham Farrell (Swinburne University), Paul Kindler (Swinburne University), David Hall (Swinburne University)*

Fair, just and accountable assessment for the individual student in the capstone project requires establishing criteria beyond the final project output to meet the objectives of the capstone project experience. Current research is limited and requires solutions that are not subjective, enable formative and summative assessment with student input and are not onerous on supervisors and/or students. This paper draws on input from highly experienced academics supervising capstone projects in ICT, critically reviews recent publications pertaining to capstone project assessment and current capstone students opinions. The outcome of this input led to the development, trial and evaluation of a toolkit to assist supervisors in both formative and summative assessment of capstone projects.

**Comparing the Effectiveness of Different Educational Uses of Program Animations**

*Jaime Urquiza-Fuentes (Universidad Rey Juan Carlos), J. Ángel Velázquez-Iturbide (Universidad Rey Juan Carlos)*

In this paper we study two different approaches to using program animations with educational aims: their construction by students –a constructivist and active approach– and their vision –a less active approach. In addition, we compare both approaches to a traditional teaching methodology where animations are not used. We have conducted an experiment with functional program animations using an existing IDE with visualization features called WinHIPE. We have analyzed the results in terms of Bloom's Taxonomy and the complexity of the topics covered. We have detected learning improvements in high levels of Bloom's Taxonomy, namely analysis and synthesis. Moreover, our results show that program animations are unnecessary for simple topics, support the joint use of vision and construction tasks in medium-complexity topics, and recommend vision tasks together with the typical methodology but without the use of animations in the most complex topics.

**Pros and Cons for Teaching Classroom and Online Simultaneously**

*J. Mark Pullen (George Mason University)*

In adding an online delivery option to an existing graduate program in computer science, important choices include (1) synchronous or asynchronous delivery and (2) for synchronous delivery, whether to deliver classroom and online teaching simultaneously. This paper describes the decisions made in adding an online option to an MSCS program, which now is maturing. The factors in favor of and against simultaneous delivery are discussed in detail and the conclusions reached for one particular program are described, including experience gained in the process and an open-source software system developed to *support this form of delivery.*

**SpecCheck: Automated Generation of Tests for Interface Conformance**

*Chris Johnson (University of Wisconsin, Eau Claire)*

Grading code in large introductory programming classes is fraught with difficulties. In particular, students deviate from the required specifications---and these deviations in turn break instructors' test code, delay feedback, hurt grades, and drive instructors to create oppressive homework policies. We present a system called SpecCheck for automatically generating tests that inform students of these deviations before they submit their homework. Our system inspects a instructor's reference implementation of a homework and produces tests to validate student code for interface conformance. It aids in conditioning code for faster grading but avoids the hazards introduced by some autograders. We present the tool and an experience report of its effectiveness in a large first-semester programming course.

**PETCHA - A Programming Exercises Teaching Assistant**

*Ricardo Queirós (CRACS & INESC-Porto LA & DI-ESEIG/IPP), José Paulo Leal (2CRACS & INESC-Porto LA, University of Porto)*

This paper presents a tool called Petcha that acts as an automated Teaching Assistant in computer programming courses. The ultimate objective of Petcha is to increase the number of programming exercises effectively solved by students. Petcha meets this objective by helping both teachers to author programming exercises and students to solve them. It also coordinates a network of heterogeneous systems, integrating automatic program evaluators, learning management systems, learning object repositories and integrated programming environments. This paper presents the concept and the design of Petcha and sets this tool in a service oriented architecture for managing learning processes based on the automatic evaluation of programming exercises. The paper presents also a case study that validates the use of Petcha and of the proposed architecture.

**Spaghetti for the Main Course? Observations on Naturalness of Scenario-Based Programming**

*Michal Gordon (Weizmann Institute of Science), Assaf Marron (Weizmann Institute of Science), Orni Meerbaum-Salant (Weizmann Institute of Science)*

Scenario-based programming is an approach to software development which calls for developing independent software modules to describe different behaviors that the system should or should not follow, and then coordinating the interwoven execution of these modules at run time. We show that patterns previously shown to exist in programs written in the Scratch environment, which is not specifically scenario oriented, by children who did not have other training, and were not guided to write in a scenario-based manner, are also characteristic to scenario-based programming. These patterns include extremely fine-grain decomposition and bottom-up development. This result suggests that scenario-based programming concepts are "natural" in some ways. Thus, with an appropriate environment and a matching set of tools, scenario-based programming concepts could have an important role in early-stage computer-programming curricula.

**A New Curriculum for Junior-High in Computer Science**
*Iris Zur (Babeş-Bolyai University of Romania)*

Israel's Ministry of Education has launched a unique program to enhance science - technology education. It is a six year program for grades seven through twelve. The program introduces a new curriculum in computer science for junior-high school students. The computer science curriculum focuses on developing computational thinking. The purpose of this paper is to describe that curriculum and the preliminary evaluation of students' achievements.

**Outreach for Improved Student Performance: a game design and development curriculum**
*Katelyn Doran (University of North Carolina at Charlotte), Acey Boyce (University of North Carolina at Charlotte), Samantha Finkelstein (Carnegie Mellon University), Tiffany Barnes (University of North Carolina at Charlotte)*

We present a curriculum for computer science outreach using Game Maker. This curriculum has been adapted over six iterations of a 10-week, middle school apprenticeship on Game Design and Development. Through multiple iterations we have adjusted for many of the issues one could expect to encounter when running a similar outreach program. While many outreach curricula are independent from coursework, our video game design curriculum is targeted to student learning objectives and designed for integration into the classroom. We demonstrate that students' language arts and math classroom performance have improved with participation in this apprenticeship.

## 4.10 Thursday 14:00-15:15 Paper Session 13, 14, 15

**CS1001.py: A Topic-Based Introduction to Computer Science**
*Benny Chor (Tel-Aviv University), Rani Hod (Tel-Aviv University)*

We describe the curriculum, initial experience, and preliminary evaluation of an introductory CS course for students taking CS as their single or double major. The course is taught during the first or second semester of the first year of studies. It is centered around eleven to thirteen topics, offering a wide cover of major CS subjects. Many of these topics are not covered in "traditional" introductory CS courses, and some of them are not even covered through the standard undergraduate curricula. Examples: digital image representation and processing, error correction and detection codes, and text compression. The programming language used in the course is Python. The students are exposed to all standard programming language constructs (commands, assignments, functions, conditionals, iterations, recursion, etc.) and basic data types (e.g. integers, floating-point numbers, lists, dictionaries, and sets), as well as less basic constructs, like higher order functions, lambda expressions, classes and methods, and iterators. We have set a dual learning outcome: the students are expected to acquire a good knowledge and proficiency of programming and understanding short programs. We also expect them to get a broad view of central subjects in Computer Science.

**Are Children Capable of Learning Image Processing Concepts? Cognitive and Affective Aspects**

*Khaled Asad (Alqasemi Academic College of Education & Ben-Gurion University of the Negev), Moshe Barak (Ben-Gurion University of the Negev)*

Many children are making extensive use of technological devices such as cellphones, digital cameras and music players. Are these children attracted to learning the computer principles on which these devices are based? Are they qualified to learn computer science concepts in an in-depth fashion? The goal of this study was to examine the case of teaching an advanced computer science course in image processing to middle school students. During the stage of learning the theory, the students exhibited rather low motivation; later, when they were involved in preparing individual projects on image processing, they demonstrated greater interest and success in learning the subject matter.

**OpenIRS-UCM: an Open-Source Multi-platform for Interactive Response Systems**

*Carlos García Sánchez (Universidad Complutense de Madrid), Fernando Castro Rodríguez (Universidad Complutense de Madrid), José Ignacio Gómez Pérez (Universidad Complutense de Madrid), Christian Tenllado Tenllado (Universidad Complutense de Madrid), Daniel Ángel Chaver Martínez (Universidad Complutense de Madrid), José Antonio López Orozco (Universidad Complutense de Madrid)*

Interactive Response Systems have been gaining acceptance within the educational community in recent years and a clear proof is the growing number of commercial systems available today in the market. However, most solutions are based on closed systems, rigid and dependent on proprietary keypad or platform. \openirsS is a free teaching tool for interactive polling that solves these drawbacks. It is an open source software that allows the development of new functionality by users outside the creators. It has a friendly interface and it is easy to use without high computer skills. It enables the coexistence of several commercial clickers simultaneously with next-generation mobile phones, tablets or other modern electronic devices. It is developed in Java so its use is not restricted to systems based on Microsoft Windows and it is independent of any proprietary software. The excellent acceptance of this teaching tool by the Complutense University of Madrid community has encouraged a repository creation in order to facilitate its widespread use.

**A Method To Construct Counterexamples For Greedy Algorithms**

*Midthala Jagadish (IIT Bombay), Sridhar Iyer (IIT Bombay)*

Problem solving is the focus of any algorithm design course. Being able to construct counterexamples is a critical part of the problem solving process. We address the following question: Is there a teachable approach to construct counterexamples? We present a technique, called the Anchor Method, that could prove useful in constructing counterexamples for greedy algorithms. The basic idea of the method is to get insight about the structure of a counterexample by weakening the greedy algorithm. We show the applicability of the method by constructing counterexamples for some classic problems in graph theory. We also investigate the teachability of the technique through experiments.

**Integrating AI and Machine Learning in Software Engineering course for High School Students**

*Ahuva Sperling (Leo Baeck education Center), Dorit Lickerman (Leo Baeck education Center)*

This paper describes a unique software engineering curriculum for high-school students that includes subjects in artificial intelligence and machine learning. The students in the course deal with the implementation of solutions to riddles and games (complex algorithmic problems), use the DrRacket functional programming language as a tool that supports their comprehension and thorough understanding of blind search algorithms, informed search algorithms, search games trees and machine learning algorithms. During their studies, the students engage in self-learning, collaborative learning and peer teaching. At the end of the course, each student writes a final research project which integrates the main aspects that have been learned in the course.

**An Interactive Functional Programming Tutor**

*Alex Gerdes (Open Universiteit Nederland), Johan Jeuring (Utrecht University), Bastiaan Heeren (Open Universiteit Nederland)*

We introduce an interactive tutor that supports the stepwise development of simple functional programs. Using this tutor, students receive feedback about whether or not they are on the right track, can ask for a hint when they are stuck, and get suggestions about how to refactor their program. Our tutor generates this semantically rich feedback from model solutions, using advanced concepts from software technology. We show how a teacher can add an exercise to the tutor, and fine-tune feedback. We report on an experiment in which we used our tutor.

**V-Lab: A Cloud-based Virtual Laboratory Platform for Hands-On Networking Courses**

*Le Xu (Arizona State University), Dijiang Huang (Arizona State University), Wei-Tek Tsai (Arizona State University)*

For computer-network education, hands-on laboratories are essential in addition to lectures. Existing laboratory solutions are usually expensive to build, configure and maintain, while still lacking reusability, flexibility and scalability. This paper presents a cloud-based virtual laboratory education solution, called V-Lab, where instructors can use an interactive web interface to customize computer network testbeds based on a set of dedicated virtual computers interconnected through VLAN-based virtual networks. The established virtual network system can be accessed via remote access using standard Secure Shell (SSH), Virtual Network Computing (VNC), or Remote Desktop Protocol (RDP). By using a flexible and configurable design, V-Lab greatly reduces the effort needed to establish and maintain a physical laboratory including hardware and software, while providing a secure and reliable environment that encourages students to use the resources based on their own schedule. V-Lab also helps re-design laboratory curriculum to focus on six educational factors, and the survey results show that V-Lab system is easy to use and setup and has satisfactory performance and reliability. It is also indicated that V-Lab helps students understand and solve real-world problems with sufficient laboratory resources and improved efficiency.

**Serious Toys: Teaching the Binary Number System**

*Yvon Feaster (Clemson University), Farha Ali (Lander University), Jason Hallstrom (Clemson University)*

The binary number system is the lingua franca of computing, requisite to myriad areas, from hardware architecture and data storage to wireless communication and algorithm design. Given its significance to such a broad range of computing topics, it is not surprising that the binary number system plays a prominent role in K-12 outreach efforts. It is even less surprising that the topic is often viewed as a dreary introduction to the discipline. Motivated by these observations and the potential of binary arithmetic to connect future students to a wide spectrum of computing topics, we have developed a new approach to teaching binary arithmetic in the K-12 curriculum. The approach relies on the use of a "serious toy", an embedded hardware platform designed to teach the binary number system while engaging visual and kinesthetic learners. We describe the design of the curriculum module and the supporting toy and detail our experiences using the approach in three independent outreach efforts. The results are largely positive, supporting our supposition that teaching the binary number system can achieve strong content understanding and improved attitudes toward the discipline.

**Bio1 as CS1: evaluating a crossdisciplinary CS context**

*Zachary Dodds (Harvey Mudd College), Ran Libeskind-Hadas (Harvey Mudd College), Eliot Bush (Harvey Mudd College)*

We present the curriculum, deployment, and initial evaluation of a course, BioCS1, designed to serve as an introductory course in both biology and CS. Co-taught by professors in both fields, BioCS1 interweaves fundamental biology and computational topics in a manner similar to contextual approaches to CS1. In contrast to other contextual approaches, however, BioCS1 emphasizes both CS and its context equally. The results suggest that such cross-disciplinary collaborations can thrive at the introductory level, just as they have later in the curriculum.

**A Study of Stereotype Threat in Computer Science**

*Amruth Kumar (Ramapo College of New Jersey)*

A controlled study was conducted to detect stereotype threat on harder topics in introductory Computer Science. Students in the control group were asked to identify their demographic information before taking the test whereas students in the experimental group were asked to do so after completing the test. So, the control group was indirectly reminded of stereotypes before taking the test, when it could affect performance on the test, whereas the experimental group was reminded after the test when it could not affect test performance. Tests on two different loops were used, along with a partial cross-over design: a random group of students served as the control group on one test, and the experimental group on the other and vice versa. Mixed factor ANOVA analysis of the data showed that all the students scored significantly higher when not reminded of stereotypes before the test regardless of sex or race. In addition, average/below-average students scored significantly higher when not reminded of stereotypes before the test,

again, regardless of sex or race. So, on harder topics in Computer Science, stereotype threat affects all the students, and in particular, the less-prepared students. In light of this, some suggestions are offered for avoiding stereotype threat during tests.

**What Do Computer Scientists Do? A Survey of CS and Non-CS Liberal Arts Faculty**
*Hannah Fidoten (Knox College), Jaime Spacco (Knox College)*

We asked all of the liberal arts faculty who advise undergraduates on course selection at the 14 colleges in the Associated Colleges of the Midwest a series of questions regarding their perceptions of the personality traits of Computer Science (CS) students, topics they think are covered in CS classes, and their overall impressions of CS. Our goal was to assess empirically the hypotheses that (1) many non-CS faculty do not really know what computer science is, and (2) many non-CS faculty are unaware of the differences between CS and Information Technology (IT). We received over 250 survey responses, which revealed that, among non-CS faculty, 9% disagree or are neutral that CS should even be part of a liberal arts curriculum, 9% think that CS classes teach students to fix printers and other peripherals, and 34% believe that CS students are taught to use Microsoft products in the classroom. However, over 95% of non-CS faculty also recognize the importance of both programming and algorithms to the study of computer science.The overall data suggests that a majority of non-CS faculty in the Associated Colleges of the Midwest have a basic understanding of CS, while the remainder have an overly broad definition of CS (i.e. they think that CS includes IT).

## 4.11 Thursday 15:30-17:10 Paper Session 16, 17, 18

**Novices' Perceptions and Experiences of a Mobile Social Learning Environment for Learning of Programming**
*Mercy Maleko (RMIT University), Margaret Hamilton (RMIT University), Daryl D'Souza (RMIT University)*

In this paper we report programming novices' perceptions and experiences of a Mobile Social Learning Environment (MSLE) designed to support the learning of programming through enabling increased novice-to-novice interactions. By capturing and analysing such interactions early it is hoped that instructors may identify misunderstandings and misconceptions of novice programmers, and provide timely feedback to alleviate novices' misunderstandings. The MSLE incorporates the use of mobile devices as well as the access to dedicated social networking sites. Novices from three different universities, each with different modes of delivery, participated in this research. Novices' perceived advantages of the MSLE include among other things instant access to discussions, increased interactions, instant feedback in a student-student context, and the ability to share ideas and views anytime, anywhere. Novices have experienced increased interactions among themselves and improved engagement with learning in the MSLE. Novices were able to form learning communities and able to share knowledge and ideas about programming language with each other. Some novices were able to provide assistance to others, hence indirectly, strengthening

strengthening their own knowledge of programming. The cost of mobile devices and Internet charges on the other hand were perceived as disadvantages of a MSLE. Small screens, lack of programming applications for smart phones, laptops' battery life and distractions from other internet media were some of negative experiences observed by novices who used the MSLE.

### Choosing a Study Mode in Blended Learning

*Mikko Myllymäki (University of Jyväskylä), Ismo Hakala (University of Jyväskylä)*

Education providers aim to meet today's education requirements by employing, among other things, education technology solutions that increase flexibility. This has also happened with the master's level mathematical information technology degree directed to adults. In the degree program, lecture videos brought in together with face-to-face teaching provide good opportunities for flexible educational arrangements. Education with the help of videos can be arranged in such a way that students themselves will be able to choose their study mode in accordance with their needs. When students themselves can choose their study modes, many different ways to take advantage of the flexibility provided are created. It seems that students with different participation styles achieve the aims for their study progress in different ways. The purpose of this paper is to profile students with different study modes, the profiling being based on a structured questionnaire. In this way we try to find out what are the factors that affect the study mode. Our investigation focuses on both the effect of internal factors, such as motivation, and the effect of external factors, such as distance and time limitations, in the choice of a study method.

### cs4fn: A Flexible Model for Computer Science Outreach

*Chrystie Myketiak (Queen Mary University of London), Paul Curzon (Queen Mary University of London), Jonathan Black (Queen Mary University of London), Peter W. McOwan (Queen Mary University of London), Laura R. Meagher (UK Technology Development Group)*

There are a number of initiatives to attract secondary school students to computer science. cs4fn is one such project. It combines a magazine, website and live shows, telling stories about computer science in spirited and creative ways. Here we focus on the use of the magazine and, using sociolinguistic discourse analysis, we analyze comments from students and teachers to understand why they have requested (free) subscriptions to the magazine and how they plan to use it. Our analysis shows that both students and teachers are attracted to the flexibility that cs4fn provides, and use it in a variety of learning contexts. We find that the flexibility of the magazine makes it a valuable tool to engage students and teachers and that they use it to further enthuse others (i.e., other students and teachers). We suggest that cs4fn magazine is a powerful form of outreach that can be widely disseminated within computer science and other academic disciplines, raising the profile of computing to both students and teachers, and spreading enthusiasm for computer science.

**Anatomy, Dissection, and Mechanics of an Introductory Cyber-Security Class's Curriculum at the United States Naval Academy**

*Christopher Brown (US Naval Academy), Frederick Crabbe (US Naval Academy), Rita Doerr (US Naval Academy), Raymond Greenlaw (US Naval Academy), Chris Hoffmeister (US Naval Academy), Justin Monroe (US Naval Academy), Donald Needham (US Naval Academy), Andrew Phillips (US Naval Academy), Anthony Pollman (US Naval Academy), Stephen Schall (US Naval Academy), John Schultz (US Naval Academy), Steven Simon (US Naval Academy), David Stahl (US Naval Academy), Sarah Standard (US Naval Academy)*

Due to the high priority of cyber security education, the United States Naval Academy rapidly developed and implemented a new cyber-security course that is required for all of its first-year students. During the fall semester in 2011, half of the incoming class (about 600 students) took the course through a total of 31 sections offered by 16 instructors from a variety of disciplines and backgrounds. In the following spring semester, the remaining half of the first-year students will take the course. This paper explains the motivation that instigated and drove course development, the curriculum, teaching mechanics implemented, personnel required, as well as challenges and lessons learned from the first offering of the course. The information contained in this paper will be useful to those thinking of implementing a technical course required of all students at the same level in an institution (in our case first-year students) and particularly those interested in implementing such a course in cyber security.

**Fuzzy OOP: Expanded and Reduced Term Interpretations**

*Ronit Shmallo (SCE College of Engineering and Tel-Aviv University), Noa Ragonis (Beit Berl College and Technion ITT), David Ginat (Tel-Aviv University)*

We display a novel perspective of novice OOP difficulties. In a thorough study of 120 undergraduates, in their first OOP course, we noticed a variety of misconceptions and difficulties with a wide range of basic terms and notions. Careful analysis revealed a recurring phenomenon, of fuzzy OOP conceptions, which derive from expansion and reduction of basic term features. Novices tended to expand and/or reduce properties not only of terms such as "class" and "object", but also of terms such as "static", "access", and "instance". Particular misconceptions were of the forms: "One vs. Many". We display a thorough categorization of our findings, in an ordered hierarchical structure, and discuss its cognitive characteristics.

**Formal Learning Groups in an Introductory CS Course: A Qualitative Exploration**

*Julie Krause (Colorado School of Mines), Irene Polycarpou (Colorado School of Mines), Cyndi Rader (Colorado School of Mines)*

In this paper we present an exploratory study on students' learning experiences with formal learning groups in an introductory Computer Science course. To provide insights into ways to make learning groups productive in this setting, we report on the properties of learning groups that students indicated as more or less effective for various reasons. Additionally, we explore the impact of specific learning group characteristics such as the ratio of male to

female students, length of time in the group, and inclusion of students with varying levels of knowledge and experience.

**Competitive evaluation in a video game development course**

*Manuel Palomo-Duarte (University of Cadiz), Juan Manuel Dodero (University of Cadiz), Jose Tomas Tocino (University of Cadiz), Antonio Garcia-Dominguez (University of Cadiz), Antonio Balderas-Alberico (University of Cadiz)*

The importance of the video game and interactive entertainment industries has continuously increased in recent years. Academia has introduced computer game development in its curricula. In the case of the [SELF REFERENCE], a non-compulsory course about ``Video Game Design'' is offered to Computer Science students. In the artificial intelligence lesson, students play a serious game to earn part of their grade. Each student individually implements a strategy to play a predefined board game. The different strategies compete in a software environment that implements the game. Grading this part of the course depends on the results obtained in the competition. Additionally, before running the competition, students have to read the source code written by their classmates and bet a part of their grade on other teams' results. This way, they elaborate a critical analysis of the source code written by other programmers, a very valuable skill in the professional world. Students that are not so proficient in coding are still rewarded if they demonstrate they can do a good analysis of the source code written by others.

**User interface evaluation by novices**

*Dennis Bouvier (Southern Illinois University, Edwardsville), Tzu-Yi Chen (Pomona College), Gary Lewandowski (Xavier University), Robert McCartney (University of Connecticut), Kate Sanders (Rhode Island College), Tammy VanDeGrift (University of Portland)*

This study examines the extent to which novice computing students with minimal computer science coursework and no training in user interface (UI) evaluation consider UI concepts such as usability, user experience, and the context in which software will be used when evaluating an interface. In analyzing the responses of 149 novice computer science students who were asked to evaluate two interfaces for converting temperatures between Fahrenheit and Celsius, we observed that students generally considered usability and user experience factors, but were less likely to consider context. For educators, this exact task could be given to a class in order to initiate discussion of user-centered design; the study also provides a framework for structuring the discussion. More generally, the results of this study provide insight into some opportunities and challenges in teaching good interface design and evaluation skills.

**MyTuringTable: A Teaching Tool to Accompany Turing's Original Paper on Computability**

*Barry Fagin (US Air Force Academy), Dino Schweitzer (US Air Force Academy)*

MyTuringTable is a Turing Machine simulator designed to help students read and understand Turing's seminal 1937 paper on computability. We discuss our reasons for developing the tool, and report on its use in a "Great Ideas in Computer Science" course for the Air Force Academy Cadet Scholars program.

**Integrating the Teaching of Algorithmic Patterns into Computer Science Teacher Preparation Programs**

*Noa Ragonis (Beit Berl College and Technion ITT)*

The concept of patterns appears in the teaching of computer science in three main forms: algorithmic patterns, design patterns, and pedagogical patterns. A pattern is the identification of an abstract structure that can be further used in other, different contexts. This paper refers to algorithmic patterns as a tool that may be applied in computer science problem-solving processes, suggests an activity aimed at imparting algorithmic patterns to prospective computer science teachers, and presents part of an investigation of prospective teachers' understanding of algorithmic patterns. The main findings of the research support the inclusion of this topic in computer science teacher preparation programs, present the aspects of the concept for which the prospective teachers show meaningful understanding, and expose difficulties they experience when coping with patterns and composing recursion patterns. The focus is on abstract abilities demonstrated by computer science prospective teachers in the different stages of the activity.

**microPython: Non-Majors Programming from the Very First Lecture**

*John Aycock (University of Calgary)*

We wanted to give first-year non-major students experience programming very early, right from the first lecture. To support this endeavor, we built a web-based subset of Python, called microPython. It allowed immediate use by students, overcame a number of practical constraints, and gave a gradual introduction and transition into the full version of Python. Our data demonstrate that microPython has a miniscule impact on the server side, running easily on a desktop computer. A student survey shows an overwhelmingly positive response to programming in the first class; it also shows that a lot of students were using microPython to try examples from lecture, that they thought microPython was helpful for learning Python, and that they liked being able to write Python code within their browser.

**Engaging Computer Science in Traditional Education: The ECSITE Project**

*Debra Goldberg (University of Colorado Boulder), Dirk Grunwald (University of Colorado Boulder), Clayton Lewis (University of Colorado Boulder), Jessica Feld (University of Colorado Boulder), Sarah Hug (University of Colorado Boulder)*

The Engaging Computer Science in Traditional Education (ECSITE) Program is a 5-year program that began in 2009 to bring computer science into traditional K-12 classrooms. Rather than seeking to draw students into computing courses, we bring computing into the courses that students are already taking. To date, these have included art, biology, health education, mathematics, and social studies courses as well as a Native American focus program. Middle school and high school students are introduced to computational thinking and computer science concepts including algorithms, graph theory, and simulations in interdisciplinary contexts, mirroring the ways in which computing technologies are utilized in research and industry.

**A Systematic Approach to Teaching Abstraction and Mathematical Modeling**

*Chuck Cook (Clemson University), Svetlana Drachova (Clemson University), Jason Hallstrom (Clemson University), Joe Hollingsworth (Indiana University Southeast), David Jacobs (Clemson University), Joan Krone (Denison University), Murali Sitaraman (Clemson University)*

The need for undergraduate CS students to create and understand mathematical abstractions is clear, yet these skills are rarely taught in a systematic manner, if they are taught at all. This paper presents a systematic approach to teaching abstraction using rigorous mathematical models and a web-based reasoning environment. It contains a series of representative examples with varying levels of sophistication to make it possible to teach the ideas in a variety of courses, such as introductory programming, data structures, and software engineering. We also present results from our experimentation with these ideas over a 3-year period at our institution in a required course that introduces object-based software development, following CS2.

# 5. Working Groups

Four working groups will convene concurrently with the conference beginning on Sunday, July 1st. Each working group will present a work in progress report at the Wednesday 12:15 session.

### WG 1: Academic Integrity Policies: Addressing Best Practices in Education and Industry

*Leader: Charles P. Riedesel (University of Nebraska – Lincoln)*

This international working group of computing educators will develop a model departmental-level academic integrity policy that is compatible with best practices (both best educational practices and best practices in industry), and adaptable to a wide range of educational institutions, courses, and scenarios.

This will involve researching the current state of academic integrity policies and identifying best practices in preparation for updating and expanding a recent paper by this working group leader. (Riedesel C, Manley E, Poser S and Deogun J, A Model Ethics and integrity policy for Computer Science Departments, *Proc. of 40th SIGCSE Technical Symposium on Computer Science Education*, Chattanooga, 2009).

The resulting report will include a brief overview of the evolution of departmental academic integrity policies, a discussion on the impact of policy features on supporting best practices, an annotated model policy, and guidelines for developing, implementing, and assessing policies in terms of understandability, utility, comprehensiveness, adaptability, and effectiveness in simultaneously fostering ethical behavior and supporting best practices.

It is expected that a well-constructed academic integrity policy can foster an environment that is conducive toward educating students in the best practices of computing, and ideally promoting ethical behavior as well! It should be adaptable to particular class needs without needing to be frequently overridden, thereby providing more consistency. While many existing policies provide flexibility for instructors to apply their own collaboration, sharing, and Internet use specifications, the default is often rigid, yet uncomfortably vague, despite obvious attempts at improving clarity. The result is a mishmash of rules that vary by instructor and course, greatly lacking in consistency and quality. It may be more desirable to set the default to accommodate the more industry-friendly constraints of collaboration than to the most restrictive.

## WG 2: First Year Programming Projects for Computing for the Social Good

*Leader: Michael Goldweber, Xavier University, Cincinnati, OH USA*

Computing for the social good (CSG) is an umbrella term meant to incorporate any activity, from small to large, that endeavors to convey and reinforce computing's social relevance and potential for positive societal impact. This approach taps into reported students' desires to pick a major which they believe will allow them to have a positive societal impact; as opposed to the misunderstood belief that computing is boring, tedious, and irrelevant. The goal of this working group is to provide interested computer science instructors with concrete CSG-oriented activities. In particular, the activities will be programming projects suitable for use in the first year computing curriculum.
The plan for this working group is for each selected participant to create, prior to the conference proper, two CSG-based programming projects suitable for first year students. For example, one project might be a polymorphism project while the other may be centered around arrays. The ultimate goal is to provide the computer science education community with a resource enumerating real (class tested?) projects that instead of being centered around puzzles or games, illustrate computing's potential for positive societal impact.

## WG3: Teaching Modeling in Computing Curricula

*Leaders: Jürgen Börstler (Blekinge Institute of Technology, Sweden), Ludwik Kuzniarz (Blekinge Institute of Technology, Sweden), Carl Alphonce (University at Buffalo, USA)*

Modeling is a key skill any software developer needs to master to be able to solve complex problems. The development and proper usage of conceptual models, UML diagrams, or workflows are important learning objectives in any computer science related curriculum and recognized in several ACM/IEEE curriculum guidelines. However, there is very little systematic work on the role of modeling in CS/SE education beyond OOD or particular modeling languages and tools.
In this working group, we want to investigate and map the current state-of-the-art in the area of the teaching of software modeling in CS/SE curricula. The area will be explored from three different points of view: (1) state-of-practice, (2) curriculum recommendations, and (3) educators' experiences. Working group participants are expected to collect data from typical courses that involve some form of teaching modeling and carry out at least 2 (semi-structured) interviews with teachers. Furthermore, each participant is expected to review one example of a common curriculum recommendation (like e.g., SE2004) for specific recommendations related to modeling. The WG co-organizers will draft interview scripts and data collection forms, but all WG members are expected to actively participate in completing them. Please note that we are primarily interested in conceptual modeling, domain modeling, software design, etc. Mathematical modeling/simulation or modeling in science/engineering in general is out of the scope of the working group.

**WG4: Educating for Mobile Computing: Addressing the New Challenges**

*Leader: Barry Burd (Drew University)*

Mobile computing is exploding into the consumer space. This trend has some important implications for Computer Science education:

- Students carry processors in their pockets, and use computers (of one kind or another) all day long.
- The job market for computer professionals is shifting.
- The principles for making optimal use of hardware, and for creating good user interfaces, require refinement.
- Hardware requirements in computer science courses are changing.

Several colleges are offering degrees in mobile computing. Others offer regular courses or occasional topics courses. Still others integrate mobile computing into existing Computer Science courses.

In this working group, we investigate the impact of mobile computing on all aspects of the Computer Science curriculum. The working group's goal is to compile a list of resources, and to make recommendations for Computer Science programs of various sizes (small private institutions, large public institutions, secondary schools, and so on).

To prepare for our meeting at ITiCSE 2012, we will collect information (curriculum documents, anecdotes, and so on) describing current practices in mobile computing instruction. Using this information we will create a list of issues to be discussed and questions to be addressed during the July meeting.

Working group participants should have experience or interest in teaching mobile computing or closely related topics. Participants should also be open to new ideas about instruction in mobile computing. Ideally, the working group members bring a cross section of knowledge and experience, including

- Use of mobile computing as a learning context,
- Models of integrating mobile computing throughout curriculum,
- The use of different platforms (Android, iOS, Windows Phone, and others),
- Familiarity with usage patterns in different countries and different cultures,
- Interests in UI design as well as application programming, and
- Experience with all aspects of the mobile development life cycle.

# 6. Israeli High-School students' projects

Amit Zandman, Harel High School, Mevaseret Zion. Project: **Computer Graphics - Nuclear Reactor Simulation**. Teachers and mentors: Memi Gutbie, Reuven Bodenhimer.

Tal Sabag and Tomer Shvadron, Gymnasia Realit High School, Rishon LeZion. Project: **Artificial Intelligence - Intelligent Board Games**. Teacher and mentor: Ela Lev.

Bar Zach, Atid High School for Science, Lod. Project: **Operating Systems - Day Organizer**. Teacher and mentor: Alla Mayanovsky.

Daniel Kashi, Ohel-Shem High School, Ramat Gan. Project: **Web Services - Cinema World**. Teacher and mentor: Iris Zur Barguri.

Maor Carmi, LeoBaeck High-School, Haifa. Project: **Machine learning - Computer Learns to Play Pentago**. Teacher and mentor: Ahuva Sperling.

Sleem Galock, Hashalom High School, Maza'a. Project: **Mobile Phone Application Development - Bluetooth Remote control**. Teacher and mentor: Samich Abbas.

Yonathan Amir, Democratic School of Kfar Saba, Kfar Saba. Project: **Computer Graphics and Graph Theory – Pathfinding**. Mentor: Vladimir Nodelman, HIT. Teacher: Galit Sluzki.

Elad Gershon, Atid Raziel High School, Herzeliya. Project: **Computer Graphics and Game Development – Pick-UP Stick 3D Game**. Mentor: Michael Chernovilsky, Netanya College. Teacher: Asaf Amir.

Miriam Botwinik, Amit Renanim High School, Ra'anana. Project: **Image Processing - Recognition of Geometric Shapes**. Mentor: Yulia Kempner, HIT, Teacher: Irit Herman.

Liran Mesika, Herzog High School, Kfar Saba. Project: **Logic Puzzels – The Griddlers Game**. Mentor: Dan Ophir, The University Center Ariel. Teacher: Ziva Kuntzman.

Nevo Himmelhoch, Dror Kiryat Hinuch School, Bney Dror. Project: **Communication Networks - Controlling Computers Through Email.** Mentor: Gilad Tsur, Weizmann Institute of Science. Teacher: Leah Geva

Tali Elias, Ironi Yud Dalet High School, Tel Aviv. Project: **Image Processing - X-Ray Image Processing**. Mentor: Dror Malka, HIT and Bar-Ilan University. Teacher: Leah Brur.

# 7. Space for your notes!

# 8. Social Events

## 8.1 Lunches/Coffee Breaks and Poster Sessions

All conference coffee breaks and poster sessions will take place in the foyer area near the **Segoe Auditorium**.

Lunch on Wednesday and Thursday will be served in the Student Union building (in "HaUlam HaShakuf"). Please remember to **bring the lunch ticket with you!**

## 8.2 Monday Evening Reception

There will be an informal welcome reception for all conference participants and their guests, starting at 18:00 on Monday, July 2. The reception will be held at **MAKAM**, 7 Gotlib Shumacher St., close to the Colony Hotel in the German Colony. For information on Shuttles to the reception please see page 52.

## 8.3 Conference Banquet

The conference banquet will be held on Wednesday, July 4, starting at 19.00, at the Palm Beach Hotel, Acre. The Banquet will begin outside, on the hotel lawn, where we will have a chance to enjoy the sunset over the Haifa Bay. In the garden there will be food stalls with a wide variety of dishes. Later, we will enter the dining room for the "real" dinner. During dinner, a group of musicians will play for us International, Jewish and Israeli music.

The buses to the Banquet will leave from the conference hotels at (see page 52 for details). Please arrive on time. If you prefer to skip the banquet or arrange private transportation, let us know of this beforehand so we won't have to look for you or hold the buses.

Please make sure that you **bring your banquet ticket** with you, as well as (if applicable) the banquet ticket(s) for your accompanying person(s).

## 8.4 Excursions

On Tuesday morning, three excursions (provided by Amiel Tours) will leave from the conference hotels: The Sea of Galilee Tour, The Nazareth and Kana Tour, The Acre Tour.

All the tours include lunch. All the tours will return to the conference hotels around 13:30, which will give you some time to rest before the opening of the conference.

# 9. Essentials

## 9.1 Emergencies

Hopefully, there will be no emergencies! If there are, we have a contact person in each of the conference hotels: at the Theodor Hotel (and Art Gallery), your contact person is Michael Caspersen; at the Colony Hotel, your contact person is Judith Gal-Ezer; at the Dan Panorama, your contact person is Tami Lapidot.

Some useful numbers: Police 100, Magen David Adom (medical emergency services) 101, Fire Department 102.

Egged buses information center 03-6948888

Train information center 03-6117000 / *5770

Ben Gurion Airport information center 03-9755555 / *6663

## 9.2 Getting around and transportation

Haifa is a large city. Please use the map in your conference bag to explore the city.
While the conference venue is the Technion campus, most ITiCSE 2012 attendees are staying in three areas of the city: the German Colony (the Colony Hotel, Port Inn), Hadar (the Theodor Hotel, Art Gallery Hotel, Lui), and on Mount Carmel (Dan Panorama Hotel, Har HaCarmel Hotel, Crowne Plaza). Free shuttles will be available from the conference hotels to the Technion during the conference.

**Theodor Hotel**

| Sunday | Monday | Tuesday | Wednesday | Thursday |
|---|---|---|---|---|
| 8:30<br>to Technion<br>18:00<br>from Technion | 8:30<br>to Technion<br>18:00<br>to reception | 15:20<br>to Technion<br>19:30<br>from Technion | 8:30<br>to Technion<br>16:30<br>from Technion<br>18:15<br>to Banquet | 8:30<br>to Technion<br>18:30<br>from Technion |

If you miss the bus or want to come on your own, take the number 19 bus or taxi service 19.
Sun, Mon: Get off at the 4th station and cross the street.
Tues, Wed, Thu: Get off at the first station on the Technion campus and walk down to the conference area.

**Colony Hotel**

| Sunday | Monday | Tuesday | Wednesday | Thursday |
|---|---|---|---|---|
| 8:20<br>to Technion<br>18:00<br>from Technion | 8:20<br>to Technion | 15:15<br>to Technion<br>19:30<br>from Technion | 8:15<br>to Technion<br>16:30<br>from Technion<br>18:15<br>to Banquet | 8:15<br>to Technion<br>18:30<br>from Technion |

If you miss the bus or want to come on your own, take the number 17 bus.
Sun, Mon: Get off at the 4th station and cross the street
Tues, Wed, Thu: Get off at the first station on the Technion campus and walk down to the conference area.

**Dan Panorama Hotel**

| Sunday | Monday | Tuesday | Wednesday | Thursday |
|---|---|---|---|---|
| 8:30<br>to Technion<br>18:00<br>from Technion | 8:30<br>to Technion<br>17:30<br>to reception | 15:20<br>to Technion<br>19:30<br>from Technion | 8:20<br>to Technion<br>16:30<br>from Technion<br>18:00<br>to Banquet | 8:20<br>to Technion<br>18:30<br>from Technion |

If you miss the bus or want to come on your own, take the number 31 bus (to the Technion) or the number 28 bus (to Ziv), and from there you can walk or take any bus or service taxi to Technion.
Sun, Mon: Get off at the 4th station and cross the street
Tues, Wed, Thu: Get off at the first station on the Technion campus and walk down to the conference area.

**Technion**

| Sunday | Monday | Tuesday | Wednesday | Thursday |
|---|---|---|---|---|
| | 18:00<br>to reception | | 18:20<br>to Banquet | |

## 9.3 Security

In some public places you may be asked to open your bags. Please be patient with the guard – it's for your own security.
Pickpockets and robbers are by no means common in Haifa, but—as in any other town—they do exist. Keep an eye on your belongings, especially your wallet. The more crowded a place is, the more likely it is to be a place for a pickpocket.

## 9.4 Computer Access

The Technion campus provides wireless access in many areas, including the Segoe building, the Student Union building, Butler Hall, the foyer of the Computer Science department (first and second floor), as well as the open area in front of the Segoe building (between the Student Union and the Computer Science department).

The Technion public network is called **TechPublic** and you can login free of charge.

## 9.5 Restaurants

There are several areas in Haifa with a wide variety of restaurants, cafés and pubs:

- The German Colony (Sderot Ben Gurion St.)
- The Carmel center (Sderot Moria, Sderot Hanassi)
- The promenade near the sea
- Shopping malls (Grand Kenyon, Kenyon Haifa, Castra Center, Horev Center, Merkaz Panorama)

**Recommended Chef's restaurants:**

Carola (Italian kitchen), Sderot Moria 64
Ha-Slik (bar & restaurant), Sderot Hanasi 124
Voila, Sderot Hanasi 119
Luiz (health food), Sderot Moria 58
HaNamal 24, Hanamal st. 24

**Recommended fish & seafood restaurants**:
Shrimp House, Hahagana 105
Jacko, Sderot Moria 11
Calamaris, Stela Maris 100 (with a magnificent view of the bay)

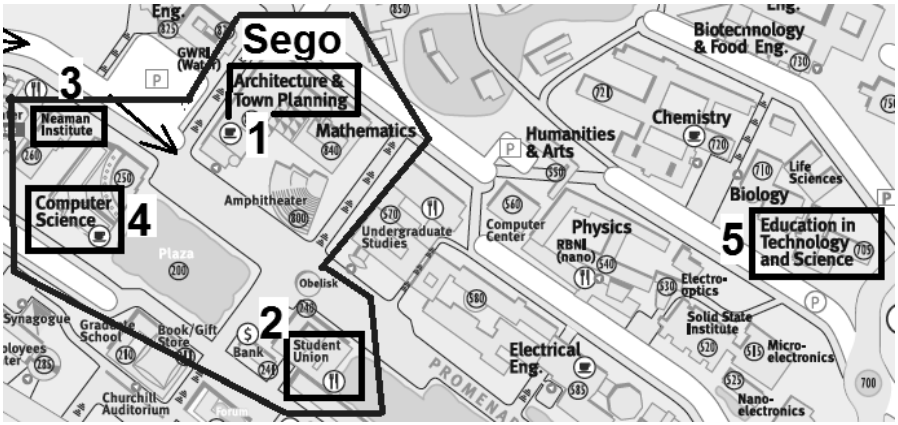
**Recommended Meat restaurants:**
Sinta Bar, Sderot Moria 127
Rak Basar, Sderot Ben Gurion 15 (German Colony)
Basarela, Fliman 8 (Castra Center)

**Recommended Asian food restaurants:**
Japanika (Japanese, Asian food), Sderot hanasi 122
Mina Tomei (Asian, Indian, Japanese, Korean, Thai), Fliman 8 (Castra center)
Giraffe (Asian), Sderot Hanasi 131

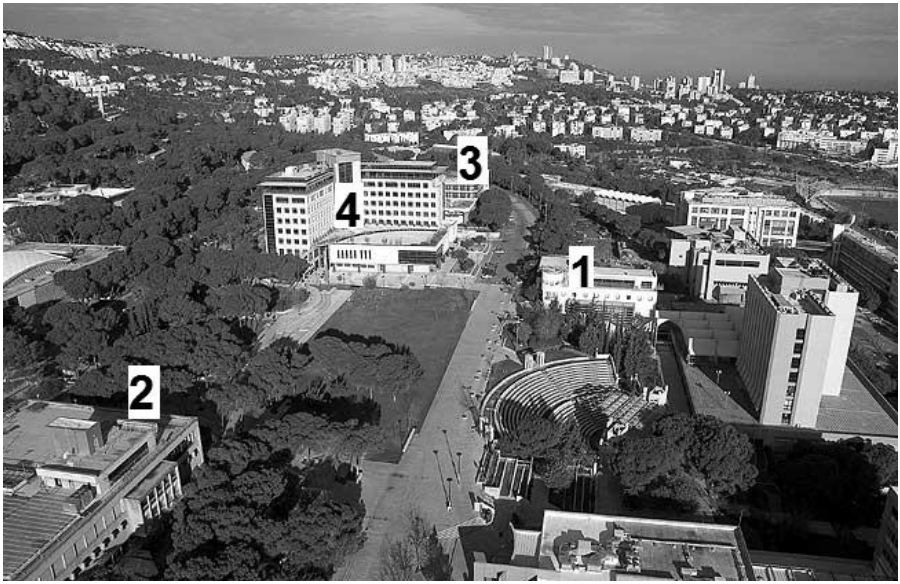
**Recommended cafés:**
Shani, Trumpeldor 29 (close to the Technion), Sderot Moria 29
Greg, 18 branches in Haifa (in the Technion - CS building, Sderot Moria 105)
Aroma, 6 branches in Haifa (Sderot Hanasi 121)

**Recommended fast food restaurants:**
Pizza Panorama, Halevanon 6
Shawarma Ktana Vematrifa, Gat 4 (near the Horev Center Mall) – excellent falafel and shawarma
Burekas Ha-agala, Sderot Moria 52
Agadir, Pinsker 6 (near Sderot Moria), excellent hamburgers
Khaled Adaui, near the Technion (back entrance)
Angos, Hamaginim 27 (oriental)
Abu Yusuf, Kikar Paris 1

# 10. Conference buildings




1 – Segoe (Architecture) ground floor – the main conference venue
2 – Students' Union (Lunch, and the Cinema Hall)
3 – Samuel Neaman Institute (Butler Auditorium)
4 – Computer Science (room 337)



# 11. Conference at a Glance

| | Tuesday | Wednesday | Thursday |
|---|---|---|---|
| 9:00 | Excursions | Keynote: Lenore Blum | Keynote: David Harel |
| 10:00 | | Coffee Break & Posters I (Segoe) | Short Break |
| 10:15 | | | TT&C II (Segoe), P8 (Cinema), P9 (Butler) |
| 10:45 | | Panel II (Segoe), P3 (Cinema), P4 (CS) | |
| 11:30 | | | Short Break |
| 11:45 | | | Papers P10 (Segoe), P11 (Cinema), P12 (Butler) |
| 12:00 | | Short Break | |
| 12:15 | | WGs report (Segoe), P5 (Cinema), P6 (CS) | |
| 13:00 | | | Lunch |
| 13:30 | | Lunch | |
| 14:00 | | Panel III (Segoe), TT&C I (Cinema), P7 (CS) | Papers P13 (Segoe), P14 (Cinema), P15 (Butler) |
| 14:30 | | | |
| 15:15 | | | Short Break |
| 15:30 | | | Papers P16 (Segoe), P17 (Cinema), P18 (Butler) |
| 15:45 | | Coffee Break & Posters II (Segoe) | |
| 16:00 | Welcome & Opening Session (Segoe) | Buses to hotels & short break in hotels | |
| 16:30 | | | |
| 17:10 | Keynote: Michael Rabin | | Coffee Break (Segoe) |
| 17:30 | Coffee break & demonstration of Israeli high-school students' projects (Segoe) | Buses to the Banquet in Acre | Closing Session |
| 18:00 | | | |
| 18:15 | Panel (Segoe), P1 (Cinema), P2 (Butler) | | |
| 18:30 | | | Buses to hotels |
| 19:00 | | Banquet | |
| 19:30 | Buses to hotels | | |
| 23:00 | | Buses to Haifa hotels | |