

ERIC S. ROBERTS
Curriculum Vitae

WORK ADDRESS

Department of Computer Science
Stanford University
Stanford, CA 94305-9015
(650) 723-3642
eroberts@cs.stanford.edu

PRINCIPAL AREAS OF EXPERTISE

Computer science education, programming languages, programming environments, multiprocessor systems, social implications of computing, computing and public policy.

CURRENT POSITION

Charles Simonyi Professor of Computer Science, *emeritus*, Stanford University. Originally appointed to the Stanford faculty as Associate Professor in 1990, promoted to full professor in 1995, and named the Charles Simonyi Professor in 1997. Retired in September 2015 and recalled to active duty for 2015-16 and 2016-17.

Bass University Fellow in Undergraduate Education, Stanford University, 2002-present.

EDUCATION

Ph.D., Applied Mathematics, Harvard University, June 1980. Dissertation: *Software Techniques for Practical Multiprocessors*, Advisor: Thomas E. Cheatham.

S.M., Applied Mathematics, Harvard University, June 1974.

A.B., *cum laude*, Applied Mathematics, Harvard University, June 1973.

ACADEMIC EXPERIENCE

Stanford University

Faculty Director for Interdisciplinary Science Education, Office of the Vice Provost for Undergraduate Education, 2003-06.

Chair of the Faculty Senate, 2005-06.

Senior Associate Dean for Student Affairs, School of Engineering, Stanford University, 2001-03. Responsible for the academic programs in the eight departments that make up the School of Engineering.

Associate Chair and Director of Undergraduate Studies, Department of Computer Science, 1990-2002.

Swarthmore College

Eugene Lang Visiting Professor for Social Change, January to June, 2000. Taught and wrote about computing technology and its influence on society.

Harvard University

Visiting Lecturer in Computer Science, 1984-85. Taught Computer Science 163: Theory and Construction of Compilers and began writing my first textbook, *Thinking Recursively*.

Wellesley College

Assistant Professor of Computer Science, 1980-85 (on leave 1983-85). As Wellesley's first full-time faculty member in computer science, had the principal responsibility for designing new courses, developing curricular materials, advising the College on the integration of computer science into the overall program, and setting the foundation necessary to establish computer science as an autonomous department. Taught courses in introductory programming, data structures, programming languages, assembly language programming, and operating systems. Also served as department chair in 1982-83.

BOOKS AND BOOK CHAPTERS

The Intellectual Excitement of Computer Science, Pearson, forthcoming. Written for a broad college-educated audience curious about what makes computer science interesting on an intellectual level.

Understanding Programming through JavaScript, Pearson, 2018. Presents the material in a traditional CS1 introductory course using JavaScript, which is the standard programming language for web interactivity. The book does not cover all of JavaScript but instead focuses on what Douglas Crockford, author of *JavaScript: The Good Parts*, identifies as the “beautiful, elegant, highly expressive language” that constitutes the core of the language.

Programming Abstractions in Java, Pearson, 2016. Covers the material in a traditional CS2 course on data structures and algorithms using the Java programming language. This book, like its C++ counterpart, adopts a “client-first” approach in which students begin by using the various collection classes as clients.

Programming Abstractions in C++, Pearson, 2014. Covers the material in a traditional CS2 course on data structures and algorithms using the C++ programming language. The text adopts a “client-first” approach in which students begin by using the various collection classes as clients; once students know how to use these structures, they are in a much better position to understand issues of implementation and algorithmic performance. This book has been translated into Chinese.

The Art and Science of Java, Addison-Wesley, 2008. An introductory Java text with much the same overall flavor as my 1995 text, *The Art and Science of C*. This book has been translated into Chinese.

Thinking Recursively with Java, John Wiley and Sons, 2006. A twentieth-anniversary update of my 1986 text, *Thinking Recursively*. The new version uses Java in place of the original Pascal.

“Computers and society,” in *Encyclopedia of Computer Science* (fourth edition), Anthony Ralston, Edwin Reilly, and David Hemmendinger (editors), Grove’s Dictionaries, 2000.

Programming Abstractions in C, Addison-Wesley, 1998. A companion volume to *The Art and Science of C* that covers the material in the standard CS2 course. The text offers a classical introduction to data structures and algorithms, with a strong focus on the use of abstraction and encapsulation. This text has been translated into Modern Greek and Chinese.

The Art and Science of C: A Library-Based Introduction to Computer Science, Addison-Wesley, 1995. A general introduction to programming that covers the material in the CS1 course described in the ACM Curriculum ’78 document. The text uses ANSI C as its programming language, limiting the complexity of the presentation by using library packages to defer consideration of complex topics until students are able to assimilate that material. The book emphasizes the development of good software engineering methodology and effective strategies for modular decomposition and data abstraction. This text has been translated into Modern Greek and Chinese.

“Computers and society,” in *Encyclopedia of Computer Science* (third edition), Anthony Ralston and Edwin Reilly (editors), Van Nostrand Reinhold, 1992.

“Computers and the Strategic Defense Initiative” with Steve Berlin, included as Chapter 8 of *Computers in Battle: Will They Work?*, David Bellin and Gary Chapman (editors), Harcourt Brace Jovanovich, 1987.

Thinking Recursively, John Wiley and Sons, 1986. A sophomore-level text that gives students a strategic overview of how to approach recursive programming problems. It has been translated into German, Italian, and Japanese.

ARTICLES AND REPORTS

Assessing and Responding to the Growth of Computer Science Undergraduate Enrollments (prepublication release), Committee on the Growth of Computer Science Undergraduate Enrollments, Washington, DC: National Academies Press, October 2017.

<https://www.nap.edu/catalog/24926>

“A history of capacity challenges in computer science,” position paper for the National Academies Committee on the Growth of Computer Science Undergraduate Enrollments, March 2016.

<https://cs.stanford.edu/~eroberts/CSCapacity/>

“A portable graphics library for introductory CS” with Keith Schwarz, *Proceedings of the SIGCSE Conference on Innovation and Technology in Computer Science Education*, Canterbury, England, July 2013.

<http://cs.stanford.edu/~eroberts/papers/PortableGraphicsLibrary.pdf>

“Meeting the challenges of rising enrollments,” *Inroads*, September 2011.

<http://doi.acm.org/10.1145/2003616.2003617>

“Informatics education using nothing but a browser” with Chris Piech, *Proceedings of the IFIP Conference on ICT and Informatics in a Globalised World of Education*, Mombasa, Kenya, August 2011.

<http://cs.stanford.edu/~eroberts/papers/NothingButABrowser.pdf>

“Point/counterpoint: Technology education for the early 21st century,” debate with Stephen J. Andriole, *Communications of the ACM*, July 2008.

<http://doi.acm.org/10.1145/1364782.1364792>

“Resurrecting the applet paradigm,” *Proceedings of the Thirty-eighth SIGCSE Technical Symposium on Computer Science Education*, Covington, Kentucky, March 2007.

<http://doi.acm.org/10.1145/1227310.1227488>

“Sparking self-sustained learning: Report on a design experiment to build technological fluency and bridge divides” with Brigid Barron and Caitlin Kennedy Martin, *International Journal of Technology and Design Education*, 17:1, January 2007.

<http://cs.stanford.edu/~eroberts/papers/SelfSustainedLearning.pdf>

“An interactive tutorial for Java,” *Proceedings of the Thirty-seventh SIGCSE Technical Symposium on Computer Science Education*, Houston, Texas, February 2006.

<http://doi.acm.org/10.1145/1121341.1121447>

“Designing a computer science curriculum for Bermuda’s public schools” with Brigid Barron and Caitlin Kennedy Martin, *World Conference on Computers in Education*, Capetown, South Africa, July 2005.

<http://cs.stanford.edu/~eroberts/papers/CSCurriculumForBermuda.pdf>

“What can computer science learn from a fine-arts approach to teaching?” with Lecia Barker and Kathy Garvin-Doxas, *Proceedings of the Thirty-sixth SIGCSE Technical Symposium on Computer Science Education*, St. Louis, Missouri, February 2005.

<http://doi.acm.org/10.1145/1047344.1047482>

“Imagining possible futures: Course taking and knowledge use within trajectories of technological fluency” with Brigid Barron, Caitlin K. Martin, Emma M. Mercier, and Sara McPhee, *Proceedings of the American Educational Research Association Annual Meeting*, San Diego, California, April 2004.

<http://cs.stanford.edu/~eroberts/papers/ImaginingPossibleFutures.pdf>

“The dream of a common language: The search for simplicity and stability in computer science education,” *Proceedings of the Thirty-fifth SIGCSE Technical Symposium on Computer Science Education*, Norfolk, Virginia, March 2004.

<http://doi.acm.org/10.1145/971300.971343>

“Strategies for promoting academic integrity in computer science courses,” *Proceedings of the 2002 Frontiers in Education Conference*, Boston, Massachusetts, November 2002.

<http://doi.ieeecomputersociety.org/10.1109/FIE.2002.1158209>

“Designing and assessing ongoing professional development: Opportunities for high school computer science teachers” with Caitlin Kennedy Martin and Brigid Barron, *Proceedings of the Fifth International Conference of the Learning Sciences*, Seattle, Washington, October 2002.

<http://cs.stanford.edu/~eroberts/papers/ProfessionalDevelopment.pdf>

“A design experiment to build technological fluency and bridge divides” with Caitlin Kennedy Martin and Brigid Barron, *Proceedings of the Fifth International Conference of the Learning Sciences*, Seattle, Washington, October 2002.

<http://cs.stanford.edu/~eroberts/papers/DesignExperiment.pdf>

“Encouraging women in computer science” with Marina Kassianidou and Lilly Irani, *Inroads*, June 2002.

<http://doi.acm.org/10.1145/543812.543837>

“Design experiments at a distance: Lessons from developing a secondary school curriculum for Bermuda public schools” with Brigid Barron, Caitlin Kennedy Martin, Alex Osipovich, and Michael Ross, *Proceedings of the Computer Support for Collaborative Learning (CSCL) Conference*, Boulder, Colorado, January 2002.

<http://cs.stanford.edu/~eroberts/papers/TechnologicalFluency.pdf>

Computing Curricula 2001, edited by Eric Roberts and Gerald Engel, IEEE Computer Society Press, December 2001.

<http://doi.acm.org/10.1145/384274.384275>

“An overview of MiniJava,” *Proceedings of the Thirty-second SIGCSE Technical Symposium on Computer Science Education*, Charlotte, North Carolina, February 2001.

<http://doi.acm.org/10.1145/364447.364525>

“Computing education and the information technology workforce,” prepared for the National Research Council Study on Workforce Needs in Information Technology with the endorsement and support of the Education Board of the Association of Computing Machinery, reprinted in *SIGCSE Bulletin*, June 2000.

<http://doi.acm.org/10.1145/355354.355383>

“Strategies for encouraging individual achievement in introductory computer science courses,” *Proceedings of the Thirty-first SIGCSE Technical Symposium on Computer Science Education*, Austin, Texas, March 2000.

<http://doi.acm.org/10.1145/330908.331873>

“Conserving the seed corn: Reflections on the academic hiring crisis,” *SIGCSE Bulletin*, December 1999.

<http://doi.acm.org/10.1145/349316.349363>

“Curricula 2001: Bringing the future to the classroom” with Carl K. Chang, Gerald Engel, Willis King, Russ Shackelford, Robert H. Sloan, and Pradip K. Srimani, *IEEE Computer*, September 1999. Also published in *SIGCSE Bulletin*, December 1999.

<http://doi.acm.org/10.1145/349522.349403>

“Strategies for integrating technology into computer ethics courses,” *Proceedings of the SIGCSE/SIGCUE Joint Conference on Integrating Technology in Computer Science Education*, Dublin, Ireland, August 1998.

<http://doi.acm.org/10.1145/282991.283127>

“Designing a Java graphics library for CS1” with Antoine Picard and Maria Fredricsson, *Proceedings of the SIGCSE/SIGCUE Joint Conference on Integrating Technology in Computer Science Education*, Dublin, Ireland, August 1998.

<http://doi.acm.org/10.1145/290320.283129>

“Directions in computer science education,” published electronically in conjunction with the December 1996 issue of *Computing Surveys* as part of an invited collection of position papers.

<http://doi.acm.org/10.1145/242224.242357>

“Tools for creating portable demonstration programs,” *Proceedings of the SIGCSE/SIGCUE Joint Conference on Integrating Technology in Computer Science Education*, Barcelona, Spain, June 1996.

<http://doi.acm.org/10.1145/237477.237539>

“Thetis: An ANSI C programming environment designed for introductory use” with Steve Freund, *Proceedings of the Twenty-seventh SIGCSE Technical Symposium on Computer Science Education*, Philadelphia, Pennsylvania, February 1996.

<http://doi.acm.org/10.1145/236462.236560>

“A C-based graphics library for CS1,” *Proceedings of the Twenty-sixth SIGCSE Technical Symposium on Computer Science Education*, Nashville, Tennessee, March 1995.

<http://doi.acm.org/10.1145/199691.199767>

“Loop exits and structured programming: Reopening the debate,” *Proceedings of the Twenty-sixth SIGCSE Technical Symposium on Computer Science Education*, Nashville, Tennessee, March 1995.

<http://doi.acm.org/10.1145/199691.199815>

“Using undergraduates as teaching assistants in introductory programming courses: an update on the Stanford experience” with John Lilly and Bryan Rollins, *Proceedings of the Twenty-sixth SIGCSE Technical Symposium on Computer Science Education*, Nashville, Tennessee, March 1995.

<http://doi.acm.org/10.1145/199691.199716>

“Using C in CS1: The Stanford experience,” *Proceedings of the Twenty-fourth SIGCSE Technical Symposium on Computer Science Education*, Indianapolis, Indiana, February 1993.

<http://doi.acm.org/10.1145/169073.169361>

“Factors working against women in computer science,” *Tough Questions* (a publication of Student Pugwash USA), fall 1989.

“WorkCrews: An abstraction for controlling parallelism” with Mark Vandevoorde, Research Report #42, Digital Equipment Corporation Systems Research Center, Palo Alto, California, April 1989. Also published in the *International Journal on Parallel Programming*, Volume 17, Number 4, 1988.

“Implementing exceptions in C,” Research Report #40, Digital Equipment Corporation Systems Research Center, Palo Alto, California, March 1989.

“Computing implications: Report from DIAC-88” with Douglas Schuler, *Computers and Society*, ACM SIGCAS, Volume 19, Number 1, March 1989. An earlier version of this paper appeared in *Abacus* magazine, summer 1988.

“**parmake** and **dp**: Experience with a distributed, parallel implementation of make” with John Ellis, *Proceedings from the Second Workshop on Large-Grained Parallelism*, Software Engineering Institute, Carnegie-Mellon University, Report CMU/SEI-87-SR-5, November 1987.

“Programming and the Pentagon,” *Abacus* magazine, summer 1987.

“The Eastport Report: Unexpected support for SDI critics,” *The CPSR Newsletter*, Volume 4, Number 3, Summer 1986.

Elements of Basic Programming: An Introduction to Algorithmic Computation, Department of Computer Science, Wellesley College, Wellesley, MA. This draft textbook was used from 1981-87 for the introductory computer science course at Wellesley.

“Task management in Ada: A critical evaluation for real-time multiprocessors” with Arthur Evans Jr., C. Robert Morgan, and Edmund Clarke, *Software—Practice and Experience*, Volume 11, October 1981.

Software Techniques for Practical Multiprocessors (Ph.D. thesis), Technical Report TR-08-81, Center for Research in Computing Technology, Harvard University, 1981.

“The impact of multiprocessor technology on high-level language design” with Arthur Evans Jr., C. Robert Morgan, and Edmund Clarke, Report No. 4188, Bolt Beranek and Newman Inc., September 1979.

“Pluribus: An operational fault-tolerant multiprocessor” with David Katsuki, Eric Elsam, William Mann, John Robinson, F. Stanley Skowronski, and Eric Wolf, *Proceedings of the IEEE*, Volume 66, Number 10, October 1978. Reprinted in *Advances in Computer Communications and Networking*, Wesley Chu (editor), Artech House, Dedham, Massachusetts, 1979. Also reprinted in *The Theory and Practice of Reliable System Design* by Daniel P. Siewiorek and Robert W. Swarz, Digital Press, Billerica, Massachusetts, 1982.

“Software fault-tolerance in the Pluribus” with John Robinson, *Proceedings of the 1978 National Computer Conference*, June 1978.

INVITED TALKS AND PRESENTATIONS

“NAS project: Growth of computer science undergraduate enrollments” with Jodi Tims, presentation at the CMU CS Education Summit, Pittsburgh, Pennsylvania, October 2017.

“Enrollment booms and gender diversity: How do we keep history from repeating itself?”, panel presentation at Grace Hopper 2016, Houston, Texas, October 2016.

“Computer science at Wellesley: Early memories,” Cirque du CS celebration of the 32nd anniversary of the Wellesley Computer Science Department, Wellesley, Massachusetts, March 2015.

“Tsunami or sea change? Responding to the explosion of student interest in computer science” with Ed Lazowska and Jim Kurose, National Center for Women in Technology 10th Anniversary Summit, Newport Beach, California, May 2014, and the CRA Snowbird Conference, Snowbird, Utah, July 2014.

“CS+X: Building interdisciplinary bridges at Stanford University,” Harvard University, Cambridge, Massachusetts, May 2014.

“Zen and the art of debugging,” Reed College Math Colloquium, Portland, Oregon, April 2014.

“What happens when we end up teaching all students computing,” NSF Future Directions in CS Education, Orlando, Florida, January 2014.

“Computer science at Stanford: Strategies for a time of unprecedented growth,” Stanford Club of Great Britain, London UK, July 2016.

“The impact of computers on our world: The view from Stanford computer science,” Stanford Club for European Affairs, Paris, France, June 2013.

“Nifty Assignments: Turtle graphics,” panel participant at the Forty-fourth SIGCSE Technical Symposium on Computer Science Education, Denver, Colorado, March 2013.

“Bridging the gender gap: Lessons from the liberal arts,” panel participant at the Grace Hopper Celebration of Women in Computing, Portland, Oregon, November 2011.

“Computer science as an essential 21st-century skill: Making the case,” opening keynote address, Grace Hopper Celebration of Women in Computing: K-12 Workshop, Atlanta, Georgia, October 2010.

“Converting Java into JavaScript,” Google Tech Talk, Atlanta, Georgia, October 2010.

“Recognizing the most influential CS papers,” panel participant at the Forty-first SIGCSE Technical Symposium on Computer Science Education, Milwaukee, Wisconsin, March 2010.

“What journalists need to know about computing,” Knight Fellowship Program, Stanford University, Stanford, California, January 2010.

“Restoring the passion, beauty, joy, and awe,” departmental colloquium, Department of Computer Science, University of Auckland, Auckland, New Zealand, July 2009.

“Restoring the passion, beauty, joy, and awe,” panel participant for the National Advisory Commission on Computing Qualifications (NACCQ) 2009, Napier, New Zealand, July 2009.

“Why programming matters,” keynote address for the National Advisory Commission on Computing Qualifications (NACCCQ) 2009, Napier, New Zealand, July 2009.

“Passion, beauty, joy, and awe, continued,” panel participant at the Fortieth SIGCSE Technical Symposium on Computer Science Education, Chattanooga, Tennessee, March 2009.

“Computational thinking everywhere: The ‘broad audience for CS1’ approach,” National Academies Computational Thinking Workshop, Washington, DC, February 2009.

“Computing and competitiveness: Implications of the programmer shortage,” Center for International Security and Cooperation, Stanford, California, February 2009.

“Programming and the CS curriculum: The more things change . . .”, Fascinating Lectures in Computer Science series, Swarthmore, Pennsylvania, November 2008.

“Rediscovering the passion, beauty, joy, and awe: Making computing fun again,” keynote address, Liverpool, England, August 2008.

“Designing a modern computing curriculum for Bermuda,” Reed College Mathematics Colloquium, Portland, Oregon, April 2008.

“Expanding the pipeline of students in computer science” with Mehran Sahami, plenary address, Stanford Computer Forum, Stanford, California, March 2008.

“Rediscovering the passion, beauty, joy, and awe: Making computing fun again,” panel participant at the Thirty-ninth SIGCSE Technical Symposium on Computer Science Education, Portland, Oregon, March 2008.

“Making computer science fun again,” opening keynote address at Informatics Education Europe II, Thessaloniki, Greece, November 2007.

“The current crisis in computing: What are the real issues?”, panel participant at the Thirty-eighth SIGCSE Technical Symposium on Computer Science Education, Covington, Kentucky, March 2007.

“New approaches to the development of the U.S. computing work force: Assessing the issues,” panel participant at the American Association for the Advancement of Science meeting, San Francisco, California, February 2007.

“Learned societies in an international context: A perspective from the United States,” plenary presentation at the annual Conference of Professors and Heads of Departments of Computing, Glasgow, Scotland, March 2006.

“Nifty Assignments: Breakout!,” panel participant at the Thirty-seventh SIGCSE Technical Symposium on Computer Science Education, Houston, Texas, March 2006.

“The ACM Java Task Force: The Beta Release,” special session at the Thirty-seventh SIGCSE Technical Symposium on Computer Science Education, Houston, Texas, March 2006.

“The ACM Java Task Force: Implications for high schools,” presentation to the Computer Science and Information Technology Symposium, St. Louis, Missouri, February 2005.

“The ACM Java Task Force: Status report,” special session at the Thirty-sixth SIGCSE Technical Symposium on Computer Science Education, St. Louis, Missouri, February 2005.

“Teaching the psychology of debugging,” plenary talk at the winter meeting of the Psychology of Programming Interest Group. University of Nottingham, Nottingham, England, December 2004.

“Labor dynamics of the IT economy: What IT planners need to know about the nature of programming,” keynote address at the U.S. State Department IT Strategy Conference, San Francisco, November 2004.

“The Bermuda Project: Developing a new computing curriculum for Bermuda’s public schools,” plenary talk at a workshop for pre-college computer science teachers. University of Leeds, Leeds, England, July 2004.

“Expanding the audience for computer science,” keynote address at the Midwest Instruction and Computing Symposium, Morris, Minnesota, April 2004.

“Educating the next generation of computer scientists: The critical role of high school teachers,” opening keynote address for the Computer Science and Information Technology Symposium, Norfolk, Virginia, March 2004.

“SIGCSE special projects showcase,” panel presentation at the Thirty-fifth SIGCSE Technical Symposium on Computer Science Education, Norfolk, Virginia, March 2004.

“Resources to support the use of Java in introductory computer science education,” panel chair at the Thirty-fifth SIGCSE Technical Symposium on Computer Science Education, Norfolk, Virginia, March 2004.

“Great principles of computer science: Strategies for integrating fundamental concepts into the classroom,” panel presentation at the Thirty-fifth SIGCSE Technical Symposium on Computer Science Education, Norfolk, Virginia, March 2004.

“Computing Curriculum 2001: Content, development, and application,” Open University of the Netherlands, Utrecht, Holland, June 2003.

“Computing accreditation in the United States,” Open University of the Netherlands, Utrecht, Holland, June 2003.

“Strategies and tactics for the first course in computer science,” University of Kent, Canterbury, England, March 2003.

“Expanding the audience for computer science,” plenary address at the Thirty-fourth SIGCSE Technical Symposium on Computer Science Education, Reno, Nevada, February 2003.

“Professional development and Computing Curricula 2001,” plenary presentation at the annual Conference of Professors and Heads of Departments of Computing, Milton Keynes, England, October 2002.

“What makes software difficult?,” plenary presentation at the 23rd ISODARCO Summer Course on “Cyberwar, Netwar, and the Revolution in Military Affairs: Real Threats and Virtual Myths,” Trento, Italy, August 2002.

“Recruitment and retention: Preliminary results of the CRA study,” panel presentation at the Computing Research Association Conference, Snowbird, Utah, July 2002.

“Undergraduate curriculum and accreditation advances: Computing Curricula 2001,” presentation at the Computing Research Association Conference, Snowbird, Utah, July 2002.

“Computing Curricula 2001 and the challenge of defining an international curriculum,” keynote address at the IFIP WG3.2 Working Conference on Informatics Curricula, Teaching Methods, and Best Practice, Florianopolis, Brazil, July 2002.

“Computing Curricula 2001: Implementing the recommendations,” panel chair at the Thirty-third SIGCSE Technical Symposium on Computer Science Education, Cincinnati, Ohio, February 2002.

“Computing Curricula 2001: An unveiling,” panelist at the 2001 Frontiers in Education Conference, Reno, Nevada, October 2001.

“Panel on Computing Curricula 2001: Computer engineering,” panelist at the 2001 Frontiers in Education Conference, Reno, Nevada, October 2001.

“Technology against terrorism,” panelist at a campus-wide symposium on the events of 9/11, Stanford University, October 2001.

“Computing Curricula 2001: En route to the Steelman draft,” presentation at the ITiCSE 2001 Conference, Canterbury, England, June 2001.

“Computing Curricula 2001 and its relation to ADMI institutions,” presentation at the 2001 Conference of the Association of Computer Information Science and Engineering Departments at Minority Institutions (ADMI), Hampton, Virginia, May 2001.

“Computing Curricula 2001,” panel chair at the Thirty-second SIGCSE Technical Symposium on Computer Science Education, Charlotte, North Carolina, February 2001.

“Curricula 2001 for computer science and engineering,” panelist at the 2000 Frontiers in Education Conference, Kansas City, Missouri, October 2000.

“Computing Curriculum 2001: Getting Down to Specifics,” Consortium for Computing in Small Colleges (Northwest), Beaverton, Oregon, October 2000.

“Computing Curricula 2001: Status report,” presentation at the Computing Research Association Conference, Snowbird, Utah, July 2000.

“Computing Curricula 2001: Preliminary overview,” plenary presentation at the annual Conference of Professors and Heads of Departments of Computing, Brighton, England, April 2000.

“Computing Curricula 2001: Evaluating the Strawman Report,” panel presentation at the Thirty-first SIGCSE Technical Symposium on Computer Science Education, Austin, Texas, March 2000.

“The Internet Revolution: Promises and Pitfalls,” Eugene M. Lang Annual Lecture, Swarthmore College, March 2000. Also presented at Princeton University in July 2000.

“Curricula 2001 for computer science and engineering,” panelist at the 1999 Frontiers in Education Conference, San Juan, Puerto Rico, November 1999.

“Curriculum 2001: Interim report from the ACM/IEEE-CS task force,” panel chair at the Thirtieth SIGCSE Technical Symposium on Computer Science Education, New Orleans, Louisiana, March 1999.

“Integrating professionalism into undergraduate degree courses in computing,” panel participant at the third SIGCSE/SIGCUE Joint Conference on Integrating Technology in Computer Science Education, Dublin, Ireland, August 1998.

“Cyberlibertarianism vs. technorealism: Visions of the new millennium,” Distinguished Speaker Series, Eleanor Roosevelt College, University of California at San Diego, April 1998.

“The Microsoft monopoly: Fact or fabrication,” panel participant, Stanford University, April 1998.

“The retention of women in computer science,” panel participant at the Twenty-ninth SIGCSE Technical Symposium on Computer Science Education, Atlanta, Georgia, February 1998.

“Large introductory courses in research computer science departments,” panel participant at the Twenty-ninth SIGCSE Technical Symposium on Computer Science Education, Atlanta, Georgia, February 1998.

“Expanding the audience for science and engineering,” conference panel sponsored by the Feminist Studies program and the Center for Teaching and Learning, Stanford, June 1997.

“Strategies for increasing student involvement in computer science courses,” Reed College, Portland, Oregon, March 1997. Also presented at the Dartmouth College Computer Science Department, Hanover, New Hampshire, May 1997.

“Strategic directions in computer science education,” panel participant at the Twenty-eighth SIGCSE Technical Symposium on Computer Science Education, San Jose, California, February 1997.

“Managing large introductory courses,” panel participant at the Twenty-eighth SIGCSE Technical Symposium on Computer Science Education, San Jose, California, February 1997.

“Designing a wide-audience introduction to computer science,” opening address at the Symposium on New Ideas in Teaching Computer Science, University of Toronto, December 1995.

“The culture of science,” panel discussion in the President’s Lecture Series on “Culture and Cultures,” Stanford University, May 1995.

“Women in computer science: barriers to academic success,” Jing Lyman Lecture Series, Institute for Research on Women and Gender, February 1995. Also presented at the U.S. Geological Survey in Menlo Park, March 1995.

“The introductory CS curriculum at Stanford: Strategies and tactics,” New England Consortium for Undergraduate Science Education (NECUSE) Conference on Computer Science Education, Harvard University, January 1995.

“Global Information Infrastructure: Connecting the future,” panel discussion at the Student Pugwash National Chapter Conference, Stanford University, January 1994.

“Using C in CS1,” Computer Science Colloquium Series, University of Nevada, April 1993.

“The National Information Infrastructure,” interview aired on PBS’s McNeill-Lehrer News Hour, April 1993.

“Redesign of the introductory computer science course at Stanford,” NECUSE Conference on Computer Science Education, Harvard University, January 1993.

“The ethical responsibility of the scientist,” National Public Radio’s Talk of the Nation program, August 1992.

“Software reliability,” panel presentation, American Society of Mechanical Engineers, Computers in Engineering Symposium, San Jose, California, August 1991.

“Technology and the poor,” keynote panel presentation, Conference on Computers and Social Change, sponsored by the Boston Computer Society, Boston, Massachusetts, April 1991.

“Undergraduate computer science education at Stanford,” NECUSE Conference on Computer Science Education, MIT, January 1991.

“Priorities for science and technology research: Addressing the problems of the 21st century,” University of Southwestern Louisiana, February 1991.

“The ethical responsibility of the computer scientist,” keynote address, Drew University Graduate School Colloquium on “Scholarship and moral choice in contemporary society,” April 1990. Also presented in August 1990 at the University of Minnesota, the University of Wisconsin, and Carnegie Mellon University. Presented as part of a university colloquium series at Northeastern Illinois University, January 1991.

“Concurrent programming in the United States: A research overview,” Institute of Machine Tools, Moscow, USSR, September 1989.

“Programming environments at DEC/SRC: Vulcan and Vesta,” International Federation of Information Processing Working Group 2.4 (Systems Implementation Languages), Warsaw, Poland, September 1989.

Senior participant for session on “Computers and human interaction,” Student Pugwash Conference, Boulder, Colorado, June 1989.

“Computers and the workplace,” Computer Professionals for Social Responsibility (CPSR), Berkeley chapter, March 1989; Stanford University, May 1989; George Washington University, August 1990.

“Professional responsibility and computer science,” Society of Women Engineers Conference, Stanford, January 1988; Democratic Socialists of America (DSA) Summer Youth Conference, Kalamazoo, Michigan, August 1988; Stanford University, March 1989.

“Trip report: Inside the Blue Cube,” CPSR/Palo Alto, October 1988.

“*parmake* and *dp*: Experience with a distributed, parallel implementation of make,” BASS-11 (Bay Area System Seminar), Cupertino, December 1987.

“WorkCrews: An abstraction for controlling parallelism,” Reed College, October 1987; Bolt Beranek and Newman, April 1988; DEC/Littleton, April 1989; University of Wisconsin, August 1990.

“The Star Wars computer system: Is the software feasible?,” Reed College, October 1987; Institute for Advanced Study, October 1987; Wellesley College, October 1987; San Francisco State University, December 1987; University of California at Berkeley, February 1988; California State University at Fresno, February 1988; Bolt Beranek and Newman, April 1988; Stanford University, April 1989.

“Defense procurement and the programming industry,” CPSR/Boston, May 1985; CPSR/Palo Alto, October 1985; Stanford University, November 1987; CPSR/Denver, July 1988.

COURSES TAUGHT

Stanford University, courses in core computer science

Programming Methodology (CS 106A, CS 106AJ, ENGR 70A, OSPOXFRD 22). Stanford’s standard introductory programming course, for both majors and non-majors who want a serious grounding in the basics of software development. Taught over the years in Pascal, C, Java, and JavaScript. (23 offerings, including one at Stanford’s Overseas Studies campus in Oxford, most recently in 2016-17)

Programming Abstractions (CS 106B, ENGR 70B). Stanford’s standard second-quarter computer science class covering data structures and algorithms. (16 offerings, most recently in 2014-15)

Programming Methodology and Abstractions (CS 106X, ENGR 70X). Accelerated introduction covering both CS 106A and CS 106B. (1992-93)

Introduction to Computing (CS 105). General introduction to computer science for non-majors. (1990-91)

Sophomore College: The Intellectual Excitement of Computer Science (CS 10SC). Three-week intensive seminar offered prior to the start of each year to 12 rising sophomores with an intellectual passion for computer science. (16 offerings, including two with President John Hennessy and one with Mehran Sahami, most recently in 2015-16)

Freshman Seminar: Great Ideas in Computer Science (CS 54N). Seminar that covers much the same material as the Sophomore College class, spread out over an entire quarter. (Seven offerings from 2000-01 to 2016-17)

Software Engineering in C (CS 193U). Practicum on software development using C and Unix. (1991-92)

Senior Project (CS 194). Group-project class that fulfills the capstone requirement for majors. (Four offerings, most recently in 1997-98)

Undergraduate Seminar: Overview of Faculty Research (CS 200). Seminar for undergraduates interested in pursuing graduate research in computer science. (Five offerings, most recently in 1999-2000)

Stanford University, courses with a significant computer science component

Computers, Ethics, and Public Policy (CS 181, CS 201, ETHICSOC 100, STS 215, SYMSYS 201C, OSPBER 45). Introduction to the social and ethical implications of computing that meets the university requirements in ethical reasoning and writing in the major. (15 offerings, including two small courses at Stanford's Berlin campus, most recently in 2013-14)

Introduction to Cognitive Science (SYMSYS 100, LINGUIST 144, PHIL 190, PSYCH 132). Required immigration course for undergraduates majoring in Stanford's Symbolic Systems program, which spans the departments of computer science, philosophy, psychology, and linguistics. (2008-09 with Tom Wasow and Jay McClellan)

Networks: Ecological, Revolutionary, Digital (THINK 29) (2012-13 with Deborah Gordon and Dan Edelstein)

MLA Seminar: Great Ideas in Computer Science (MLA 321). Seminar for eight mostly nontechnical students in Stanford's Masters of Liberal Arts program. (2015-16)

British Technologies of World War II (OSPOXFRD 18). Seminar on the history of British wartime advances in technology, including cryptography, radar, and antibiotics. (2003-04 at the Stanford campus in Oxford)

Stanford University, courses in computer science pedagogy

Teaching Computer Science (CS 198). Quarter-long course to prepare our undergraduate section leaders to teach in the introductory sequence. (Oversight responsibility for more than 30 quarters between 1990-91 and 2014-15)

Seminar in Computer Science Education (CS 298) Advanced seminar on computer science curriculum and pedagogy for undergraduates and graduate students. (14 offerings, most recently in 2016-17)

Graduate Seminar on Teaching and Communication (CS 301). Seminar designed to help develop pedagogical skills of teaching assistants. (2006-07)

Intellectual History of Computer Science (CS 208E). Seminar for students in our pilot Master's program in Computer Science Education, designed to help increase the supply of university and college teachers. (2016-17)

Stanford University, interdisciplinary courses

Senior Colloquium (STS 200). Capstone course for Stanford's program in Science, Technology, and Society. My first seminar focused on science and anti-science in America, the second on the promise and peril of the digital age. (Two offerings, most recently in 2009-10)

Freshman Seminar: The Two Cultures—Bridging the Gap (CS 99E, ENGLISH 99E). Builds on C. P. Snow's "Two Cultures" essay by looking at a set of readings from the perspective of both a literary and a scientific intellectual. (Two offerings with Ramón Saldivar)

Technological Visions of Utopia (CS 68N, IHUM 58, THINK 9) (11 offerings, six with Rob Robinson and two with Ursula Heise, most recently in 2014-15)

Le meilleur des mondes possibles: French utopias in theory and practice (OSPPARIS 71). Four-student seminar on French utopian writing. (2012-13 at Stanford's campus in Paris)

Activism in the 1960s: England and the United States (OSPOXFRD 25). A comparative analysis of the political history of the 1960s in the two countries. (2002-03 at the Stanford campus in Oxford)

Reading and Writing Poetry about Science (STS 103Q, ENGLISH 103Q). Seminar/practicum on the analysis and composition of poetry informed by science. (Three offerings with Lauren Rusk, most recently in 2015-16)

Wellesley College

Introduction to Computer Programming and Computation (CS 110). Traditional first-semester programming course using BASIC. (Five offerings, most recently in 1982-83)

Data Structures (CS 230). Traditional second-semester course covering data structures in Pascal, along with algorithmic efficiency and machine representation. (Three offerings, most recently in 1982-83)

Assembly Language Programming (CS 240). Introduction to programming for the DEC PDP-10. (Two offerings, most recently in 1982-83)

Compilers (CS 301). Traditional course on compilers and programming languages. (1982-83)

Swarthmore College

Computers, Ethics, and Social Responsibility (CS 91). Introduction to the social and ethical implications of computing similar to Stanford's CS 181. (1999-2000)

Harvard University

Theory and Construction of Compilers (CS 163). Course on compilers that began with an intensive implementation relying on first principles, followed by a reimplementaion that took advantage of theoretical results. (1983-84)

Automatic Computing (NatSci 110). Summer school version of Harvard’s pioneering introductory course. (Two offerings, most recently in 1975)

INDUSTRIAL RESEARCH EXPERIENCE

Principal Software Engineer, Systems Research Center, Digital Equipment Corporation, 1985-90. Member of the SRC research staff—a team of research scientists seeking, in the words of the SRC charter, “to design, build, and use new digital systems five to ten years before they become commonplace.” At SRC, my principal research activities included the Vulcan programming environment for Modula-2+, design and implementation of the **lauralee** mail system, and a parallel version of the Unix **make** utility for the Firefly multiprocessor.

Consultant, Digital Equipment Corporation, summer 1982. Implemented a significantly enhanced version of BASIC-PLUS-2 for the DECsystem-20 based on experimental language modifications undertaken at Wellesley.

Staff scientist, Bolt Beranek and Newman Inc., Cambridge, Massachusetts, 1976-79. My work at BBN was concentrated in multiprocessor system research, primarily in connection with the BBN Pluribus, and with various Unix-based systems.

AWARDS, FELLOWSHIPS, AND DISTINCTIONS

Fellow of the Association of Computing Machinery (ACM).

Fellow of the American Association for the Advancement of Science (AAAS).

Award for Lifetime Service to the Computer Science Education Community, ACM Special Interest Group in Computer Science Education (SIGCSE), 2018. This award honors an individual who has a long history of volunteer service to the computer science education community.

Karl V. Karlstrom Outstanding Educator Award, ACM, 2012. The Karlstrom Award recognizes educators who have “advanced new teaching methodologies; effected new curriculum development in Computer Science and Engineering; or contributed to ACM’s educational mission.”

Taylor Booth Education Award, Institute for Electrical and Electronic Engineers Computer Society (IEEE-CS), 2012. The Taylor Booth award is given annually to recognize an “outstanding record in computer science and engineering education.”

Laurance and Naomi Carpenter Hoagland Prize, Stanford University, 2004. This award focuses specifically on the teaching of undergraduates and is awarded annually to a “member of Stanford’s faculty who excels in this activity as an opportunity to pursue new directions in teaching that will benefit Stanford undergraduates on a continuing basis.”

Award for Outstanding Contribution to Computer Science Education, ACM SIGCSE, 2003. The description for this award, given annually since 1981, reads as follows: “This award goes to an individual or group in recognition of a significant contribution to computer science education. The contribution may take many forms, such as: curriculum design, innovating teaching methods, textbook authorship, development of new teaching tools, or any of a number of other significant contributions to computer science education. The contribution should have had long lasting impact on, and made a significant difference in, computing education.”

John A. and Cynthia Fry Gunn University Fellow in Undergraduate Education, Stanford University, 2002. These fellowships were established “to reward faculty who make truly outstanding contributions to Stanford’s undergraduate experience.” My fellowship was one of the inaugural eight fellowships awarded under this program.

Lloyd W. Dinkelspiel Award, Stanford University, 1998. This award recognizes “distinctive and exceptional contributions to undergraduate education at Stanford University.” Two awards are made each year at commencement “to the faculty or staff members adjudged to have made the most distinctive contribution to the development and enrichment of undergraduate education in the broadest sense.”

Founders’ Award, Computer Professionals for Social Responsibility (CPSR), June 1996. The text of the citation indicates that the award is made “in appreciation of his enduring commitment to providing the public and policymakers with realistic assessments of the power, promise, and problems of information technology.”

Social Service in Computing Award, ACM Special Interest Group on Computers and Society (SIGCAS), 1996. The text of the award reads: “The ACM Special Interest Group on Computers and Society recognizes the outstanding work that Professor Eric Roberts has done over the past two decades to raise the awareness of social and ethical responsibility among computer professionals. His tireless effort and leadership in such groups as Computer Professionals for Social Responsibility has made a significant impact upon the computer field and has helped shape public policy toward the beneficial and equitable use of computing technology in our society.”

Perin Award for Undergraduate Engineering Education, Stanford University, 1995. Two of these awards are given each year to faculty members in the School of Engineering to support the improvement of undergraduate education.

Bing Fellowship, Stanford University, 1993-95. This fellowship was created “to recognize excellence in teaching and a committed interest to the teaching of undergraduates.”

National Science Foundation Graduate Fellow, 1973-76.

National Merit Scholar, 1969.

PROFESSIONAL ACTIVITIES

Member of the ACM Education Board, 1997-2017 (chair from 2005-07). The ACM Education Board sets policies related to education for the Association for Computing Machinery, which is the premier professional society for computing.

Member of the Board of Advisors for Camp Amelia, 2004-07. Camp Amelia runs computer engagement camps for elementary school children in low-income areas of the United States as well as international camps in Ghana, South Africa, and Vietnam.

Member of the Academic Alliance for the National Center for Women in Information Technology, 2004-present.

Member of IFIP Working Group 3.2 (Informatics and ICT in Higher Education), 2003-present.

Chair of the ACM Java Task Force, 2003-07. The Java Task Force has received funding in excess of \$50,000 from the National Science Foundation and other sources.

Member of the Board of Advisors for the Computer Science Teachers Association, 2003-present.

Member of Computing Research Association study panel on Recruitment and Retention of Faculty in Computer Science and Engineering, 2001-02.

Co-chair of the ACM Task Force for Computing Curriculum 2001, 1998-2001.

Information Director for the ACM Special Interest Group on Computer Science Education (SIGCSE), 1997-2005.

Member-at-large of the Board of Directors for the ACM Special Interest Group on Computer Science Education (SIGCSE), 1997-2001.

Member of the Board of Directors, Student Pugwash USA, 1997-2013 (chair from 1999-2002). Student Pugwash USA is the student affiliate of the Pugwash Conferences on Science and World Affairs, which was awarded the Nobel Peace Prize in 1995. The mission of Student Pugwash is “to encourage young people to examine the ethical, social, and global implications of science and technology, and to make these concerns a guiding focus of their academic and professional endeavors.”

President, Computer Professionals for Social Responsibility, 1990-96. CPSR is a public-interest organization of computer scientists and other professionals concerned about the impact of computing technology on society. Prior to becoming President, served as the CPSR National Secretary from 1987 to 1990.

Member of the Education Working Group for the Strategic Directions in Computing Research workshop, 1996. The Strategic Directions in Computing Research workshop was an invitational symposium held in June 1996 to identify research priorities in many different areas of computer science. The forum was sponsored by the Association for Computing Machinery, the Computing Research Association, the National Science Foundation, and the Office of Naval Research.

Member of the Board of Advisors for the Ethics Center for Engineering and Science, 1995-99. This center, funded by the National Science Foundation and directed by Professor Caroline Whitbeck at MIT, focused its attention on developing materials “to help engineers and scientists clarify the problems they face.”

Member of the ImpactCS Steering Committee, 1994-98. This group consisted of 25 nationally recognized academicians involved in teaching ethics and computer science.

Member of the Aspen Institute summit group on computer science policy, 1992. This group consisted of the chief executives and elected presidents of eight computer science organizations (AAAI, ACM, CPSR, CRA, CSPP, CSTB, IEEE/CS, and SIAM).

Reviewer for National Science Foundation, Instrument and Laboratory Improvement Program, 1992.

Member of the National Board, Research Center on Computing and Society, New Haven, Connecticut, 1991-93.

Member of the National Board, The Democracy Project, San Francisco, California, 1991-94.

Editor of the “Computing and the Citizen” column for *Abacus* magazine, 1986-88.

Editor of *Working Notes*, a monthly newsletter published by CPSR’s Computers in the Workplace Project, 1987-90.

Local Arrangements Chair, ACM SIGSOFT/SIGPLAN Software Engineering Symposium on Practical Software Development Environments, December 1986.

Member of the Association for Computing Machinery (ACM); ACM Special Interest Groups SIGCSE and SIGCAS; the IEEE Computer Society; and the American Association for the Advancement of Science.