Nominations for the ACM SIGCSE Top Ten Symposium Papers of All Time Award

Brief overview of the 20 nominated papers

Year	Title	Author(s)	Abstract	URL
1974	Relating Computer Science programs to the needs of industry through internships and cooperative programs	Alton R. Goddard	The purpose of this paper is to discuss some approaches for achieving more meaningful relationships between degree programs at the college level in Computer Science and the needs…	(https://dl.acm.org/citation.cfm?id=810463)
1974	What should we teach in an introductory programming course?	David Gries	An introductory course (and its successor) in programming should be concerned with three aspects of programming: 1. How to solve problems, 2. How to describe an algorithmic solution to a problem, 3. How to verify that…	(https://dl.acm.org/citation.cfm?id=810447)
1978	The introductory programming course in computer science: ten principles	G. Michael Schneider	The first course in the overwhelming majority of Computer Science Departments is an introductor y course in high-level language programming [1] . Because of this a number of papers…	(https://dl.acm.org/citation.cfm?id=990598)
1983	A project oriented course on software engineering	Sallie Henry	This paper makes an attempt to outline Software Engineering course with particular attention on the administration of the project. Section II describes the type of students taking…	(https://dl.acm.org/citation.cfm?id=952978.801013)

1997	Undergraduate women in computer science: experience, motivation and culture	Allan Fisher, Jane Margolis, Faye Miller	For the past year, we have been studying the experiences of undergraduate women studying computer science at Carnegie Mellon University, with a specific eye toward understanding the…	(https://dl.acm.org/citation.cfm?id=268127)
1998	Conceptual models and cognitive learning styles in teaching recursion	Cheng-Chih Wu, Nell B. Dale, Lowell J. Bethel	An experimental research design was implemented in an attempt to understand how different types of conceptual models and cognitive learning styles influence novice programmers when…	(https://dl.acm.org/citation.cfm?id=274315)
1998	Constructivism in computer science education	Mordechai Ben-Ari	Constructivism is a theory of learning which claims that students construct knowledge rather than merely receive and store knowledge transmitted by the teacher. Constructivism has been…	(https://dl.acm.org/citation.cfm?id=274308)
1998	Design patterns: an essential component of CS curricula	Owen Astrachan, Garrett Mitchener, Geoffrey Berry, Landon Cox	The field of software patterns has seen an explosion in interest in the last three years. Work to date has been on the recognition, cataloging, and finding of patterns with little…	(https://dl.acm.org/citation.cfm?id=273182)
2001	Contributing to success in an introductory computer	Brenda Cantwell Wilson, Sharon Shrock	This study was conducted to determine factors that promote success in an	(https://dl.acm.org/citation.cfm?id=364581)

	science course: a study of twelve factors		introductory college computer science course. The model included twelve possible predictive factors includi…	
2002	Defensive Climate in the Computer Science Classroom	Lecia Jane Barker, Kathy Garvin-Doxas, Michele Jackson	As part of an NSF-funded IT Workforce grant, the authors conducted ethnographic research to provide deep understanding of the learning environment of computer science classrooms…	(https://dl.acm.org/citation.cfm?id=563354)
2003	Gender differences in computer science students	Sylvia Beyer, Kristina Rynes, Julie Perrault, Kelly Hay, Susan Haller	We examined gender differences and differences in Computer Science (CS) majors vs. non-majors in ability in quantitative areas, educational goals and interests, experience with com…	(https://dl.acm.org/citation.cfm?id=611930)
2003	Improving the CS1 experience with pair programming	Nachiappan Nagappan, Laurie Williams, Miriam Ferzli, Eric Wiebe, Kai Yang, Carol Miller, Suzanne Balik	Pair programming is a practice in which two programmers work collaboratively at one computer, on the same design, algorithm, or code. Prior research indicates that pair programmers…	(https://dl.acm.org/citation.cfm?id=612006)
2003	Teaching objects-first in introductory computer science	Stephen Cooper, Wanda Dann, Randy Pausch	An objects-first strategy for teaching introductory computer science courses is receiving increased attention from CS educators. In this	(https://dl.acm.org/citation.cfm?id=611966)

			paper, we discuss the challenge of the obje…	
2004	Using software testing to move students from trial-and-error to reflection-in-action	Stephen H. Edwards	Introductory computer science students rely on a trial and error approach to fixing errors and debugging for too long. Moving to a reflection in action strategy can help students…	(https://dl.acm.org/citation.cfm?id=971300.971312)
2008	Programming by choice: urban youth learning programming with scratch	John H. Maloney, Kylie Peppler, Yasmin Kafai, Mitchel Resnick, Natalie Rusk	This paper describes Scratch, a visual, block-based programming language designed to facilitate media manipulation for novice programmers. We report on the Scratch programming expe…	(https://dl.acm.org/citation.cfm?id=1352260)
2010	Identifying student misconceptions of programming	Lisa C. Kaczmarczyk, Elizabeth R. Petrick, J. Philip East, Geoffrey L. Herman	Computing educators are often baffled by the misconceptions that their CS1 students hold. We need to understand these misconceptions more clearly in order to help students form cor…	(https://dl.acm.org/citation.cfm?id=1734299)
2011	App inventor and real-world motivation	David Wolber	App Inventor is a visual "blocks" language for creating mobile apps. As part of a Google pilot program, App Inventor was taught to university students in a core curriculum course…	(https://dl.acm.org/citation.cfm?id=1953329)
2012	The fairy performance assessment: measuring	Linda Werner, Jill Denner, Shannon	Computational thinking (CT) has been described as an	(https://dl.acm.org/citation.cfm?id=2157200)

	computational thinking in middle school	Campe, Damon Chizuru Kawamoto	essential capacity to prepare students for computer science, as well as to be productive members of society. But efforts to eng…	
2016	A Multi-institutional Study of Peer Instruction in Introductory Computing	Leo Porter, Dennis Bouvier, Quintin Cutts, Scott Grissom, Cynthia Lee, Robert McCartney, Daniel Zingaro, Beth Simon	Peer Instruction (PI) is a student-centric pedagogy in which students move from the role of passive listeners to active participants in the classroom. Over the past five years, the…	(https://dl.acm.org/citation.cfm?id=2844642)
2018	Upward Mobility for Underrepresented Students: A Model for a Cohort-Based Bachelor's Degree in Computer Science	Sathya Narayanan, Kathryn Cunningham, Sonia Arteaga, William J. Welch, Leslie Maxwell, Zechariah Chawinga, Bude Su	CSin3 is a cohort-based, three-year computer science bachelor's degree program that has increased graduation rates of traditionally underrepresented computer science students. A co…	(https://dl.acm.org/citation.cfm?id=3159551)