

Michael D. George

www.cs.cornell.edu/~mdgeorge

mdgeorge@cs.cornell.edu

607.351.4153

EXPERIENCE

Lecturer 2013–2020

Cornell University

- Covered a wide breadth of theoretical and applied subjects in computer science, courses listed below
- Effectively taught both large courses (300–450 students) and small (15–50 students)
- Organized, trained, mentored, and managed 25–40 graduate and undergraduate TAs each semester
- Expanded existing courses to cover new material, with emphasis on practical skills
- Improved student participation and performance by using active learning techniques
- Developed new problem sets, programming assignments, and labs
- Created new lecture notes, an extensive course wiki, and interactive video lectures

Graduate research assistant 2006–2013

Cornell University

- Published 4 peer-reviewed papers in top conferences and journals, listed below
- Research spanning programming languages, type systems, distributed systems, and security
- One of four lead developers of a large programming language and distributed system (~100k loc)
- Research also involved formal modeling and security proofs for smaller programming languages

Previous positions 1999–2006

Saga software, CNS Digital systems, Cornell University, University of Rochester

- Teaching assistant for several graduate and undergraduate math and computer science courses
- Part-time software research and development consultant

EDUCATION

Ph.D. Computer Science 2004–2013

Cornell University

- Dissertation: Trust, Authority, and Information Flow in Secure Distributed Systems.
- Degree conferred December 2020.

MA Mathematics 2003–2004

University of Rochester

- Primary area of study: algebraic topology.

BS Computer Science; Honors BA Mathematics 2000–2003

University of Rochester

- Graduated magna cum laude. Co-valedictorian of the computer science department.

SKILLS

- Professional experience in OCaml, Java, LaTeX, C, C++, Python, and Linux.
- Familiarity with Coq, Rust, Haskell, JavaScript, and SQL.
- Proficiency in software design, compiler development, systems programming, mathematics, and testing.
- Professional experience developing and giving lectures, training, mentoring, publishing academic research papers, writing documentation, and developing instructional material.
- Good-natured and pleasant to work with.

TEACHING

Discrete mathematics

Cornell, CS 2800

Taught 14 semesters, most recently spring 2020

Course covers mathematical tools used in computer science: sets, functions, and relations, countability, probability, logic, combinatorics, finite automata and regular expressions, elementary number theory, induction, graphs. Heavy emphasis on formal reasoning, building mathematical models, and proofs.

Introduced discussion sections and other active learning techniques. Developed extensive course wiki. Increased focus on formal proofs and mathematical rigor. Implemented remote instruction for spring 2020.

Object-oriented programming and data structures

Cornell, CS 2110

Taught in summer 2006, 2019, co-taught in fall 2016

Course covers Java, object-oriented programming, design patterns, introductory data structures and algorithms, GUI programming.

Improved coverage of software engineering tools including debugging, source control, and design patterns. Introduced discussion sections to the summer offering.

Functional programming and data structures

Cornell, CS 3110

Co-taught 3 semesters, most recently spring 2015

Course covers functional and concurrent programming, writing and using specifications, modular programming and data abstraction, reasoning about program correctness, reasoning about system performance, useful and efficient data structures. Course is project-oriented; students complete 6 small to medium sized projects.

Participated in redesign of course materials in fall 2015, including development of labs and new assignments. Expanded coverage of asynchronous programming and added new lectures on monads.

Operating systems

Cornell, CS 4410

Taught or co-taught 7 semesters, most recently summer 2018

Course covers hardware support for operating systems, threaded programming and synchronization, memory management, filesystems, networking.

Algorithms

Cornell, CS 4820

Taught in summer 2013

Course covers greedy algorithms, dynamic programming, divide and conquer, np completeness, undecidability, proofs of correctness, asymptotic complexity.

PUBLICATIONS

Fabric: Building Open Distributed Systems Securely by Construction. J. Computer Security, 2017. Jed Liu, Owen Arden, Michael D. George, and Andrew C. Myers.

Warranties for Faster Strong Consistency. 11th USENIX Symp. on Networked Systems Design and Implementation (NSDI), April 2014. Jed Liu, Tom Magrino, Owen Arden, Michael D. George, and Andrew C. Myers.

Sharing Mobile Code Securely with Information Flow Control. IEEE Symp. on Security and Privacy, May 2012. Owen Arden, Michael D. George, Jed Liu, K. Vikram, Aslan Askarov, and Andrew C. Myers.

Fabric: A Platform for Secure Distributed Computation and Storage. 22nd ACM Symp. on Operating System Principles (SOSP), October 2009. Jed Liu, Michael D. George, K. Vikram, Xin Qi, Lucas Wayne, and Andrew C. Myers.