Geoffrey Challen

Updated 6/2/2026

né Werner-Allen

My professional goal is to teach computer science to as many people as possible, and to inspire them to use their skills to change the world for the better.

Department of Computer Science University of Illinois Thomas M. Siebel Center for Computer Science 201 North Goodwin Avenue, Urbana, IL, 61801-2302

Email:	challen@illinois.edu
Phone:	716.464.2749
Website:	https://geoffreychallen.com
Public Course Materials:	https://learncs.online
Primary Course Website:	https://cs124.org
CV:	https://geoffreychallen.com/CV
Statements:	https://www.geoffreychallen.com/statements
Previous Research Group Website:	https://bluegroup.systems
Google Scholar Profile:	https://geoffreychallen.com/scholar

My online CV is always more up-to-date than any paper copy.

At this particular moment in computing history, AI has changed what it means to build with computers—and therefore what it means to teach computing. My current work responds along several connected threads.

I've redesigned CS 124's larger programming project around AI coding agents: every student designs and builds their own Android app, and a new independent project teaches the skills that matter in the AI era—writing specifications, working with agents, and evaluating what they produce—while classical programming remains assessed through proctored quizzes.

Alongside that integration, I'm building new courses for students who arrive without a CS background. I piloted CS 199 UAI: Using and Understanding AI in Spring 2026 for non-technical students, and am preparing its permanent form, CS 108: Exploring Generative AI, which I'll first teach in Spring 2027. I'm leading the design of Applied Computing, a proposed four-year undergraduate degree organized around studio-based design, domain integration, and AI-assisted building. I'm designing CS 199 ASK: Conversational Programming, a new course for anyone who wants to build—no prerequisites, no prior experience, no code—where students learn to work like architects: exploring problems, designing solutions, and partnering with AI to bring ideas to life.

Much of this depends on tools and infrastructure I continue to build: conversational assessment using a two-agent AI architecture, interactive demos that let students explore how AI systems work, and course platforms that are themselves built through conversation with AI. Applying my own skills as a computer scientist to create interactive and engaging learning environments remains central to how I work.

This work has begun to draw recognition beyond my own courses: in 2026 I was selected as a Replit Faculty Fellow—an invited fellowship from Replit, a widely used AI software-development platform—and as a Faculty AI Fellow in the University of Illinois Summer AI Studio, where I provide disciplinary guidance and peer mentorship to colleagues integrating AI into their teaching.

CS 124 remains the foundation for this work, and currently enrolls over 2000 students per year. Since I began teaching the course in Fall 2017, enrollment has doubled, drop rates have fallen, success rates have increased, student performance in later courses has improved, a gender performance gap has disappeared, students are practicing and learning more, the amount of content covered has increased, we have introduced Kotlin as a language option alongside Java, and hundreds of staff members have made contributions that help diversify the course's voice.

Appointments and Preparation

I'm a Teaching Professor at the University of Illinois. My appointment is 60% teaching, 20% scholarship, and 20% service. My focus is teaching at scale.

Previous Position

From 2011–2017 I worked in mobile systems as a research professor at the University at Buffalo. My research group designed, built, and evaluated novel computer systems. We focused on smartphones, since they represent the most pervasive and powerful distributed system ever deployed. My previous group's website is the best source of information about my prior research, and includes project descriptions, copies of all of our published papers and funded grant proposals, information about the courses that we created and taught, and details about former group members.

Academic Appointments

- 2024–: Teaching Professor, University of Illinois
- 2017–2024: Associate Teaching Professor, University of Illinois
- 2011–2017: Assistant Professor, University at Buffalo

Professional Preparation

- 2010–2011, Postdoctoral Associate, Massachusetts Institute of Technology (MIT). Supervised by Hari Balakrishnan.
- 2010, Ph.D, Computer Science, Harvard University. Dissertation: "Data Fidelity and Resource Management for Data-Rich Sensor Networks". Advised by Matt Welsh.
- 2003, AB, Physics, Harvard University.

Honors and Awards

- **2026:** Replit Faculty Fellow, an invited fellowship from Replit, a widely used AI software-development platform, recognizing leadership in teaching computing in the AI era.
- **2026:** Faculty AI Fellow, Summer AI Studio, Center for Innovation in Teaching & Learning, University of Illinois. A campus-wide summer intensive in which experienced faculty provide disciplinary guidance and peer mentorship to colleagues integrating AI into their courses.
- **2026:** Rose Award for Teaching Excellence, The Grainger College of Engineering, University of Illinois. Awarded annually to a member of the College of Engineering's undergraduate teaching faculty in recognition of innovative teaching methods and instructional programs.
- **2026:** Scott H. Fisher Computer Science Teaching Award, Siebel School of Computing and Data Science, University of Illinois. The Siebel School's top honor dedicated to excellence in undergraduate teaching.
- **2017:** National Science Foundation CAREER Award for research on implementation flexibility and runtime adaptation in mobile systems.
- **2015:** SEAS Early Career Teacher of the Year, School of Engineering and Applied Sciences, University at Buffalo.
- **2009:** Siebel Scholar.

Teaching

Over 21 terms at Illinois I have taught a total of 16,107 students[1]. I have focused my efforts on expanding and improving our CS1 course for majors and students with deep interest in the material—previously numbered CS 125 and now CS 124. More recently I've been creating new courses on AI: CS 199 UAI: Using and Understanding AI (continuing as CS 108) and CS 199 ASK: Conversational Programming.

Additional information about each course is linked off its main website. Data about student success in each course and detailed student evaluation results available by request. For more details, please review my teaching statement.

Overview

Only primary teaching assignments are listed.

- **Spring 2027–:** "CS 108: Exploring Generative AI" (New Course) (University of Illinois) — ongoing form of the CS 199 UAI pilot
- **Fall 2026–:** "CS 199 ASK: Conversational Programming" (New Course) (University of Illinois)
- **Spring 2026:** "CS 199 UAI: Using and Understanding AI" (Pilot; superseded by CS 108) (University of Illinois)
- **Fall 2024–:** "CS 199 SOC: Technology and Society" (New Course) (University of Illinois)
- **Fall 2021–:** "CS 124: Introduction to Computer Science I" (Renumbered Course) (University of Illinois)
- **Fall 2017–Summer 2021:** "CS 125: Introduction to Computer Science" (University of Illinois)
- **Fall 2016:** "CS 199: How the Internet Works" (New Course) (University at Buffalo)
- **Spring 2011–Spring 2017:** "CS 421: Introduction to Operating Systems" (University at Buffalo)

Course and Curricular Innovations and Improvements

Links in the items below are usually to pages on `learncs.online`, which contains a publicly-accessible version of the materials and innovations I've developed to support my university CS1 course at Illinois, CS 124.

- **2026–:** Leading the design of a new Applied Computing degree, a proposed four-year undergraduate program at Illinois organized around studio-based design, domain integration, and AI-assisted building
- **2026–:** Designing "CS 199 ASK: Conversational Programming," a Fall 2026 course for anyone who wants to build—no prerequisites, no prior experience, no code—where students learn to work like architects and partner with AI to bring ideas to life
- **2026:** Developed CS 108: Exploring Generative AI as the permanent form of the CS 199 UAI pilot, launching Fall 2026
- **2026:** Created and deployed CS 199 UAI: Using and Understanding AI as a pilot course teaching non-technical students to use and understand AI through hands-on creative projects
- **2026:** Developed novel conversational assessment system using a two-agent AI architecture for scalable oral-exam-style evaluation
- **2026:** Replaced traditional CS 124 programming assignments with a new independent project, where each student designs and builds their own Android application using AI coding agents
- **2025:** Started work on several integrations of AI into course infrastructure, systems, and materials
- **2025:** Began teaching effective use of coding agents on the course larger Android programming project

- **2025:** Enhanced online proctoring interface to support remote students
- **2025:** Added quiz retake opportunities, allowing students additional chances to demonstrate mastery
- **2025:** Completed major database reorganization for improved system performance and scalability
- **2024:** Deployed autogenerated testing exercises, providing students with practice writing test suites
- **2024:** Integrated code quality instruction and automated feedback into daily lesson content
- **2024:** Launched coaching program to proactively contact students in need of support and offer individual course strategy sessions
- **2024:** Deployed Redis-based caching system for better platform performance at scale
- **2024:** Expanded compilation infrastructure to support mixed Java/Kotlin projects
- **2023:** Introduced mutation testing framework for automated debugging challenge generation
- **2023:** Began work adding support for automated test evaluation to our homework autograder
- **2023:** Added profiles of diverse computer scientists to our daily lessons as part of a CS People series
- **2022:** Integrated debugging exercises throughout our CS1 daily lessons
- **2022:** Upgraded our online quiz tool to add improved quiz security features and provide more practice content
- **2022:** Made most of our highly-successful CS1 materials publicly available at https://www.learncs.online/
- **2022:** Established a successful course staff mentorship program
- **2022:** Created a practice page providing students with access to our entire library of over 700 small programming problems
- **2021:** Officially introduced Kotlin as a new language option for CS 124, with parallel lessons, content, and assessments running alongside our existing Java materials
- **2021:** Created a new tool for generating small debugging challenges, allowing successful student submissions to existing homework problems to generate large numbers of new, interesting, and autogradeable questions
- **2021:** Created a series of introductory Kotlin lessons to complement and eventually run side-by-side with our existing Java lessons
- **2020:** Developed a new interactive walkthrough format for deploying live coding explanations and soliciting contributions from course staff
- **2020:** Created a new online quiz system to support remote Zoom proctoring
- **2020:** Restructured CS 125 to support asynchronous instruction in a daily interactive lesson format

- **2020:** Designed and developed a new framework accelerating the development of small Java programming problems
- **2020:** Created an efficient online help system to support students after CS 125 transitioned online
- **2019:** Authored and deployed the first multi-part Android programming project for CS 125
- **2019:** Deployed CS 125 on local cloud infrastructure providing enhanced scalability and robustness
- **2019:** Developed a new Java and JVM playground backend and toolkit to support interactive programming exercises in CS 125
- **2019:** Reauthored our on-demand Git autograder for larger CS 125 assignments to run on Kubernetes
- **2018:** Began authoring a large and growing library of small Java homework problems to support CS 125
- **2018:** Developed a Java autograder and programming problems for CS 125 quizzes
- **2018:** Created on-demand Git autograder for CS 125 assignments
- **2017:** Created on-demand Subversion autograder for CS 125 assignments
- **2016:** Implemented novel video delivery and tracking systems for internet-class.org, the website supporting a new course on the internet
- **2015–2016:** Developed `test161`, a new testing framework for operating system assignments using the OS/161 instructional operating system

Semester-by-Semester Details

Below I provide a summary of the enrollment for each semester I've taught CS1 at Illinois, along with a detailed description of various changes and improvements made to the course each term. Note that we usually try to deploy substantial changes to course materials and infrastructure in the fall.

CS 199 UAI: Spring 2026 (In Progress)

Website

Enrollment: 11
Format: In-person with AI-facilitated activities and conversational assessments

1. Inaugural semester of Using and Understanding AI, a course for non-technical students
2. Students use AI to create websites, music, and data analyses while exploring how AI works and what it means for society
3. Deployed conversational assessments using a two-agent architecture as a novel scalable alternative to traditional exams
4. Created interactive AI demos including Digit Recognition Network, Neuron Explorer, and Markov Babbler
5. Entire course platform built through Claude Code (167 sessions over 47 days)

CS 124: Spring 2026 (In Progress)

Syllabus // Staff

Enrollment: 400
Format: Tutorial format with both online and in-person support, frequent small assessment
Languages: Kotlin, Java

1. Launched a new independent project, replacing the traditional shared machine project with individual student-driven Android applications built using AI coding agents
2. Curriculum now teaches prompt writing, AI code validation, and test-driven development with AI
3. Classical programming assessment maintained through weekly proctored quizzes

CS 124: Fall 2025 (In Progress)

Syllabus // Staff

Enrollment: 1100
Format: Tutorial format with both online and in-person support, frequent small assessment
Languages: Kotlin, Java

1. Integrated effective use of AI coding agents into the larger Android programming project to ensure students are prepared for the future of software development
2. Began work on several AI integrations into course materials and systems, including retrieval-augment generation, code variable name analysis, and automatically generated quiz retake study guides

CS 124: Spring 2025

Syllabus // Staff

Enrollment: 700
Format: Tutorial format with both online and in-person support, frequent small assessment
Languages: Kotlin, Java

1. Enhanced online proctoring interface to support remote students.
2. Added quiz retake opportunities, allowing students additional chances to demonstrate mastery.
3. Completed major database reorganization for improved system performance and scalability.

CS 124: Fall 2024

Syllabus // Staff

Enrollment: 1294
Format: Tutorial format with both online and in-person support, frequent small assessment
Languages: Kotlin, Java

1. Deployed autogenerated testing exercises, providing students with practice writing test suites.
2. Integrated code quality instruction and automated feedback into daily lesson content.
3. Began coaching program to proactively contact students in need of support and offer individual course strategy sessions.
4. Deployed Redis-based caching system for improved platform performance at scale.

CS 124: Spring 2024
Syllabus // Staff

Enrollment: 782
Format: Tutorial format with both online and in-person support, frequent small assessment
Languages: Kotlin, Java

1. Enhanced compilation infrastructure to support mixed Java/Kotlin projects.
2. Modernized platform architecture for improved performance and scalability.

CS 124: Fall 2023
Syllabus // Staff

Enrollment: 1281
Format: Tutorial format with both online and in-person support, frequent small assessment
Languages: Kotlin, Java

1. Deployed enhanced mutation testing framework for automated debugging challenge generation.
2. Continued stability improvements to website frontend and backend systems.
3. Enhanced assessment platform with improved question delivery and evaluation.

CS 124: Spring 2023
Syllabus // Staff // Lessons

Enrollment: 879
Format: Tutorial format with both online and in-person support, frequent small assessment
Languages: Kotlin, Java

1. Added profiles of diverse computer scientists to our daily lessons and weekly quizzes as part of a CS People series.

2. Began work on adding support for autograding test suites for all of the 700 questions in our problem library.
3. Transitioned the course website to a new static site framework to improve login integration and avoid library deprecation.
4. Deployed several new code quality analysis features.
5. Completed work on Kotlin syntactic feature analysis, achieving parity with Java in preparation for a new feature-based code quality analysis pipeline.
6. Began work with Leon Li and Katie Cunningham to analyze data on student usage of online tutoring resources.

CS 124: Fall 2022

Syllabus // Staff // Lessons

Enrollment: 1399
Format: Tutorial format with both online and in-person support, frequent small assessment
Languages: Kotlin, Java

1. Fully incorporated debugging challenges throughout the daily lessons, alongside an updated set of homework problems.
2. Completed work on several new quiz security features, in particular a focus-tracking system that automatically closes a student's assessment when it loses focus more than a configurable number of times.
3. Began work with Rachel Zhou and Luc Paquette on processing data from our daily lessons to analyze how students develop preferences for different instructors.

CS 124: Spring 2022

Syllabus // Staff

Enrollment: 666
Format: Tutorial format, frequent small assessment
Languages: Kotlin, Java

1. Began work on debugging challenges generated from our growing library of student solutions to our homework problems.
2. Released our CS1 materials publicly at `learncs.online`.
3. Improved the staff mentoring program to connect junior and senior course staff members, including involving mentors in staff scheduling to help reduce absenteeism.
4. Began work on improved quiz security features to better secure our online assessments.

CS 124: Fall 2021

Syllabus // Staff

Enrollment: 1441
Co-Instructor: Colleen Lewis
Format: Tutorial format, frequent small assessment
Languages: Kotlin, Java

1. Introduced Kotlin as a language option alongside Java, requiring the development of a parallel set of lessons, homework, and assessments, and project.
2. Colleen Lewis joined as a co-instructor and provided a complete set of alternate explanations for Java-language content.
3. Worked with Colleen Lewis to update several course policies, including enabling students to collaborate on homework problems and on the machine project.
4. Prepared data collection from the daily lessons to examine how students develop preferences when explanations from multiple instructors are available.

CS 125: Spring 2021

(Course website no longer online.)

Enrollment: 613
Format: Asynchronous online, frequent small assessment
Language: Java

1. Continued improving and stabilizing the systems supporting the new course format launched in Fall 2020.

CS 125: Fall 2020

(Course website no longer online.)

Enrollment: 1103
Format: Asynchronous online lessons, frequent small assessment
Language: Java

1. Relaunched the course in a new asynchronous online daily lesson format to support student success during the pandemic, comprising daily interactive lessons integrating text, code playgrounds, interactive walkthroughs, homework problems, and video explanations.
2. Deployed a novel homework autograder enabling rapid authoring and accurate evaluation of programming exercises.
3. Transitioned to a new quizzing and proctoring tool integrating the homework autograder and multiple-choice questions designed to support efficient and secure online assessment.
4. Integrated the online tutoring tool directly into the new course website, further lowering the bar for students to ask questions and receive one-on-one assistance.
5. Completed and deployed a new service enabling continuous saving of student work in online code editors, to support student coding on the homework problems and during

quizzes.

CS 125: Spring 2020

Syllabus // Staff

Enrollment: 380
Format: Synchronous online lectures, frequent small assessment
Language: Java

1. Transitioned the course online mid-semester due to onset of the COVID-19 pandemic.
2. Over spring break, completed the first version of our online tutoring site to allow staff to efficiently provide remote support.
3. Began preparing for fully-online instruction the next semester.

CS 125: Fall 2019

Syllabus // Staff

Enrollment: 696
Format: Synchronous lectures, frequent small assessment
Language: Java

1. Deployed the first iteration of the Android machine *project*, a multi-part assignment where students build a complete Android app through multiple weekly checkpoints, in a process that more accurately reflects real-world software development.
2. Created and deployed our first distributed cloud-native continuous autograder, which we have continued to use to perform on-demand containerized grading of student Android programming assignments.
3. Migrated all course backend systems to new server infrastructure running in our own private cloud managed using Kubernetes.

CS 125: Spring 2019

Website // Syllabus // Staff

Enrollment: 459
Format: Synchronous lectures, frequent small assessment
Language: Java

1. Began work on an initial version of Jeed as an improved backend for the in-class programming examples embedded in the lecture slides.
2. Improved the test suites for the Android programming assignments.

CS 125: Fall 2018

Website // Syllabus // Staff

Enrollment: 742
Format: Synchronous lectures, frequent small assessment
Language: Java

1. Created and deployed an initial set of ~60 daily programming exercises to accompany and support the CS 125 course content.
2. Wrote an additional ~45 small programming exercises to accompany each weekly assessment, marking the first time that every CS 125 quiz included programming questions.
3. Eliminated the high-stakes final exam, completing a transition to frequent small assessment.
4. Established a course tutoring center in a dedicated 40-seat room in the basement of the Siebel Center, replacing a woefully undersized office used for the purpose in previous semesters.
5. Created several new assignments, including an adaptation of a previous photo manipulation assignment to Android, and a molecule-naming assignment using graph traversal, developed with course staff member Ben Nordick.

CS 125: Spring 2018
Website // Syllabus // Staff

Enrollment: 439
Format: Synchronous lectures, high-stakes assessments
Language: Java

1. Rewrote all lectures to use interactive online materials and integrate participation tracking and in-class programming examples into classroom presentations.
2. Deployed the first version of what became Jeed to support web-based exploration of Java code snippets through safe and high-speed untrusted code execution.
3. Completed and deployed continuous assignment autograding.
4. Deployed the first CS 125 online gradebook.
5. Developed Java autograding support for PrairieLearn to enable programming questions on weekly quizzes, which had previously been entirely multiple-choice questions.
6. Introduced Android programming to the course with new labs and assignments.
7. Held the first ever CS 125 Project Fair, featuring 165 Android projects developed by teams of students.
8. Moved the final exam to the computer testing center (CBTF), enabling autograded programming questions and reducing staff grading burden.
9. Updated course development tools, replacing Eclipse with IntelliJ and Subversion with Git and GitHub Classroom.
10. Began a course developer program with 10 students working on projects including new assignments, a new project fair, plagiarism detection, and data analysis.

11. Created several new assignments on imperative programming, functions, and Android programming using APIs.

CS 125: Fall 2017
Website // Syllabus // Staff

Enrollment: 698
Co-Instructors: Bill Chapman, Lawrence Angrave
Format: Synchronous lectures, high-stakes assessment
Language: Java

1. Updated course assignments and infrastructure while Bill Chapman and Lawrence Angrave gave the lectures.
2. Rewrote and updated all assignments to use a modern build system (Gradle), exhaustive test cases, linter (`checkstyle`), and modern Java idioms.
3. Rewrote all lab activities to better match course content and use online (rather than printed) materials.
4. Designed and deployed a new course website and forum.
5. Began work on a continuous Git autograder and online grade return system integrated into the course website.
6. Created several new assignments covering photo manipulation, object-oriented programming, and recursion.
7. Began work on a fast and safe Java small code execution engine, originally based on the Janino high-speed Java compiler.

University at Buffalo (2011–2016)

During my five years at UB I taught a total of 1150 students—an average of 104 per semester—and offered four different courses to beginning undergraduates, advanced undergraduates and graduate students. Despite being extremely challenging and required for undergraduate computer science majors, my course on computer operating systems was among the most popular in the department and a favorite of both graduate and undergraduate students. My graduate seminar on rotating topics in mobile systems attracted a small group of motivated students and recruited several new Ph.D. students into my group, while giving advanced undergraduates the chance to explore research topics. In Fall 2016 I introduced a new course on the internet to freshman undergraduates.

Courses taught at UB, in reverse chronological order:

- **Spring 2017:** CSE 421/521: Introduction to Operating Systems — Enrollment 118 (83 / 35).

- **Fall 2016:** CSE 199: How the Internet Works (New Undergraduate Course) — Enrollment 440; head instructor supervising 4 activity instructors, 8 recitation instructors, and 25 undergraduate teaching assistants.
- **Fall 2016:** CSE 723: Improving Smartphone Quality of Experience (New Graduate Seminar) — Enrollment 8.
- **Spring 2016:** CSE 421/521: Introduction to Operating Systems — Enrollment 140 (86 / 54); instructor rating 4.37, course rating 3.97.
- **Fall 2015:** CSE 720: Using Uncertainty to Program Mobile Systems (New Graduate Seminar) — Enrollment 7.
- **Spring 2015:** CSE 421/521: Introduction to Operating Systems — Enrollment 144 (88 / 56); instructor rating 4.65, course rating 4.24.
- **Fall 2014:** CSE 720: Personal Cloud Computing (New Graduate Seminar) — Enrollment 5.
- **Spring 2014:** CSE 421/521: Introduction to Operating Systems — Enrollment 161 (86 / 75); instructor ratings 4.47 / 4.64, course ratings 3.82 / 4.74.
- **Fall 2013:** CSE 720: Smartphone Sustainability (New Graduate Seminar) — Enrollment 8.
- **Spring 2013:** CSE 421/521: Introduction to Operating Systems — Enrollment 90 (39 / 51); instructor ratings 4.14 / 4.59, course ratings 4.59 / 4.68.
- **Fall 2012:** CSE 622: Advanced Systems Research (Co-taught with Steven Y. Ko) — Enrollment 19; instructor rating 3.77, course rating 3.15.
- **Spring 2012:** CSE 421/521: Introduction to Operating Systems — Enrollment 99 (48 / 51); instructor ratings 3.3 / 3.8, course ratings 4.0 / 4.0.
- **Fall 2011:** CSE 622: Advanced Systems Research (Co-taught with Steven Y. Ko) — Enrollment 23.

My UB online teaching portfolio has information about these and other teaching-related activities at UB. I also have copies of a teaching statement and statements of course effectiveness and improvement that I prepared in 2016.

Curriculum Development

- **2026–:** Leading the design of a new Applied Computing degree, a proposed four-year undergraduate program at Illinois that prepares students to use AI to build for the domains they care about. Organized around studio-based design, a required domain concentration, breadth in studio art, moral reasoning, literature, and advanced writing, and a capstone thesis with public defense. Proposed as a school-level collaboration between the Siebel School, the School of Art and Design, Gies, and the iSchool. Substantive feedback from Lawrence Angrave, Tal August, Max Fowler, Daniel Gonzalez, Cory Gwin, John Hart, Derek Hoiem, and Eric Shaffer.
- **2026–:** Designing “CS 199 ASK: Conversational Programming,” a new Fall 2026 course at Illinois for anyone who wants to build—no prerequisites, no prior experience, no code.

Students learn to work like architects: exploring problems, designing solutions, prototyping with AI, critiquing what emerges, iterating through conversation, and validating with real users.

- **2026:** Piloted "CS 199 UAI: Using and Understanding AI", a course for non-technical students that teaches them to use AI tools for creative projects while building understanding of how AI systems work (University of Illinois). Superseded by the permanent CS 108: Exploring Generative AI, launching Fall 2026.
- **2024:** Designed and deployed "CS 199 SOC: Technology and Society", an exploratory seminar examining students' relationship with technology through reading, reflection, and conversation (University of Illinois).
- **2020–:** Led a team of faculty that proposed and passed a significant revision to the core programming sequence at the University of Illinois. The revision created several new courses and improved the accessibility of the computer science degree program for non-majors.
- **2015–2017:** Chaired the subcommittee that led a comprehensive revision of the undergraduate computer science curriculum at the University at Buffalo (approved Fall 2016). The revision introduced a new three-course programming sequence and a condensed three-course theory sequence, removed outdated requirements, and roughly doubled the number of electives while still meeting ABET accreditation and SUNY seamless-transfer requirements. As part of this effort I co-designed the new introductory programming sequence with Jesse Hartloff.
- **2016:** Designed and deployed "CS 199: How the Internet Works", an undergraduate course at the University at Buffalo.

Projects

Please see my projects page or my statement on scholarly activities for more details.

- CS 199 UAI: Using and Understanding AI is a new course for non-technical students featuring novel conversational assessments, interactive AI demos, and AI-facilitated group discussions. The entire course platform was built through Claude Code.
- CS 124 AI integration rethinks introductory programming for the AI era. Traditional programming assignments are replaced by a new independent project, where each student designs and builds their own Android application using AI coding agents, while classical programming assessment continues through proctored quizzes.
- `learncs.online` is the freely-available public version of the materials developed to support CS 124—my CS1 course at Illinois. It provides 61 daily lessons in Java and Kotlin, 902 runnable playground examples, 2,540 interactive walkthroughs contributed by 264 instructors, and over 700 autograded programming problems. In its first year it supported thousands of independent learners who generated hundreds of thousands of homework

submissions, and it has begun to attract external users and contributors—including Ellen Spertus at Northeastern University, who uses it to support her Kotlin course, and a Summer 2023 high-school outreach program run through the Discovery Partners Institute in Chicago.

- Interactive walkthroughs are a novel way to deliver live coding demos online while preserving code interactivity. They've also allowed us to create a vibrant and diverse community of instructional voices.
- Questioner is a novel tool allowing us to write programming exercises 10x faster and more accurately than previous approaches. It leverages the fact that, when authoring homework problems, the solution is known, and can be used to both generate and validate a testing strategy.
- Code quality autograding allows us to provide students with automated feedback on not just correctness, but also on multiple aspects of quality: linting (format checking), cyclomatic complexity, runtime and memory efficiency, source line counts, dead code detection, and recursion and feature analysis.
- Autogenerated debugging exercises provide students with nearly-unlimited debugging practice, improving their ability to identify small mistakes, work with unfamiliar code, and apply minimal fixes.
- Autogenerated testing exercises provide students with the chance to practice writing test suites. Writing test suites is not only a critical component of software creation, but provides students the opportunity to demonstrate their understanding of a problem, as well as learn to think defensively by anticipating corner cases and problematic inputs.
- Our online tutoring site efficiently connects students with staff for one-on-one tutoring sessions, and has become a primary way that we support my CS1 course's asynchronous format.
- Jeed, the speedy JVM execution and analysis toolkit, is a central component supporting all of our other Java and Kotlin tools. It allows running untrusted code in a secure sandbox 1000 times faster than other approaches, and also supports a large and growing collection of source and bytecode analysis tools.

Earlier instructional and research infrastructure from my time at the University at Buffalo:

- `ops-class.org` is an online operating systems instructional framework built around the OS/161 instructional operating system. It provides the `test161` automated grading tool for the OS/161 programming assignments, along with hundreds of hours of videotaped lectures, slides, and exams from my CSE 421/521 courses.
- PhoneLab was an NSF-funded public smartphone testbed used by researchers at the University at Buffalo and other institutions. Approximately 150 UB students, faculty, and

staff carried instrumented smartphones and received subsidized service in exchange for participating in smartphone experiments.
- `internet-class.org` is a collaborative open-source course on how the internet works, supported by nearly 300 online videos and several dozen in-class activities. It represented an early attempt at developing a modular open-source course.

Publications

My work spans computer science education and, earlier in my career, mobile systems and wireless sensor networks. Full-text copies of my earlier research publications are available on my previous group's website, and citation counts are tracked on Google Scholar.

Computer Science Education

- **02/2025:** Yiqiu Zhou, Luc Paquette, and Geoffrey Challen. **Investigating the Presence and Development of Student Instructor Preferences in a Large-Scale CS1 Course.** In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education (SIGCSE 2025)*.
- **02/2025:** Geoffrey Challen and Ben Nordick. **Accelerating Accurate Assignment Authoring Using Solution-Generated Autograders.** In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education (SIGCSE 2025)*.
- **03/2024:** Hongxuan Chen, Ang Li, Geoffrey Challen, and Kathryn Cunningham. **Implementation of Split Deadlines in a Large CS1 Course.** In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education (SIGCSE 2024)*. DOI: 10.1145/3626252.3630873
- **11/2020:** Liia Butler, Geoffrey Challen, and Tao Xie. **Data-Driven Investigation into Variants of Code Writing Questions.** In *Proceedings of the 32nd IEEE Conference on Software Engineering Education and Training (CSEE&T 2020)*. DOI: 10.1109/CSEET49119.2020.9206195
- **11/2019:** Jonathan Osei-Owusu, Angello Astorga, Liia Butler, Tao Xie, and Geoffrey Challen. **Grading-based test suite augmentation.** In *Proceedings of the 34th IEEE/ACM International Conference on Automated Software Engineering (ASE 2019)*. DOI: 10.1109/ASE.2019.00030

Mobile Systems (University at Buffalo, 2011–2017)

- **2017:** Jinghao Shi, Shuvendu Lahiri, Ranveer Chandra, and Geoffrey Challen. **Wireless Protocol Validation Under Uncertainty.** In *Formal Methods in System Design (FMSD)*, 2017. Extended journal version of the RV'16 **Best Paper**.

- **2017:** Guru Prasad Srinivasa, Rizwana Begum, Scott Haseley, Mark Hempstead, and Geoffrey Challen. **Separated By Birth: Hidden Differences Between Seemingly-Identical Smartphone CPUs.** In *Proceedings of the 18th Workshop on Mobile Computing Systems and Applications (HotMobile'17).*
- **2017:** Scott Haseley and Geoffrey Challen. **Measuring and Improving Smartphone QoE Using the Screen as a Sensor** (Poster Abstract). In *Proceedings of the 18th Workshop on Mobile Computing Systems and Applications (HotMobile'17).*
- **2016:** Rizwana Begum, Guru Prasad Srinivasa, Geoffrey Challen, and Mark Hempstead. **Algorithms for CPU and DRAM DVFS Under Inefficiency Constraint.** In *Proceedings of the 34th IEEE International Conference on Computer Design (ICCD'16).*
- **2016:** Jinghao Shi, Shuvendu Lahiri, Ranveer Chandra, and Geoffrey Challen. **Wireless Protocol Validation Under Uncertainty.** In *Proceedings of the 16th International Conference on Runtime Verification (RV'16).* **Best Paper Award.**
- **2016:** Scott Haseley and Geoffrey Challen. **QoE-Centric Mobile Operating System Design** (Poster Abstract). In *Proceedings of the 14th International Conference on Mobile Systems, Applications and Services (MobiSys'16).*
- **2016:** Jinghao Shi, Lei Meng, Aaron Striegel, Chunming Qiao, Dimitrios Koutsonikolas, and Geoffrey Challen. **A Walk on the Client Side: Monitoring Enterprise Networks Using Smartphone Channel Scans.** In *Proceedings of the 2016 IEEE International Conference on Computer Communications (INFOCOM'16).*
- **2016:** Jinghao Shi, Edwin Santos, and Geoffrey Challen. **Why and How to Use PhoneLab.** In *GetMobile Mobile Computing and Communications Review, Volume 19 Issue 4.*
- **2015:** Tong Guan, Wen Dong, Dimitrios Koutsonikolas, Geoffrey Challen, and Chunming Qiao. **Robust, Cost-Effective and Scalable Localization in Large Indoor Areas.** In *Proceedings of the IEEE 2015 Global Telecommunications Conference (GLOBECOM'15).*
- **2015:** Anudipa Maiti, Yihong Chen, and Geoffrey Challen. **Jouler: A Policy Framework Enabling Effective and Flexible Smartphone Energy Management.** In *Proceedings of the Seventh International Conference on Mobile Computing, Applications and Services (MobiCASE'15).*
- **2015:** Rizwana Begum, Guru Prasad Srinivasa, David Werner, Geoffrey Challen, and Mark Hempstead. **Energy-Performance Trade-offs on Energy-Constrained Devices with Multi-Component DVFS.** In *Proceedings of the 2015 IEEE Symposium on Workload Characterization (IISWC'15).*
- **2015:** Jinghao Shi, Liwen Gui, Chunming Qiao, Dimitrios Koutsonikolas, and Geoffrey Challen. **A Little Sharing Goes a Long Way: The Case for Reciprocal Wifi Sharing.** In *Proceedings of the 2nd ACM Workshop on Hot Topics in Wireless (HotWireless'15).*

- **2015:** Oliver Kennedy, Jerry Ajay, Geoffrey Challen, and Luke Ziarek. **Pocket Data: The Need for TPC-MOBILE.** In *Proceedings of the 7th TPC Technology Conference on Performance Evaluation and Benchmarking (TPCTC'15).*
- **2015:** Geoffrey Challen, Jerry Ajay, Nick DiRienzo, Oliver Kennedy, Anudipa Maiti, Anandatirtha Nandugudi, Guru Prasad Srinivasa, Sriram Shantharam, Jinghao Shi, and Luke Ziarek. **`maybe` We Should Enable More Uncertain Mobile Systems Programming.** In *Proceedings of the Sixteenth Workshop on Mobile Computing Systems and Applications (HotMobile'15).*
- **2015:** Anudipa Maiti and Geoffrey Challen. **The Missing Numerator: Toward a Value Measure for Smartphone Apps.** In *Proceedings of the Sixteenth Workshop on Mobile Computing Systems and Applications (HotMobile'15).*
- **2015:** Jinghao Shi, Zhangyu Guan, Chunming Qiao, Tommaso Melodia, Dimitrios Koutsonikolas, and Geoffrey Challen. **Crowdsourcing Access Network Spectrum Allocation Using Smartphones** (Poster Abstract). In *Proceedings of the Sixteenth Workshop on Mobile Computing Systems and Applications (HotMobile'15).*
- **2014:** Nick DiRienzo and Geoffrey Challen. **Controlling Smartphone User Privacy via Objective-driven Context Mocking.** In *Proceedings of the Sixth International Conference on Mobile Computing, Applications and Services (MobiCASE'14).*
- **2014:** Anandatirtha Nandugudi, Carl Nuessle, Geoffrey Challen, Emiliano Miluzzo, and Yih-Farn Chen. **The PocketLocker Personal Cloud Storage System.** In *Proceedings of the Sixth International Conference on Mobile Computing, Applications and Services (MobiCASE'14).*
- **2014:** Jinghao Shi, Zhangyu Guan, Chunming Qiao, Tommaso Melodia, Dimitrios Koutsonikolas, and Geoffrey Challen. **Crowdsourcing Access Network Spectrum Allocation Using Smartphones.** In *Proceedings of the 13th ACM Workshop on Hot Topics in Networks (HotNets'14).*
- **2014:** Nick DiRienzo and Geoffrey Challen. **Should Smartphone Users Mock Apps?** In *Proceedings of the 6th ACM HotPlanet Workshop (HotPlanet'14).*
- **2014:** Anandatirtha Nandugudi, Taeyeon Ki, Carl Nuessle, and Geoffrey Challen. **PocketParker: Pocketsourcing Parking Lot Availability.** In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'14).*
- **2014:** Geoffrey Challen and Margo Seltzer. **Enabling MOOC Collaborations Through Modularity.** In *Proceedings of Learning with MOOCs: A Practitioner's Workshop (LWMOOC'14).*
- **2014:** Geoffrey Challen, Scott Haseley, Anudipa Maiti, Anandatirtha Nandugudi, Guru Prasad Srinivasa, Mukta Puri, and Junfei Wang. **The Mote is Dead. Long Live the**

Discarded Smartphone! In *Proceedings of the Fifteenth Workshop on Mobile Computing Systems and Applications (HotMobile'14)*.

- **2014:** Guru Prasad Srinivasa, Scott Haseley, Rizwana Begum, Mark Hempstead, and Geoffrey Challen. **New Interfaces for Achieving Power Agility on Mobile Devices** (Poster Abstract). In *Proceedings of the Fifteenth Workshop on Mobile Computing Systems and Applications (HotMobile'14)*.
- **2014:** Nick DiRienzo, Gino Buzzelli, and Geoffrey Challen. **Smartphone Users Want to Be Mocked** (Poster Abstract). In *Proceedings of the Fifteenth Workshop on Mobile Computing Systems and Applications (HotMobile'14)*. **Best Poster Award.**
- **2013:** Anandatirtha Nandugudi, Anudipa Maiti, Taeyeon Ki, Fatih Bulut, Murat Demirbas, Tevfik Kosar, Chunming Qiao, Steven Y. Ko, and Geoffrey Challen. **PhoneLab: A Large Programmable Smartphone Testbed** (Invited Paper). In *Proceedings of the First International Workshop on Sensing and Big Data Mining (SenseMine'13)*.
- **2013:** Sean Purdon, Branislav Kusy, Raja Jurdak, and Geoffrey Challen. **Model-Free HVAC Control Using Participant Feedback.** In *Proceedings of the Second IEEE International Workshop on Global Trends in Smart Cities (goSmart'13)*.
- **2013:** Anandatirtha Nandugudi, Anudipa Maiti, Fatih Bulut, Sonali Batra, Taeyeon Ki, Geoffrey Challen, Murat Demirbas, Steven Y. Ko, Tevfik Kosar, and Chunming Qiao. **Participant Behavior in PhoneLab.** In *Proceedings of the Third Conference on the Analysis of Mobile Phone Datasets (NetMob'13)*.
- **2011:** Geoffrey Challen and Mark Hempstead. **The Case for Power Agile Computing.** In *Proceedings of the 13th Workshop on Hot Topics in Operating Systems (HotOS'11)*.

Wireless Sensor Networks (Harvard, 2004–2010)

- **2010:** Geoffrey Challen, Jason Waterman, and Matt Welsh. **IDEA: Integrated Distributed Energy Awareness for Wireless Sensor Networks.** In *Proceedings of the 8th Annual International Conference on Mobile Systems, Applications and Services (MobiSys'10)*.
- **2009:** Konrad Lorincz, Bor-rong Chen, Geoffrey Challen, Atanu Roy Chowdhury, Shyamal Patel, Paolo Bonato, and Matt Welsh. **Mercury: A Wearable Sensor Network Platform for High-Fidelity Motion Analysis.** In *Proceedings of the Seventh ACM Conference on Embedded Networked Sensor Systems (SenSys'09)*.
- **2009:** Jason Waterman, Geoffrey Challen, and Matt Welsh. **Peloton: Coordinated Resource Management for Sensor Networks.** In *Proceedings of the 12th Workshop on Hot Topics in Operating Systems (HotOS'09)*.

Published under the name Geoffrey Werner-Allen:

- **2008:** Geoffrey Werner-Allen, Stephen Dawson-Haggerty, and Matt Welsh. **Lance: Optimizing High-Resolution Data Collection in Wireless Sensor Networks.** In *Proceedings of the Sixth ACM Conference on Embedded Networked Sensor Systems (SenSys'08)*.
- **2008:** Konrad Lorincz, Bor-rong Chen, Jason Waterman, Geoffrey Werner-Allen, and Matt Welsh. **Resource-Aware Programming in the Pixie OS.** In *Proceedings of the Sixth ACM Conference on Embedded Networked Sensor Systems (SenSys'08)*.
- **2008:** Konrad Lorincz, Bor-rong Chen, Jason Waterman, Geoffrey Werner-Allen, and Matt Welsh. **Pixie: An Operating System for Resource-Aware Programming of Embedded Sensors.** In *Proceedings of the Fifth Workshop on Embedded Networked Sensors (HotEmNets'08)*.
- **2006:** Geoffrey Werner-Allen, Konrad Lorincz, Jeff Johnson, Jonathan Lees, and Matt Welsh. **Fidelity and Yield in a Volcano Monitoring Sensor Network.** In *Proceedings of the Seventh USENIX Symposium on Operating Systems Design and Implementation (OSDI'06)*.
- **2006:** Geoffrey Werner-Allen, Konrad Lorincz, Mario Ruiz, Omar Marcillo, Jeff Johnson, Jonathan Lees, and Matt Welsh. **Deploying a Wireless Sensor Network on an Active Volcano.** In *IEEE Internet Computing*, Special Issue on Data-Driven Applications in Sensor Networks, March/April 2006.
- **2005:** Geoffrey Werner-Allen, Geetika Tewari, Ankit Patel, Radhika Nagpal, and Matt Welsh. **Firefly-Inspired Sensor Network Synchronicity with Realistic Radio Effects.** In *Proceedings of the Third ACM Conference on Embedded Networked Sensor Systems (SenSys'05)*.
- **2005:** Geoffrey Werner-Allen, Pat Swieskowski, and Matt Welsh. **MoteLab: A Wireless Sensor Network Testbed.** In *Proceedings of the Fourth International Conference on Information Processing in Sensor Networks (IPSN'05)*.
- **2005:** Geoffrey Werner-Allen, Jeff Johnson, Mario Ruiz, Jonathan Lees, and Matt Welsh. **Monitoring Volcanic Eruptions with a Wireless Sensor Network.** In *Proceedings of the Second European Workshop on Wireless Sensor Networks (EWSN'05)*.
- **2004:** Victor Shnayder, Mark Hempstead, Bor-rong Chen, Geoffrey Werner-Allen, and Matt Welsh. **Simulating the Power Consumption of Large-Scale Sensor Network Applications.** In *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys'04)*.

Edited Volumes and Book Chapters

- **2010:** Elena Gaura, Mike Allen, Lewis Girod, James Brusey, and Geoffrey Challen (editors). **Wireless Sensor Networks: Deployments and Design Frameworks.** Springer, 2010.

- **2010:** Geoffrey Challen and Matt Welsh. **Volcano Monitoring: Addressing Data Quality Through Iterative Deployment.** In *Wireless Sensor Networks: Deployments and Design Frameworks*, Springer, 2010.

Presentations

- **June 29, 2026**: Workshop on the Influence of AI on Education in Computer Science and Engineering, Constructor University, Bremen (virtual): *The New Computing*
- **June 11, 2026**: GitHub Education Educator Summit and the Illinois Computer Science Summer Teaching Workshop: *Using and Understanding AI*
- **June 9, 2026**: GitHub Education Educator Summit (GitHub HQ): *The New Computing*
- **May 12, 2026**: Georgia Tech School of Computing Instruction: *The New Computing: Education in the AI Era*
- **February 11, 2026**: University of Sydney: *The Educational Engineer*
- **February 27, 2025**: SIGCSE 2025: *Accelerating Accurate Assignment Authoring Using Solution-Generated Autograders* (with Ben Nordick)
- **February 27, 2025**: SIGCSE 2025: *Investigating the Presence and Development of Student Instructor Preferences in a Large-Scale CS1 Course* (with Yiqiu Zhou and Luc Paquette)
- **March 21, 2024**: SIGCSE 2024 Panel: *Interviewing the Teaching Faculty Hiring Process* (with Victoria Dean, Nate Derbinsky, Matt Wang, and Jacqueline Smith). DOI: 10.1145/3626253.3631664
- **June 6, 2023**: Illinois Computer Science Summer Teaching Workshop (iCSTWS): *The Dynamic Evolution of Student Preferences Towards Instructors in a CS1 Course: A Clickstream Data Analysis* (Slides)
- **June 5, 2023**: Illinois Computer Science Summer Teaching Workshop (iCSTWS): *Guided by Complexity: Automated Code Quality Feedback in CS1* (Slides)
- **August 16, 2022**: Illinois Computer Science Summer Teaching Workshop (iCSTWS): *All the Reasons for Frequent Small Assessment* (Video, Slides)
- **May 6, 2022**: Western Canadian Conference on Computing Education (WCCCE): *How the Pandemic (Permanently) Transformed My Teaching* (Video, Slides)
- **April 28, 2015**: Rochester Institute of Technology Computer Science Seminar: *Building Less Certain Mobile Apps* (invited by Peizhao Hu)

Funding

Across my career I have received almost $3M in competitive external funding, primarily from the National Science Foundation and Google, most as PI. Full copies of my funded grant proposals are available on my previous group's website.

Computer Science Education (Illinois)

- **2021–2022**: Interactive Code Walkthroughs, $10K from the Grainger College of Engineering through the Strategic Instructional Innovations Program (SIIP). PI with Tiffani Williams and Michael Nowak.
- **2020–2022**: Revising the CS Introductory Programming Sequence, $8.5K from the Grainger College of Engineering through the Strategic Instructional Innovations Program (SIIP). Co-PI with Michael Nowak, Carl Evans, Margaret Fleck, Michael Woodley, and Craig Zilles.

Mobile Systems Research (University at Buffalo)

- **2017–2022**: CAREER: Harnessing Implementation Flexibility to Enable Runtime Adaptation, $525,582 from the National Science Foundation. Sole PI.
- **2016–2017**: CI-SUSTAIN: Collaborative Research: Sustaining Successful Smartphone Testbeds to Enable Diverse Mobile Experiments, $75,000 from the National Science Foundation. Co-PI with Chunming Qiao.
- **2016–2018**: Enabling Pocket-Scale Data Management Research, $100,000 from the National Science Foundation. Co-PI with Oliver Kennedy and Luke Ziarek.
- **2015–**: Expressing Uncertainty Using the `maybe` System, $37,156 from Google. PI with Co-PIs Oliver Kennedy and Luke Ziarek.
- **2014–2019**: Jouler: A Cross-Device Application Energy Management Framework for Smartphones, $499,185 from the National Science Foundation. Sole PI.
- **2014–2018**: Architecture and System Support for Power-Agile Computing, $561,766 ($282,930 to UB) from the National Science Foundation. Co-PI with Mark Hempstead.
- **2012–2016**: PhoneLab: A Programmable Participatory Smartphone Testbed, $1,322,510 from the National Science Foundation (plus a $36,000 REU supplement awarded in 2013). PI with co-PIs Steven Y. Ko, Murat Demirbas, Tevfik Kosar, and Chunming Qiao.
- **2011–2012**: PhoneLab: A Participatory Smartphone Cloud Testbed, $60,994 from Google. Co-PI with Steven Y. Ko, Murat Demirbas, Tevfik Kosar, and Chunming Qiao.

Advising

At the University at Buffalo I graduated one Ph.D. student and supervised 47 other students, including Ph.D. students, Masters students, undergraduates, a visiting Ph.D. student, and the PhoneLab system administrator.

Ph.D. Students

- Anandatirtha Nandugudi (2012–2015) — Wifi streaming, PocketParker, PocketLocker, and smartphone sustainability; co-advised with Chunming Qiao. *(graduated; my only Ph.D.*

graduate)

- Anudipa Maiti (2012–) — Jouler energy management and smartphone app value estimation.
- Guru Prasad Srinivasa (2013–) — power-agile operating systems and thermal management for mobile devices.
- Jinghao Shi (2013–) — wireless protocol validation with Microsoft Research and PhoneLab platform maintenance; co-advised with Chunming Qiao.
- Carl Nuessle (2013–) — smartphone file system access patterns.
- Ali Ben Ali (2015–) — tracking use of multiple personal devices using a smartwatch.
- Scott Haseley (2015–) — improving smartphone quality of experience; built the `test161` OS/161 testing tool.
- Sonali Batra (2011–2013) — PhoneLab and rogue wireless access points; continued the Ph.D. program under Tevfik Kosar.
- Taeyeon Ki (2012–2014) — the PhoneLab interface and PocketParker; continued the Ph.D. program under Steven Y. Ko.
- Jerry Ajay (2015–2016) — smartphone database access patterns; continued the Ph.D. program under Wenyao Xu.
- Yihong Chen (2015–2016) — the `maybe` uncertainty-based adaptation project and Jouler; now at Twitter.

Masters Students

- Micheal Benedict (2011–2012) — PhoneLab prototype; now at Twitter.
- Vinu Charanya Athangudi Purushothaman (2011–2012) — PhoneLab prototype; now at Twitter.
- Rajeshwari Adapalam (2012) — smartphone access-point impersonation; now at Cisco.
- Manoj Mylapore Chandrasekaran (2012–2013) — PhoneLab; now at Cerner.
- Anuja Raval (2012) — power-agile operating systems; now at Citi.
- Bhaavya Kapoor (2012) — PhoneLab.
- Denise Blady (2013) — PocketMocker context-mocking prototype; now at DISA.
- Eric Lehner (2013) — PocketMocker prototype; now at MITRE.
- Gino Buzzelli (2013) — PocketMocker prototype; now at Microsoft.
- Agrim Nigam (2013) — smartphone disaster preparedness; now at Holistic Labs.
- John Gerber (2014) — city-scale urban monitoring using car-mounted discarded smartphones.
- Nishanth Vasisht (2014) — smartphone disaster preparedness; now at Amazon.
- Scott Haseley (2014) — power-agile operating systems and reusing discarded smartphones; began the Ph.D. program in 2015.
- Jerry Ajay (2015) — the `maybe` uncertainty project; entered the Ph.D. program in Fall 2015.

- Sriram Shantharam (2014–2015) — city-scale car-mounted sensor network and disaster preparedness; now at Bak.
- Rakesh Balasubramanian (2015) — city-scale sensing and Wifi adaptation; now at FactSet.
- Ramya Rao (2015) — the `maybe` uncertainty-based adaptation project; now at FactSet.
- Miraj Kheni (2015–2016) — smartphone app database performance.

Undergraduates

- Brijesh Rakholia (2015–) — smartphone quality of experience; internet-class.org UTA.
- Edwin Santos (2015–) — Android Open Source Project comparison and PhoneLab maintenance.
- Kyle Schoener (2015–) — Jouler energy management; internet-class.org UTA.
- Wesley Csendom (2015–) — internet-class.org course website and tools.
- Aishani Bhalla (2015–) — online quiz tool for website-based courses; internet-class.org UTA.
- Greg Bunyea (2016–) — head UTA and content/tools for internet-class.org.
- Grant Wrazen (2016–) — smartphone database performance and Wifi optimization.
- Vighnesh Iyer (2016–) — Wifi network optimization; internet-class.org UTA.
- Lakshmi Ethiraj (2016–) — smartphone database performance; internet-class.org UTA.
- Sean Zawicki (2012) — PhoneLab; now at Voxer.
- Mitch Nguyen (2012) — PhoneLab.
- Frank Rossi (2013–2014) — PocketLocker distributed storage.
- Nick DiRienzo (2013–2015) — interface-event bug and latency detection and PocketMocker; now at Optimizely.
- Michael Ferris (2014–2015) — next-generation file system interfaces for mobile devices.
- Gino Notto (2015) — detecting transitions between personal devices.
- Er An Khoo (2015) — interface-event bug and latency detection.
- John Cherry (2015) — smartphone file system access patterns.
- Corey Kress (2015) — internet-class.org course website and tools.

Visiting Student and Research Staff

- Aslak Johansen (2013) — visiting Ph.D. student from ITU Copenhagen; interfaced new sensors to PhoneLab devices.
- Maulik Dave (2013–2015) — PhoneLab testbed administrator.

Service

For more details, please review my service statement.

Illinois

- **2023–:** Chair of the Instructional Computing Committee
- **2023–:** Service on the Grainger College of Engineering New Course and Curricular Review Process Committee
- **2022–2024:** Service on the University General Education Board
- **2022–:** Published a freely-available public version of my CS1 materials at `learncs.online`
- **2021–2022:** Served as the first teaching faculty chair of the teaching faculty hiring committee
- **2021:** Helped organize the inaugural Illinois Computer Science Summer Teaching Workshop
- **2017–:** Service on departmental committees including Undergraduate Study, Academic Appeals, Broadening Participation in Computing, and others

University at Buffalo

- **2015–2017:** Chair, CS Curriculum Subcommittee—led revisions to the undergraduate computer science curriculum.
- **2015–2017:** Member, Introductory Course Subcommittee—proposed a new introductory computer science course.
- **2015–2017:** Member of the Diversity, Faculty Search, and Undergraduate Affairs Committees.
- **2015–2016:** Created the CSE Faculty Choice Award, a faculty-funded award recognizing a graduating senior (awarded to Nick DiRienzo in 2015 and Wendy Jansson in 2016).
- **2014–2017:** Founded and advised the UB chapter of the Scientista Foundation, a national organization promoting women in STEM, and led a delegation to the 2015 Grace Hopper Celebration of Women in Computing; also served as Faculty Adviser to the ACM Student Chapter (2012–).
- **2014:** Organized the creation of a CS Diversity Mural—following a fundraising effort among colleagues and a student design competition, the winning image of Grace Hopper working on COBOL was installed in the department.
- **2011–2017:** Mentored undergraduate researchers from underrepresented groups through the SUNY Louis Stokes Alliance for Minority Participation (LSAMP) program.
- **2011–2017:** Founded and organized a weekly systems research seminar and a young-faculty lunch.
- **2011–2015:** Member of the Grievance, Faculty Search, Graduate Admissions, and Colloquium Committees.
- **2011–2014:** Judge, UB Hackathon (2012, 2014); Organizer, CSEd Week Puzzle Hunt (2011); Moderator, Your Passport to Employment (2011–2013).
- **Invited participant:** Bloomberg Engineering University Day (2016), Google Faculty Summit on Mobile (2015), the SmartAmerica Challenge kickoff at the White House complex

representing PhoneLab (2013), Google Faculty Summit (2011), and Visiting Researcher at ITU Copenhagen (invited by Philippe Bonnet, 2010).
- **Conference organization:** Local Arrangements Chair, SenSys'12; Student Travel Grant Chair, SenSys'11.

Reviewing

- **Program committees:** Served on technical program committees and as a reviewer for top mobile-systems and sensor-networks venues, including MobiCom'17, MobiSys'14/'16/'17, HotMobile'16, SenSys'13/'16, IPSN'15/'17, MobiCASE'16, MASS'15, ICDCS'14, EWSN'13, RealWSN'15, TridentCom'15, EMSOFT 2012, RTSS'11, and WPA'17. I was the first University at Buffalo faculty member invited to serve on several of these top-tier program committees.
- **Reviewing practice:** A vocal advocate for signing reviews ("reverse-blind review") to make peer review more accountable.
- **Editorships:** Column Editor (2015–), GetMobile, the ACM SIGMOBILE magazine.
- **Ad-hoc journal reviewing:** ACM Transactions on Sensor Networks (TOSN), IEEE/ACM Transactions on Networking (ToN), IEEE Transactions on Mobile Computing (TMC), and IEEE Transactions on Parallel and Distributed Systems (TPDS).

Mentoring

- **2020–** Mentored Assistant Teaching Professor Brad Solomon
- **2020–** Mentored Assistant Teaching Professor Michael Nowak
- **2017–** Mentored dozens of senior staff members for CS 125/4, and supervised over 2000 undergraduate and graduate course staff

References

1. **Nate Derbinsky**, Teaching Professor, Khoury College of Computer Sciences, Northeastern University (n.derbinsky@northeastern.edu)
2. **Margaret Fleck**, Teaching Professor, University of Illinois (mfleck@illinois.edu)
3. **Cinda Heeren**, Professor of Teaching, University of British Columbia (cheeren@cs.ubc.ca)
4. **Margo Seltzer**, Canada 150 Research Chair and Professor, University of British Columbia (mseltzer@cs.ubc.ca)
5. **Tiffani Williams**, Teaching Professor and Director of Onramp Programs, University of Illinois (tiffani@illinois.edu)
6. **Craig Zilles**, Professor, University of Illinois (zilles@illinois.edu)