Philip Guo

Assistant Professor
Department of Cognitive Science
University of California, San Diego (UCSD)

Updated: March 21, 2017
Email: philip@pgbovine.net
www.pgbovine.net

RESEARCH INTERESTS

Human-computer interaction, online learning, productivity tools for programmers and data scientists, computing education

ACADEMIC POSITIONS

07/2016 – **University of California, San Diego (UCSD)**, La Jolla, CA
Assistant Professor of Cognitive Science
Faculty affiliate in Computer Science and Engineering

07/2014 – 06/2016 **University of Rochester**, Rochester, NY
Assistant Professor of Computer Science

EDUCATION

09/2006 – 06/2012 **Stanford University**, Stanford, CA
Ph.D. in Computer Science
Dissertation: *Software Tools to Facilitate Research Programming*
Advisor: Dawson Engler

06/2005 – 06/2006 **Massachusetts Institute of Technology**, Cambridge, MA
Master of Engineering in Electrical Engineering and Computer Science
Master's Thesis: *A Scalable Mixed-Level Approach to Dynamic Analysis of C and C++ Programs*, Advisor: Michael D. Ernst
MIT EECS award for Outstanding Computer Science Master of Engineering Thesis

09/2001 – 06/2005 **Massachusetts Institute of Technology**, Cambridge, MA
Bachelor of Science in Electrical Engineering and Computer Science
GPA: 5.0/5.0

AWARDS AND HONORS

05/2017 CHI Honorable Mention Paper [C.31]

08/2015 Google Faculty Research Award

04/2014 CHI Honorable Mention Paper [C.18]

06/2012 ICSE Software Engineering In Practice Best Paper Award [C.13]

07/2009 ACM SIGSOFT Distinguished Paper Award [C.6]

04/2009 CHI Honorable Mention Paper [C.3]

09/2009 – 06/2011 National Science Foundation (NSF) Graduate Fellowship

09/2006 – 09/2009 National Defense Science and Engineering (NDSEG) Graduate Fellowship

05/2006 MIT Charles and Jennifer Johnson Thesis Award for Outstanding Computer Science Master of Engineering Thesis

PRIOR EMPLOYMENT

07/2015 – 08/2015 **Microsoft Research**, Redmond, WA
Visiting Researcher – Research in Software Engineering (RiSE) group

10/2013 – 06/2014 **Massachusetts Institute of Technology**, Cambridge, MA
Postdoctoral Researcher – CSAIL User Interface Design Group – Host: Rob Miller

06/2013 – 09/2013 **edX**, Cambridge, MA
Visiting Research Scientist – analyzed MOOC data [C.15, C.16, C.17]

07/2012 – 02/2013 **Google**, Mountain View, CA
Software Engineer – online education group – Google Research

09/2006 – 06/2012 **Stanford University**, Stanford, CA
Ph.D. Student – Department of Computer Science

09/2011 – 01/2012 **Harvard University**, Cambridge, MA
Visiting Research Fellow – Computer Systems Group – Host: Margo Seltzer

06/2011 – 09/2011 **Google**, Mountain View, CA
Software Engineering Intern – refined and deployed CDE [C.9, C.12, M.2, B.2]

06/2009 – 09/2009 **Microsoft Research**, Redmond, WA
Research Intern – Research in Software Engineering (RiSE) group

06/2007 – 09/2007 **Google**, Mountain View, CA
Software Engineering Intern – prototyped memory allocators for C and C++ programs

01/2004 – 06/2006 **Massachusetts Institute of Technology**, Cambridge, MA
Research Assistant – Program Analysis Group – Advisor: Michael D. Ernst
Undergraduate and master's research on tools for analyzing C and C++ programs

09/2003 – 01/2004 **Massachusetts Institute of Technology**, Cambridge, MA
Research Assistant – Computer Graphics Group – Advisor: Fredo Durand
Developed an HDR (high dynamic range) image editing tool for photographers

06/2004 – 08/2004, 06/2003 – 08/2003 **Teradyne**, Agoura Hills, CA
Software Engineering Intern – wrote simulators for semiconductor test hardware

09/2002 – 06/2003 **Massachusetts Institute of Technology**, Cambridge, MA
Research Assistant – Teacher Education Program – Advisor: Eric Klopfer
Developed a suite of 5 educational games for Palm OS devices

06/2002 – 08/2002 **Codehost**, Culver City, CA
Software Engineering Intern – wrote embedded Linux tablet PC software

FUNDING

National Science Foundation. CRII: CHS: Scaling Up Online Peer Tutoring of Computer Programming. $175,000 (sole PI, 2015–2017)

NSF CRII REU (Research Experiences for Undergraduates) supplement. $32,000

Google Faculty Research Award. Enabling Learners to Create Hierarchical Tutorials from How-To Videos on YouTube. $64,295 (sole PI, 2015)

University of Rochester. University Research Award: Enabling Fast and Scalable Feedback on Writing. $50,000 (sole PI, 2015)

Microsoft Research. Online Python Tutor for Office Mix. $61,308 (sole PI, 2014)

PEER-REVIEWED PUBLICATIONS

In many areas within computer science and human-computer interaction, *conferences* (not journals) are the primary venues for peer-reviewed publications.

Conference Papers

C.31 **Philip J. Guo**. Older Adults Learning Computer Programming: Motivations, Frustrations, and Design Opportunities. In Proceedings of CHI 2017: *ACM Conference on Human Factors in Computing Systems*, May 2017. (**Honorable Mention Paper**)

C.30 Jeremy Warner and **Philip J. Guo**. CodePilot: Scaffolding End-to-End Collaborative Software Development for Novice Programmers. In Proceedings of CHI 2017: *ACM Conference on Human Factors in Computing Systems*, May 2017.

C.29 Denae Ford, Justin Smith, **Philip J. Guo**, Chris Parnin. Paradise Unplugged: Identifying Barriers for Female Participation on Stack Overflow. In Proceedings of FSE 2016: *ACM SIGSOFT International Symposium on the Foundations of Software Engineering*, Nov 2016.

C.28 Parmit K. Chilana, Rishabh Singh, **Philip J. Guo**. Understanding Conversational Programmers: A Perspective from the Software Industry. In Proceedings of CHI 2016: *ACM Conference on Human Factors in Computing Systems*, May 2016.

C.27 **Philip J. Guo**. Codeopticon: Real-Time, One-To-Many Human Tutoring for Computer Programming. In Proceedings of UIST 2015: *ACM Symposium on User Interface Software and Technology*, Nov 2015.

C.26 **Philip J. Guo**, Jeffery White, Renan Zanelatto. Codechella: Multi-User Program Visualizations for Real-Time Tutoring and Collaborative Learning. In Proceedings of VL/HCC 2015: *IEEE Symposium on Visual Languages and Human-Centric Computing*, Oct 2015.

C.25 Mitchell Gordon and **Philip J. Guo**. Codepourri: Creating Visual Coding Tutorials Using A Volunteer Crowd Of Learners. In Proceedings of VL/HCC 2015: *IEEE Symposium on Visual Languages and Human-Centric Computing*, Oct 2015.

C.24 Joyce Zhu, Jeremy Warner, Mitchell Gordon, Jeffery White, Renan Zanelatto, **Philip J. Guo**. Toward a Domain-Specific Visual Discussion Forum for Learning Computer Programming: An Empirical Study of a Popular MOOC Forum. In Proceedings of VL/HCC 2015: *IEEE Symposium on Visual Languages and Human-Centric Computing*, Oct 2015.

C.23 Parmit K. Chilana, Celena Alcock, Shruti Dembla, Anson Ho, Ada Hurst, Brett Armstrong, **Philip J. Guo**. Perceptions of Non-CS Majors in Intro Programming: The Rise of the Conversational Programmer. In Proceedings of VL/HCC 2015: *IEEE Symposium on Visual Languages and Human-Centric Computing*, Oct 2015.

C.22 Jeremy Warner, John Doorenbos, Bradley N. Miller, **Philip J. Guo**. How High School, College, and Online Students Differentially Engage with an Interactive

Digital Textbook. Short paper in Proceedings of EDM 2015: *International Conference on Educational Data Mining*, June 2015.

C.21 Carrie J. Cai, **Philip J. Guo**, James Glass, Robert C. Miller. Wait-Learning: Leveraging Wait Time for Second Language Education. In Proceedings of CHI 2015: *ACM Conference on Human Factors in Computing Systems*, April 2015.

C.20 Juho Kim, **Philip J. Guo**, Carrie J. Cai, Shang-Wen (Daniel) Li, Krzysztof Z. Gajos, Robert C. Miller. Data-Driven Interaction Techniques for Improving Navigation of Educational Videos. In Proceedings of UIST 2014: *ACM Symposium on User Interface Software and Technology*, October 2014.

C.19 Jeremy Scott, **Philip J. Guo**, Randall Davis. A Direct Manipulation Language for Explaining Algorithms. Short paper in Proceedings of VL/HCC 2014: *IEEE Symposium on Visual Languages and Human-Centric Computing*, Jul 2014.

C.18 Juho Kim, Phu Nguyen, Sarah Weir, **Philip J. Guo**, Robert C. Miller, Krzysztof Z. Gajos. Crowdsourcing Step-by-Step Information Extraction to Enhance Existing How-to Videos. In Proceedings of CHI 2014: *ACM Conference on Human Factors in Computing Systems*, April 2014.
(**Honorable Mention Paper**)

C.17 **Philip J. Guo** and Katharina Reinecke. Demographic Differences in How Students Navigate Through MOOCs. In Proceedings of L@S 2014: *ACM Conference on Learning at Scale*, March 2014.

C.16 **Philip J. Guo**, Juho Kim, Rob Rubin. How Video Production Affects Student Engagement: An Empirical Study of MOOC Videos. In Proceedings of L@S 2014: *ACM Conference on Learning at Scale*, March 2014.

C.15 Juho Kim, **Philip J. Guo**, Daniel T. Seaton, Piotr Mitros, Krzysztof Z. Gajos, Robert C. Miller. Understanding In-Video Dropouts and Interaction Peaks in Online Lecture Videos. In Proceedings of L@S 2014: *ACM Conference on Learning at Scale*, March 2014.

C.14 **Philip J. Guo**. Online Python Tutor: Embeddable Web-Based Program Visualization for CS Education. In Proceedings of SIGCSE 2013: *ACM Technical Symposium on Computer Science Education*, March 2013.

C.13 Thomas Zimmermann, Nachiappan Nagappan, **Philip J. Guo**, Brendan Murphy. Characterizing and Predicting Which Bugs Get Reopened. In Proceedings of ICSE 2012: *ACM/IEEE International Conference on Software Engineering*, Software Engineering In Practice track, June 2012.
(**Best Paper Award**)

C.12 **Philip J. Guo**. CDE: Run Any Linux Application On-Demand Without Installation. In Proceedings of LISA 2011: *USENIX Large Installation System Administration Conference*, December 2011.

C.11 **Philip J. Guo**, Sean Kandel, Joseph M. Hellerstein, Jeffrey Heer. Proactive Wrangling: Mixed-Initiative End-User Programming of Data Transformation Scripts. In Proceedings of UIST 2011: *ACM Symposium on User Interface Software and Technology*, October 2011.

C.10 **Philip J. Guo** and Dawson Engler. Using Automatic Persistent Memoization to Facilitate Data Analysis Scripting. In Proceedings of ISSTA 2011: *ACM International Symposium on Software Testing and Analysis*, July 2011.

C.9 **Philip J. Guo** and Dawson Engler. CDE: Using System Call Interposition to Automatically Create Portable Software Packages. Short paper in Proceedings of USENIX 2011: *USENIX Annual Technical Conference*, June 2011.

C.8 **Philip J. Guo**, Thomas Zimmermann, Nachiappan Nagappan, Brendan Murphy. "Not My Bug!" and Other Reasons for Software Bug Report Reassignments. In Proceedings of CSCW 2011: *ACM Conference on Computer Supported Cooperative Work*, March 2011.

C.7 **Philip J. Guo**, Thomas Zimmermann, Nachiappan Nagappan, Brendan Murphy. Characterizing and Predicting Which Bugs Get Fixed: An Empirical Study of Microsoft Windows. In Proceedings of ICSE 2010: *ACM/IEEE International Conference on Software Engineering*, May 2010.

C.6 Adam Kiezun, Vijay Ganesh, **Philip J. Guo**, Pieter Hooimeijer, Michael D. Ernst. HAMPI: A Solver for String Constraints. In Proceedings of ISSTA: *ACM International Symposium on Software Testing and Analysis*, July 2009. (**ACM SIGSOFT Distinguished Paper Award**)

C.5 **Philip J. Guo** and Dawson Engler. Linux Kernel Developer Responses to Static Analysis Bug Reports. Short paper in Proceedings of USENIX 2009: *USENIX Annual Technical Conference*, June 2009.

C.4 Adam Kiezun, **Philip J. Guo**, Karthick Jayaraman, Michael D. Ernst. Automatic Creation of SQL Injection and Cross-site Scripting Attacks. In Proceedings of ICSE 2009: *ACM/IEEE International Conference on Software Engineering*, May 2009.

C.3 Joel Brandt, **Philip J. Guo**, Joel Lewenstein, Mira Dontcheva, Scott R. Klemmer. Two Studies of Opportunistic Programming: Interleaving Web Foraging, Learning, and Writing Code. In Proceedings of CHI 2009: *ACM Conference on Human Factors in Computing Systems*, April 2009. (**Honorable Mention Paper**)

C.2 **Philip J. Guo**, Jeff H. Perkins, Stephen McCamant, Michael D. Ernst. Dynamic Inference of Abstract Types. In Proceedings of ISSTA 2006: *ACM International Symposium on Software Testing and Analysis*, July 2006.

C.1 Brian Demsky, Michael D. Ernst, **Philip J. Guo**, Stephen McCamant, Jeff H. Perkins, Martin Rinard. Automatic Inference and Enforcement of Data Structure Consistency Specifications. In Proceedings of ISSTA 2006: *ACM International Symposium on Software Testing and Analysis*, July 2006.

Journal Articles

J.3 Elena L. Glassman, Jeremy Scott, Rishabh Singh, **Philip J. Guo**, Robert C. Miller. OverCode: Visualizing Variation in Student Solutions to Programming Problems at Scale. In TOCHI: *ACM Transactions on Computer-Human Interaction*, 2015.

J.2 Adam Kiezun, Vijay Ganesh, Shay Artzi, **Philip J. Guo**, Pieter Hooimeijer, Michael D. Ernst. Hampi: A Solver for Word Equations over Strings, Regular Expressions and Context-free Grammars. In TOSEM: *ACM Transactions on Software Engineering Methodology*, 2012.

J.1 Michael D. Ernst, Jeff H. Perkins, **Philip J. Guo**, Stephen McCamant, Carlos Pacheco, Matthew S. Tschantz, Chen Xiao. The Daikon system for dynamic detection of likely invariants. In *Science of Computer Programming*, 2007.

Workshop Papers

W.4 **Philip J. Guo** and Margo Seltzer. Burrito: Wrapping Your Lab Notebook in Computational Infrastructure. In Proceedings of TaPP 2012: *USENIX Workshop on the Theory and Practice of Provenance*, June 2012.

W.3 **Philip J. Guo**. Sloppy Python: Using Dynamic Analysis to Automatically Add Error Tolerance to Ad-Hoc Data Processing Scripts. In Proceedings of WODA 2011: *ACM International Workshop on Dynamic Analysis*, July 2011.

W.2 **Philip J. Guo** and Dawson Engler. Towards Practical Incremental Recomputation for Scientists: An Implementation for the Python Language. In Proceedings of TaPP 2010: *USENIX Workshop on the Theory and Practice of Provenance*, February 2010.

W.1 Joel Brandt, **Philip J. Guo**, Joel Lewenstein, Scott R. Klemmer. Opportunistic Programming: How Rapid Ideation and Prototyping Occur in Practice. In WEUSE 2008: *ACM Workshop on End-User Software Engineering*, May 2008.

Posters and Works-in-Progress

P.3 Elena L. Glassman, Jeremy Scott, Rishabh Singh, **Philip J. Guo**, Robert C. Miller. OverCode: Visualizing Variation in Student Solutions to Programming Problems at Scale. Poster in Proceedings of UIST 2014: *ACM Symposium on User Interface Software and Technology*, October 2014.

P.2 Carrie J. Cai, **Philip J. Guo**, James Glass, Robert C. Miller. Wait-Learning: Leveraging Conversational Dead Time for Second Language Education. In Proceedings of CHI 2014: *ACM Conference on Human Factors in Computing Systems*, April 2014.

P.1 Anvisha Pai, **Philip J. Guo**, Robert C. Miller. Modeling Programming Knowledge for Mentoring at Scale. In Proceedings of L@S 2014: *ACM Conference on Learning at Scale*, March 2014.

INVITED PUBLICATIONS

Magazine Articles

M.10 **Philip J. Guo**. Learning Programming at Scale. In *O'Reilly Radar*, Aug 2015.

M.9 **Philip J. Guo**. Refining Students' Coding and Reviewing Skills. In *Communications of the ACM*, Vol. 57, No. 9, Sep 2014.

M.8 **Philip J. Guo**. The Difficulty of Teaching Programming Languages, and the Benefits of Hands-on Learning. In *Communications of the ACM*, Vol. 57, No. 7, Jul 2014. (appeared alongside an article by Mark Guzdial)

M.7 **Philip J. Guo**. Clarifying Human-Computer Interaction. In *Communications of the ACM*, Vol. 57, No. 2, Feb 2014.

M.6 **Philip J. Guo**. Silent Technical Privilege. In *Slate*, Jan 2014.

M.5 **Philip J. Guo**. Helping scientists, engineers to work up to 100 times faster. In *Communications of the ACM*, Vol. 56, No. 10, Oct 2013.

M.4 **Philip J. Guo**. Teaching Programming the Way It Works Outside the Classroom. In *Communications of the ACM*, Vol. 56, No. 8, Aug 2013.

M.3 **Philip J. Guo**. Lessons from the Grind: How unglamorous grunt work can lead to creative innovation. In *MIT Technology Review*, Jan 2013.

M.2 **Philip J. Guo**. CDE: A Tool For Creating Portable Experimental Software Packages. In *Computing in Science and Engineering: Special Issue on Software for Reproducible Computational Science*, Jul/Aug 2012.

M.1 Joel Brandt, **Philip J. Guo**, Joel Lewenstein, Mira Dontcheva, Scott R. Klemmer. Opportunistic Programming: Writing Code to Prototype, Ideate, and Discover. In *IEEE Software: Special Issue on End-User Software Engineering*, Sep/Oct 2009.

BOOK CHAPTERS

B.3 **Philip J. Guo**. Parse that data! Practical Tips for preparing your raw data for analysis. Book chapter in *Perspectives on Data Science for Software Engineering*, T. Menzies, L. Williams, T. Zimmermann, eds. Morgan Kaufmann, 2016.

B.2 **Philip J. Guo**. CDE: Automatically Package and Reproduce Computational Experiments. Book chapter in *Implementing Reproducible Research*, V. Stodden, F. Leisch, R. Peng, eds. Taylor & Francis Group, 2013.

B.1 Joel Brandt, **Philip J. Guo**, Joel Lewenstein, Mira Dontcheva, Scott R. Klemmer. How the Web Helps People Turn Ideas Into Code. Book chapter in *No Code Required: Giving Users Tools to Transform the Web*, A. Cypher, M. Dontcheva, T. Lau, J. Nichols, eds. Morgan Kaufmann, 2010.

INVITED PAPERS

IP.1 Quanzeng You, Jianbo Yuan, Jiaqi Wang, **Philip J. Guo**, Jiebo Luo. Snap n' Shop: Visual Search-Based Mobile Shopping Made a Breeze by Machine and Crowd Intelligence. In Proceedings of ICSC 2015: *IEEE International Conference on Semantic Computing*, Feb 2015.

INVITED TALKS

- The Design Space of Tools for Learning Programming at Scale
  UCSD Design at Large Seminar, October 2016.
- Interactive Systems for Learning Programming at Scale
  (faculty candidate job talk)
  Northwestern University EECS and School of Education and Social Policy, UCSD Cognitive Science, CU Boulder CS, Yale University CS, UCLA CS, UC Berkeley School of Information, UCSD Computer Science & Engineering, Jan–Apr 2016.
- Interactive Systems for Learning Programming at Scale. *Stanford Human-Computer Interaction Seminar*, Stanford, CA, Feb 2016; *Recurse Center (née Hacker School)*, New York, NY, Jan 2016; *University of Maryland*, College Park, MD, Dec 2015.
- Invited panelist on tools for personalized education, *CCC visioning workshop on Computer-Aided Personalized Education*, Washington, D.C., Nov 2015.
- Learning Programming at Scale. *University of Rochester Laboratory for Laser Energetics*, Rochester, NY, Sep 2015; *Microsoft Research*, Redmond, WA, Aug 2015; *University of Washington DUB seminar*, Seattle, WA, July 2015.
- Online Python Tutor: A 5-Year Retrospective.
  Union College, Rensselaer Polytechnic Institute (RPI), New York, Oct 2014.
- How to effectively ask for help as a junior employee.
  MIT 6.UAT guest lecture, Cambridge, MA, Nov 2013.
- Hacking the Ph.D.: Three Serendipitous Projects.
  Hacker School, New York, NY, Nov 2013.
- Why Pursue A Ph.D.? Three Practical Reasons. *Amherst College, UMass Amherst, Brown, MIT, Harvard, Tufts.* Oct–Nov 2013.

- Challenges in Teaching Python Programming: Vocabulary, Meaning, and Idioms. *MIT Lincoln Laboratory*, Lexington, MA, Oct 2013.
- Twenty Lessons From The Ph.D. Grind.
  Keynote at the MIT CSAIL Student Workshop, Oct 2013.
- Software Tools for Research Programming. *Boston University*, MA, Sep 2013.
- Programming On Demand: Wrangling, Iterating, and Opportunistic Learning. (faculty candidate job talk, all in CS or EECS departments)
  University of Utah, North Carolina State University, Dartmouth College, University of San Francisco, Oregon State University, Northeastern University, University of Rochester, Washington University in St. Louis, Feb–Mar 2013.
- Online Python Tutor: Web-Based Program Visualization for CS Education. *Sonoma State University – Computer Science Colloquium*, Rohnert Park, CA, Nov 2012; *Hacker School*, New York, NY, Oct 2012.
- The Ph.D. Grind: Candid Discussions About Ph.D. Life. *UC Riverside – Computer Science Colloquium*, Riverside, CA, Oct 2012; *Google Tech Talk*, Mountain View, CA, August 2012.
- CDE: automatically creating reproducible experimental software packages. *Reproducible Research: Tools and Strategies for Scientific Computing* interdisciplinary meeting, Vancouver, Canada, July 2011; *NASA JPL*, Pasadena, CA, May 2011.
- CDE: Using System Call Interposition to Automatically Create Portable Software Packages. *Google Tech Talk*, Mountain View, CA, Feb 2011.
- The potentials and challenges of implementing automatic test generation using combined concrete and symbolic execution. *Fujitsu*, Sunnyvale, CA, Oct 2009.
- Automatic Creation of SQL Injection and Cross-site Scripting Attacks. *Samsung R&D Center*, San Jose, CA, May 2009.

SERVICE

Program Committee

- ICER 2017 review committee (Int'l Computing Education Research Conference)
- EDM 2017 (International Conference on Educational Data Mining)
- VISSOFT 2016 (IEEE Working Conference on Software Visualization)
- EDM 2016 (International Conference on Educational Data Mining)
- L@S 2016 (ACM Conference on Learning at Scale)
- L@S 2015 (ACM Conference on Learning at Scale)
- PLOOC 2015 (Workshop on Programming Languages Technology for MOOCs)
- CHESE 2015 (Int'l Code Hunt Workshop on Educational Software Engineering)
- SPLASH-E 2015 (Systems, Programming, Languages and Applications: Software for Humanity – Education Symposium)
- PLATEAU 2012 (Workshop on Evaluation and Usability of Programming Languages and Tools)
- TaPP 2012 (Workshop on the Theory and Practice of Provenance)

Paper Reviewer CHI (2014–2017), UIST (2013–2016), CSCW (2014–2017), TOCE (2017), JSME (2017), JVLC (2016), JVLSS (2016), IEEE Software (2016), TOCHI (2015), IUI (2015), MobileHCI (2015–2016), UbiComp (2015), JAIED (2015, 2017), TSE (2014–2015) PLDI (2013), EuroSys (2012), POPL (2011), ECOOP (2006, 2009)

Grant Reviewer Sloan Foundation (2012), NSF panel (2016)
(details omitted for confidentiality)

Artifact Eval. Committee	ESEC/FSE 2011 (Symposium on the Foundations of Software Engineering)

TEACHING

Instructor	• UCSD COGS120/CSE170: Human-Computer Interaction Design (Fall 2016) • UCSD COGS121: Human-Computer Interaction Programming Studio (Spring 2017) • UCSD COGS231: Grad Seminar on Human-Centered Programming (Spring 2017) • U. Rochester CSC 210: Principles of Web Application Development (Fall 2014, Fall 2015) • U. Rochester CSC 253: Dynamic Languages & Software Development (Fall 2014) • MIT 6.813 – User Interface Design & Implementation, co-taught with Rob Miller, Daniel Jackson, and David Karger (Spring 2014)
Teaching Assistant	• Stanford CS343 – Advanced Topics in Compilers (Spring 2012) • Stanford CS242 – Programming Languages (Autumn 2009) • Stanford CS243 – Advanced Compiling Techniques (Winter 2008) • MIT 6.170 – Laboratory in Software Engineering (Spring 2006)
Undergraduate Laboratory Assistant	• MIT 6.170 – Laboratory in Software Engineering (Fall 2004) • MIT 6.111 – Introductory Digital Systems Laboratory (Fall 2004) • MIT 6.001 – Structure and Interpretation of Computer Programs (Spring 2002)

CURRENT AND FORMER RESEARCH STUDENTS SUPERVISED

Ph.D.	• Logan Gittelson • Jaime Montoya • Xiong Zhang
Masters	• Davide Berdin (visitor from Uppsala University, Sweden) • Hyeonsu Kang • Kandarp Khandwala • Alok Mysore • Dan Scarafoni – first position: MIT Lincoln Laboratory • Jeremy Warner [C.22, C.24, C.30] – first position: Ph.D. student at UC Berkeley EECS • Jeffery White [C.24, C.26] • Renan Zanelatto [C.24, C.26]
Undergraduate	• Karina Banda • Lenny Brown • Charles Chen • Irene Chen – first position: Google • Jennifer (Kate) Godzicki • Mitchell Gordon [C.24, C.25] – 2015 CRA Outstanding Undergraduate Researcher Award winner, first position: Ph.D. student at Stanford Computer Science • Dan Hassin • Sara Lickers • Emy Lin – first position: Intel • Douglas Miller – first position: Jump Trading • Anvisha Pai – first position: Dropbox • Annie Zhang • Joyce Zhu [C.24] – 2015 CRA Outstanding Undergraduate Researcher Award honorable mention

Ph.D. Committee Member	Erin Brady, Anna Loparev, Phyo Thiha, Eric Seidel, Tricia Ngoon, Adam Rule