

# Adam Blank

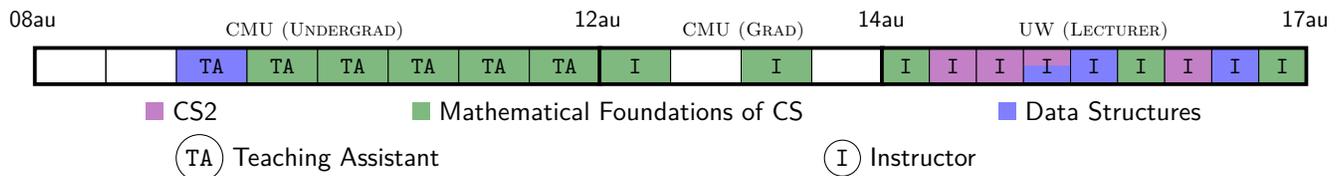
---

PERSONAL INFORMATION    E-mail: [adamblan@countablethoughts.com](mailto:adamblan@countablethoughts.com)  
Web: <http://www.countablethoughts.com>

EDUCATION    **Carnegie Mellon University** 2012 – 2014  
Master of Science in Computer Science  
Thesis: *Technological and Pedagogical Innovations for Teaching Introductory Discrete Mathematics to Computer Science Students*  
Advisors: *Randy Bryant, Klaus Sutner*

**Carnegie Mellon University** 2008 – 2012  
Bachelor of Science in Computer Science, Minor in Mathematics

EMPLOYMENT    **Lecturer** 2014 – Current  
Computer Science and Engineering, University of Washington, Seattle WA



TEACHING (UW)    **CSE 143: Computer Programming II**  
Instructor (*Spring 2015, Autumn 2015, Autumn 2016*)  
Co-instructor (*Winter 2015*)  
160 – 700 students  
This course is “CS2” in Java at UW. It introduces students to ADTs, and we implement simple data structures like Linked Lists and Binary Trees. The students are *very varied*. In Autumn 2015, I taught “CSE 143X” which combines CS1 and CS2 into a single 10 week course.

**CSE 390H: CSE 143 Honors Seminar**  
Instructor (*Autumn 2014, Winter 2015, Spring 2015, Autumn 2015*)  
50 students  
This is a seminar accompanying CSE 143. The only stated course goal is to get students interested in Computer Science. We discussed a wide variety of topics such as steganography, combinatorial games, esoteric programming languages, and cryptograms.

**CSE 311: Foundations for Computing I**  
Instructor (*Spring 2017\*, Spring 2016*)  
Co-instructor (*Autumn 2014*)  
100 – 140 students, all new majors  
UW CSE Students take this course in their first quarter in the major. CSE 311 combines many of the traditional introductory discrete math topics (e.g., logic, proofs, number theory, sets, relations, various types of induction) with some introductory computability material (e.g., regexps, CFGs, DFAs, NFAs, irregularity, uncomputability). I developed several online homework “autograders” for this course which I have helped other instructors integrate into their offerings.

### **CSE 332: Data Structures & Parallelism**

Instructor (*Autumn 2015, Winter 2016, Winter 2017\**)

120 – 280 students

UW CSE Students take this course in their second quarter in the major. When I first taught this course, the projects had not been changed for six years; so, I wrote new projects in which students implement all the “back-end” data structures that make a “real world” client work. This course is the first time students usually have projects that last several weeks; so, I introduced “checkpoints” in which students meet with someone on course staff in an open area to discuss aspects of the projects. To make things run smoothly, I built some tools which integrate with gitlab continuous integration to help monitor students for these checkpoints. Although I didn’t teach this course in Spring 2016 or Autumn 2016, both instructors adopted my projects. In Autumn 2016, the instructor also adopted my course policies and online exercises.

TEACHING  
(CMU)

### **15-151: Mathematical Foundations of Computer Science**

Course Designer and Instructor (*Fall 2012, Fall 2013*)

100 – 140 students, all freshmen

All computer science majors at CMU take two introductory discrete math courses: one in their first semester and another in their second. The course taken in the spring is “15-251” (see below), and it is traditionally a very difficult course. I created 15-151 for CS majors to make 15-251 less overwhelming. I used programming analogies and various techniques like group work, quizzes worth no points to check their understanding, and mechanisms/techniques to increase learning. In 2011 and 2012, I taught three lectures of 15-151 on Monday, Wednesday, and Friday. In 2011, my mentor, Klaus Sutner, gave half of the lectures.

### **21-127: Concepts of Mathematics**

Teaching Assistant (*Fall 2011*)

200 – 300 (50 in my section) students, primarily freshmen

This course is a generic introduction to discrete mathematics for various disciplines. I held recitation for 50 students twice a week, wrote my own handouts and quizzes, wrote several of the exams, graded for my entire recitation, and held 3-4 office hours every week.

### **15-131/98-172: Great Practical Ideas for Computer Scientists**

Course Designer and Instructor (*Fall 2011*)

Co-instructor (*Fall 2012*)

100 students, all freshmen

CMU recently changed their introductory sequence to include additional content on verification and parallelism. After this change, students needed additional background on UNIX, bash scripting, and C debugging; so, I created 98-172, a student-taught course to fill this need. I gave lecture once a week, and held four “lab-itations” (a cross between recitation and lab) each week as well. In Fall 2013, CMU decided to make 98-172 a course in the CS department, and I helped the effort to make the necessary changes.

### **15-251: Great Theoretical Ideas in Computer Science**

Head Teaching Assistant (*Spring 2012*)

Teaching Assistant (*Fall 2010, Spring 2011, Spring 2010*)

150 – 200 (40 in my section) students, primarily freshmen

This course is a very difficult introduction to discrete mathematics and computability theory. It covers a new topic every week, and the homework questions are traditionally time consuming. Every semester I TAed, I held a recitation for around 40 students, held at least 3-4 office hours every week, developed homework assignments and recitations, and graded. I also maintained the course submission and grading system and over many of the semesters, I developed a new one. In Spring 2012, I had several additional duties as the head teaching assistant. I acted as liaison between the other TAs and professors, and I held extra “conceptual” office hours every week in which I repeated the topics from lectures for students who were struggling. These office hours usually were attended by 20-30 students.

HONORS	ACM Teaching Award	2017
	Alan J. Perlis Undergraduate Student Teaching Award	2012
	Honorable Mention for CRA Outstanding Undergraduate Award	2012

EDTECH PROJECTS As a companion to my teaching, my research consists of educational technology which I often test with user studies and other formal evaluations. Many undergraduate students have contributed time and energy to several of these projects.

**Java Data Structure Visualizer** *2016 – Current*  
 I developed two Eclipse plug-ins: EZclipse and Viz. EZclipse makes the eclipse user interface more friendly to beginners. Viz provides a “logical visualization” of various features of Java programs during debugging. In particular, it helps students visualize data structures, stack traces, and fields.

**Induction Practice Interface** *2015 – Current*  
 Some students and I developed an interface (and checker) for simple induction problems. We are working on extending it and using it in an introductory discrete math course for practice and homework problems.

**Online HW Problems Framework** *2014 – Current*  
 I have written online interfaces for various problems (“construct a DFA”, “insert into this AVL tree”, etc.) in introductory discrete math and data structures courses to (1) encourage students to make multiple attempts on a problem, and (2) speed up grading. These interfaces give minimal (but useful) feedback, and the students are given a fixed number of attempts for each problem. I am working on expanding and improving as many of these interfaces as possible for various courses. They are being used by multiple instructors in both CSE 311 and CSE 332 at UW.

**A compiler for introductory discrete mathematics** *2012 – Current*  
 I designed a programming language called “Setty” and implemented a compiler. The standard library builds naturals, pairs, etc. from sets using the standard mathematical constructions. The goal of this language (which we used in 15-151) was to allow students to learn “the mathematical language” such as set comprehensions like  $\{x \in [5] \mid x > 2\}$  and quantified statements like  $\forall(x \in [5]). x < 5$  by evaluating them in a computational environment.

**Improving submission, annotation, review, and feedback of proofs** *2010 – Current*  
 I developed a course infrastructure system that handles LaTeX submissions, annotation, grading, and feedback of proofs. The system allows graders to first review and mark up the student submissions while re-using comments without worrying about point values, and then apply a rubric to their reviews. By providing an interface to easily reuse comments and update the rubric globally with a single click, we reduce the grading time while increasing the usefulness of feedback.

**Peer grading of proofs** *2010 – Current*  
 The system we developed for submission and feedback also allowed students to anonymously review each other’s submissions. I ran a study which showed that students who *reviewed* more proofs rather than *doing* more proofs wrote better proofs later.

SERVICE	<b>Co-designing and Leading Upper Division TA Training (UW CSE)</b>	<i>2016 – Current</i>
	<b>Member of UW Evidence-based Teaching and Learning Community</b>	<i>2015 – Current</i>
	<b>UW CSE ACM Student Chapter Advisor</b>	<i>2016 – Current</i>
	<b>UW CSE Scholarship Committee</b>	<i>2015 – Current</i>
	<b>CMU SCS CSD Teaching Assistant Advisory Committee</b>	<i>2012 – 2014</i>
	The TA Advisory Committee reviews and recommends policies for the TA program in the CS department. I served on a subcommittee which worked on the end-of-semester evaluation process for TAs.	
TEACHING MENTORING	<i>Michael Lee (2015–)</i> <i>Evan McCarty (2015–)</i> <i>Riley Porter (2015–2016)</i> . (first job: Lecturer at University of Washington)	
UGRAD RESEARCH SUPERVISION	Melissa Medsker-Galloway (2015–), Johan Michalove (2015–), Jefferson Van Wagenen (2015–), Logan Weber (2015–).	

## REFERENCES

### **Mark Stehlik**

*Teaching Professor*  
Department of Computer Science  
Carnegie Mellon University  
5000 Forbes Avenue  
Pittsburgh, PA 15213  
☎ (412) 268-6273  
✉ [mjs@cs.cmu.edu](mailto:mjs@cs.cmu.edu)

### **Stuart Reges**

*Principal Lecturer*  
Department of Computer Science & Engineering  
University of Washington  
AC101 Paul G. Allen Center, Box 352350  
185 Stevens Way  
Seattle, WA 98195-2350  
☎ (206) 685-9138  
✉ [reges@cs.washington.edu](mailto:reges@cs.washington.edu)

### **Ruth Anderson**

*Senior Lecturer*  
Department of Computer Science & Engineering  
University of Washington  
AC101 Paul G. Allen Center, Box 352350  
185 Stevens Way  
Seattle, WA 98195-2350  
☎ (206) 616-1742  
✉ [rea@cs.washington.edu](mailto:rea@cs.washington.edu)