Texas State University

Part-Time Faculty Excellence in Teaching Award

Nomination Form

Name **Molly A. O'Neil** Net ID **mo1162**

Department **Computer Science** College **Science & Engineering**

Current TXST teaching appointment FTE% **40**

Number of long semesters of TXST teaching at 50% or more FTE **0 (2 at 40%)**

Brief statement (100 – 150 words) of why the nominee is deserving of this award:

It is my great pleasure to nominate Mrs. Molly O'Neil for the Part-Time Faculty Excellence in Teaching Award. She has only taught a little over one semester, yet her very first teaching evaluations already far exceed our department's average (4.4 versus 3.74 at the 2000 level). This is an incredible achievement for a first-time teacher. The students describe her course as inspiring, well organized, challenging, and effective at promoting learning both in and out of class. Several students commented on her genuine empathy for her students. Clearly, Molly deeply cares about teaching and is committed to continued teaching improvement. For example, she attended a Computer Science education conference last year and has already incorporated ideas learned there into her classroom through new assignment design and the implementation of classroom practices aimed at fostering a more inclusive environment for underrepresented students in STEM. In addition to teaching, she voluntarily participates in an inter-disciplinary project aimed at increasing STEM retention at Texas State through curricular improvements and has implemented ideas from that project into her class curriculum. In my opinion, Molly is a great candidate for the Part-Time Faculty Excellence in Teaching Award. I recommend her very highly.

Sincerely,
Martin Burtscher, Ph.D.
Professor
Department of Computer Science
Texas State University

TEXAS STATE VITA

I. ACADEMIC/PROFESSIONAL BACKGROUND

A. Name: Molly A. O'Neil **Title:** Lecturer

B. Educational Background

Degree	Year	University	Major	Thesis/Dissertation
M.S.	2015	Texas State University San Marcos, TX	Computer Science	*Characterizing the Performance Bottlenecks of Irregular GPU Kernels* Advisor: Martin Burtscher
B.S.	2005	Carnegie Mellon University Pittsburgh, PA	Electrical & Computer Engineering, Engineering & Public Policy	

C. University Experience

Position	University	Dates
Lecturer	Texas State University	Sep. 2015 – present
Graduate Research Assistant	Texas State University	Sep. 2010 – May 2015
Head Teaching Assistant	Carnegie Mellon University	Aug. 2004 – Dec. 2004
Teaching Assistant	Carnegie Mellon University	Jan. 2004 – Aug. 2004

D. Relevant Professional Experience

Position	Entity	Dates
Senior Design Engineer	ARM, Inc., Austin, TX	Jan. 2009 – Mar. 2010
Design Engineer III	ARM, Inc., Austin, TX	Jul. 2006 – Jan. 2009
Design Engineer II	ARM, Inc., Austin, TX	Jun. 2005 – Jul. 2006

II. TEACHING

B. Courses Taught

Department of Computer Science, Texas State University

- CS 2308: Foundations of Computer Science II
  - Spring 2016 (44 students)
  - Fall 2015 (2 sections, 119 students)

- CS 3339: Computer Architecture
  - Spring 2016 (40 students)

III. SCHOLARLY/CREATIVE

A. Works in Print

3. Conference Proceedings

- Molly A. O'Neil and Martin Burtscher. "Microarchitectural Performance Characterization of Irregular GPU Kernels." Proc. of the IEEE International Symposium on Workload Characterization. Raleigh, NC. October 2014. (28% acceptance rate)

- Molly A. O'Neil, Dan Tamir, and Martin Burtscher. "A Parallel GPU Version of the Traveling Salesman Problem." Proc. of the 2011 International Conference on Parallel and Distributed Processing Techniques and Applications. Las Vegas, NV. July 2011.

4. Workshop Proceedings:

- Molly A. O'Neil and Martin Burtscher. "Rethinking the Parallelization of Random-Restart Hill Climbing: A Case Study in Optimizing a 2-Opt TSP Solver for GPU Execution." Proc. of the Eighth Workshop on General Purpose Processing on Graphics Processing Units. San Francisco, CA. February 2015. (65% acceptance rate)

- Molly A. O'Neil and Martin Burtscher. "Floating-Point Data Compression at 75 Gb/s on a GPU." Proc. of the Fourth Workshop on General Purpose Processing on Graphics Processing Units. Newport Beach, CA. March 2011. (37% acceptance rate)

D. Fellowships, Awards, Honors

- Graduate Research Excellence Award, Computer Science Department, Texas State University, April 2015
- Graduate Academic Excellence Award, Computer Science Department, Texas State University, April 2015
- Graduate Academic Excellence Award, Computer Science Department, Texas State University, April 2014
- Graduate Academic Excellence Award, Computer Science Department, Texas State University, April 2013
- Outstanding Graduate Student Award, College of Science & Engineering, Texas State University, April 2012
- National Science Foundation Graduate Research Fellowship, 2011 – 2015
- Graduate Research Excellence Award, Computer Science Department, Texas State University, April 2011
- Celebrity Classic Scholarship, The Graduate College, Texas State University, July 2011 [declined]
- Southwest Research Institute Women in Science and Engineering Scholarship, April 2011 [declined]

- University Honors, Carnegie Mellon University, May 2005
- International Engineering Consortium William L. Everitt Student Award of Excellence, Carnegie Mellon University, May 2005
- Carnegie Institute of Technology Dean's List, Carnegie Mellon University (5 of 8 semesters, 2001 – 2005)
- Andrew Carnegie Scholarship, Carnegie Mellon University, 2001 – 2005

IV. SERVICE

B. College/Departmental

- Participant, STEM Rising Stars Brown Bag Lunches on STEM Curriculum, 2015 – 2016

C. Community

- Alumni Interviewer, Carnegie Mellon Admissions Council, 2009 – present

D. Professional

- Reviewer, IEEE International Conference on Cluster Computing (Cluster), 2015
- Reviewer, ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), 2014
- Reviewer, ACM SIGPLAN Symposium on Principles and Practices of Parallel Programming (PPoPP), 2012 – 2015
- Reviewer, Workshop on General Purpose Processing on Graphics Processing Units (GPGPU), 2012 – 2013

CS 3339: Computer Architecture
Spring 2016

Section 251

General Information

Instructor	Ms. Molly O'Neil Comal 207H (512) 245-6670 `moneil@txstate.edu` (Email is the best way to contact me. Please allow several hours for a response and do not count on responses evenings or weekends – though you will often get them.)
Course Webpage	`http://www.cs.txstate.edu/~mo1162/cs3339`
Office Hours	M: 3:30 PM – 5:30 PM W: 10:15 AM – 12:15 PM 3:30 PM – 4:30 PM Other times by appointment
Lecture	M/W 12:30–1:50 PM Derrick Hall #235
Textbook	David Patterson and John Hennessy, Computer Organization and Design: The Hardware/Software Interface, 5th Edition, ISBN-13: 9780124077263
Readings	Chapters 1 – 6 [required]
Prerequisites	C or higher in CS 2315 (Computer Ethics) or EE 2400 C or higher in CS 2318 (Assembly Language) or EE 3420 C or higher in CS 2420 (Digital Logic) or EE 2420

Course Description
Use of fundamental hardware components. Topics include ALUs, single and multiple cycle datapath and control, RISC vs. CISC, pipelining, caches, I/O, virtual memory, and related performance issues.

Learning Objectives
1. The student will understand how computers work at many different levels of abstraction, primarily focused on datapath and control of RISC and CISC instruction sets with a look at ALUs, caches, virtual memory, and I/O
2. The student will learn about the terminology used to sell computers (e.g., clock rate, MIPS, SPEC ratings) and measure performance (e.g., CPI, instruction count) and be able to detect marketing hype and understand what they are buying

3. Abstraction is a guiding principle used to understand this material, and students should come away from this class with the ability to work comfortably at many different levels of abstraction

Attendance, Grading, Exams, & Assignments

Attendance

Class participation is required and will comprise 5% of the final course average. Regular (though not perfect) attendance is required for full participation credit. The instructor reserves the right to call on any student during any lecture. After two absences, additional absences will incur a 1% reduction in final course grade (up to 5% total).

Exams

There will be one midterm exam and a comprehensive final exam. You are allowed only the use of writing utensils at exams (i.e., no textbooks, notes, calculators, cellphones, etc.).

Assignments

There will be six written homework assignments and six coding projects, each worth 5% of the final course grade. The lowest written homework grade will be dropped; coding project grades are not dropped. You are strongly encouraged to complete all the homework assignments to prepare for exams.

Grading

Participation:	5%	
Homework	25%	(lowest dropped)
Projects	30%	
Midterm Exam	15%	Wed. Mar. 9 [*tentative*]
Final Exam (comprehensive)	25%	Mon. May 11, 11:00 AM - 1:30 PM

Final grades will be assigned on a 90-80-70-60 scale; however, I reserve the right to alter this scale to the benefit of the students.

Classroom Policies

Makeup Policy

Missed assignments cannot be made up. Exams may be made up in exceptional circumstances (e.g., a documented medical emergency or religious holy day), with instructor's approval and prior authorization where feasible.

Late Policy

Late submissions will not be accepted and extensions will not be granted except in the case of emergencies (I will ask for proof). In particular, projects/homework/exams in other classes with similar due dates will not be considered valid reason for an extension. If you cannot complete an assignment, you are encouraged to submit a compiling program implementing a subset of the assignment requirements. Please explain in header comments which features are not implemented.

Submission

Programming projects may be submitted more than once and only the most recent submission made before the deadline will be graded. I will

	not grade non-compiling code: submissions that do not compile will automatically receive a zero. I strongly encourage you to compile your code often during development. Caution: just because a program compiles on your system does not guarantee that it will compile (or execute the same) on the grading system. It is your responsibility to ensure that your code compiles and works on the grading system before submitting.
Re-grading	All requests to re-grade work must be submitted to the instructor in writing, explaining *what* should be re-graded and *why*. The re-grade request must be received within one week of the time the grade for the assignment or exam was first returned.
Communication	Communication related to this class will be sent to your txstate.edu email account. Check this account regularly. I do not guarantee that I will see emails sent from off-campus addresses.
TRACS/Web	The TRACS website will be used for project submission (Assignments tool), grades (Gradebook2 tool), and all assignment handouts and lecture slides (Resources tool). An up-to-date class schedule can be found on the course website.
Withdrawal/Drop	You must follow the withdrawal and drop policy of the University and the College of Science. You are responsible for making sure that the drop process is complete. For more information, see: `http://www.registrar.txstate.edu/registration/drop-a-class.html`. I encourage you to come talk to me before deciding to drop the course. **Last day to drop:** March 29, 2016
Accommodation	Any student with needs requiring special accommodations should inform the instructor during the first two weeks of classes. The student should also contact the Office of Disability Services in the LBJ Student Center.

Academic Honesty

You are expected to adhere to the University's Academic Honor Code (`http://www.txstate.edu/effective/upps/upps-07-10-01-att1.html`).

All assignments, exams, and quizzes must be done individually. Turning in an exam or assignment that is not entirely your own work is cheating and will not be tolerated.

Group discussion about course content is NOT cheating, and is in fact strongly encouraged! You are welcome to study together and to discuss information and concepts covered in class, as well as to offer (or receive) help with debugging or with understanding course concepts to (or from) other students. However, this cooperation should never involve students possessing a copy of work done by another student, including solutions from previous semesters, other course sections, the Internet, or any other sources.

Talking or discussion during exams/quizzes is not permitted, nor may you compare notes or copy from others. Any collaboration during exams will result in a 0 grade on the exam, potentially a lowered or failing course grade, and will be reported to the Texas State Honor Code Council (`http://www.txstate.edu/honorcodecouncil/`).

Turning in an assignment any part of which is derived from the Internet, another student's code, or any other non-approved source will result in a 0 grade on the assignment and will be reported to the Texas State Honor Code Council. Should one student copy from another, both the student who copied work and the student who gave material to be copied will receive 0s and be reported to the Honor Code Council. You should never grant anyone access to your files or email your program to anyone (other than the instructor)!

Schedule

These topics and dates are tentative and the instructor reserves the right to alter the schedule as the semester progresses. Reading assignments are to be completed before attending the stated lecture.

Week #		Date	Topic	Reading	Assignments
Week 1	M	Jan. 18	***No Class -- MLK Day***		
	W	Jan. 20	Introduction / Architectural Trends		HW1 out, Project 1 out
Week 2	M	Jan. 25	Evaluating Performance		
	W	Jan. 27	Instruction Set Architecture (ISA)		HW 1 DUE
Week 3	M	Feb. 1	Instruction Set Architecture (ISA)		
	W	Feb. 3	ISA Comparison / ALUs		Project 1 DUE
Week 4	M	Feb. 8	ALUs		
	W	Feb. 10	MIPS Datapath		HW 2 DUE
Week 5	M	Feb. 15	Pipelining		
	W	Feb. 17	Pipelining Hazards		Project 2 DUE
Week 6	M	Feb. 22	Branch Prediction & Exceptions		
	W	Feb. 24	ILP: Static Multiple Issue		HW 3 DUE
Week 7	M	Feb. 29	ILP: Dynamic Multiple Issue		
	W	Mar. 2	ILP: Dynamic Multiple Issue		Project 3 DUE
Week 8	M	Mar. 7	Midterm Review		
	W	Mar. 9	**Midterm Exam**		
Spring Break	M	Mar. 14	***No Class -- Spring Break***		
	W	Mar. 16	***No Class -- Spring Break***		
Week 9	M	Mar. 21	Midterm Handback & Review		
	W	Mar. 23	Memory Hierarchy		HW 4 DUE
Week 10	M	Mar. 28	Caches		
	W	Mar. 30	Caches		Project 4 DUE
Week 11	M	Apr. 4	Virtual Memory		
	W	Apr. 6	Parallel Architectures		HW 5 DUE
Week 12	M	Apr. 11	SMPs		
	W	Apr. 13	Cache Coherence		Project 5 DUE
Week 13	M	Apr. 18	Multithreading / SIMD & Vector		
	W	Apr. 20	GPUs		HW 6 DUE
Week 14	M	Apr. 25	Main Memory & I/O		
	W	Apr. 27	Clusters		Project 6 DUE
Week 15	M	May 2	Final Exam Review		
	W	May 4	***No Class -- Finals Week Begins***		
Week 16	W	May 11	**11:00 AM - 1:30 PM: Final Exam**		

Instructor: Molly A. O’Neil
CS 2308: Foundations of Computer Science II
(Sections 002 & 004, Fall 2015)

Anonymous Student Evaluation Comments

[Unedited]

1.)
This has been the best class I have taken at this University. I have been challenged by the coding assignments, but I have learned so much by them, I feel, that Dr. O’Neill has provided me the opportunity to grow as a programmer because I have found a new appreciation for computer science, I have always felt like she genuinely cares about her students, and that is something you don’t find in many people. I am encouraged to make mistakes and learn so so much from them, the teaching philosophy behind Dr. O’Neill’s class, encourages me to excel in my coding assignments, and if I make mistakes I’m encouraged to take the time to fix them. For me, this is my optimal learning style, learning through mistakes. I love this course!!!

2.)
Undoubtedly the best professor I've ever had. Class was conducted in such as way that one was encouraged to learn in and out of class. Grading was slower on some things, but all in all grading on such assignments is expected to take a while. Professor was quick to respond to emails and was very helpful when one approached her with questions. I learned so much more in this course solely because of the instructor.

Student Email (Oct. 5, 2015)

3.)
Hello Mrs. O’Neil,
You seem sometimes to worry if you are teaching the class well, so I wanted to offer my opinion to you. For the record, I already have a degree in literature -- summa cum laude. I am in the CS certificate program here. In my time getting the literature degree, I had one professor inspire me who I thought was truly excellent. I would consider you her counterpart in this department. I have never had a teacher who explains things better, as well organized, and who seems to want so badly for her students to succeed. Try not to lose that as you continue — I know there will be students who don’t seem to make it worth it, but there are people like me in the class to that think you do just a remarkable job. From the professors I have had in the CS department at this school and elsewhere, you are so far above it’s crazy. I was so relieved immediately to see how you were teaching this class.

So, thank you,
Chris Hudson

Part-Time Faculty Awards for Excellence in Teaching Application
Molly A. O'Neil

Teaching Practice Prompt #1:
What are your personal strengths as a teacher?

The task of educating young adults cannot stop at the classroom door. This belief motivates several of my strengths as a teacher, namely my efforts to mentor students in and out of the classroom and to concretely connect the class material to their futures. I endeavor to connect every topic we cover in class to its real-world applications, backed up by stories from my own industry experience. Students have commented in evaluations that "she offers real world experience and provides those stories to give us an even better understanding of the material" and that "[she] always has very strong endings (final words of class)." This is the result of my effort to structure each lesson around the big-picture importance of the material. Moreover, I set aside class time to discuss career options within the field of CS, how to ace technical interviews, and other real-world topics; this approach motivates students to succeed.

Setting a tone of mentorship inside the classroom encourages students to approach me for advice outside of class, too. Student evaluations mention my "great empathy" and that students feel "[she] genuinely cares about her students." For example, last semester I had a student underperform her capabilities on the first exam due to what I suspected was severe exam anxiety. I invited the student to drop by office hours, where we discussed anxiety, my own personal experience combating self-doubt, resources for help, and strategies for future exams. The student scored second-highest in her section on the next exam, and she later told me that my intervention had been critical in helping her succeed. I believe this whole-picture approach to educating – both mentoring the whole student and putting each new topic in its broader context – is my personal strength as a teacher.

Part-Time Faculty Awards for Excellence in Teaching Application
Molly A. O'Neil

<u>Teaching Practice Prompt #2:</u>

How has your teaching changed since you began teaching and what have you done to improve it?

Effective teaching requires frequent measuring and analysis of student experiences, as well as reflection. Last semester, I used the final attendance pop quiz as an optional survey of student attitudes regarding the design of the course and the assignments, and this semester I have made modifications to several assignments and re-ordered some course content based on feedback from that survey. I have also implemented changes to the way I track attendance this semester, based on reflection on the experiences of underrepresented students in my courses last semester and recent research on supporting women in STEM classrooms (see Prompt #3).

I volunteered to be one of my department's participants in the Texas State STEM Rising Star project's Brown Bag Lunches aimed at inter-disciplinary curricular improvements to support retention rates in STEM. One of the facets of the STEM Rising Stars project aims at increasing CS retention by establishing, early in the curriculum, better student understanding of what computer science is and the wide range of global and societal problems to which it can be applied. Based on conversations with the participants in that project, I voluntarily designed a new module that can be incorporated into introductory CS classes to further this goal, and this semester I incorporated that module into my CS 2308 curriculum.

Part-Time Faculty Awards for Excellence in Teaching Application

Molly A. O'Neil

Teaching Practice Prompt #3:

Give an example of a teaching challenge you have encountered and explain how you've dealt with it.

Last semester, I was frustrated to observe a phenomenon in my classes that I grew concerned perpetuated negative stereotypes about women and other underrepresented students in CS: the tendency for all prompts to the class to become a discussion involving only a few assertive students (who, in a CS classroom, are overwhelmingly likely to be male). Nothing I did seemed to get my bright but passive students to speak up.

Last spring, while still a graduate student, I attended the ACM Special Interest Group on Computer Science Education (SIGCSE) conference in search of ideas to incorporate into my future classroom. I remembered a SIGCSE panel that mentioned randomized calling on students in STEM classrooms, to mitigate the effect of those few assertive students whose hands are always first in the air. I tracked down and read several papers on this topic, and, this semester, I implemented a similar idea in my CS 3339 course. Rather than my old pop quizzes to enforce attendance, I now award participation points based on students being present when randomly called on. I shuffle a card deck of all student names at the beginning of every class. Throughout lecture, I flip over the top card and call on that student: sometimes to answer a question, sometimes to ask me a question. I have worked hard to create a classroom culture where it is acceptable to say "pass," and only absence results in a point deduction. However, this approach has levelled the playing field for all students, such that everyone gets a chance to demonstrate their knowledge (and, just as importantly, that other students often *don't* know the answer). At mid-semester, this has already created a classroom environment dominated by group collaboration involving a much broader segment of the class than I observed last semester.

Part-Time Faculty Awards for Excellence in Teaching Application
Molly A. O'Neil

Teaching Practice Prompt #4:

Please give examples of innovative assignments and course design components that promote active learning and/or engagement.

In any computer science course, programming assignments frame the majority of student learning. Assignments must be challenging enough to drive learning, large-scale enough to be rewarding while remaining realistic for the majority of students to complete, and fun enough to be motivating. I designed two new CS 2308 assignments with these objectives in mind, both of which were well-reviewed by students in surveys I conducted at semester end. The first, an image processing assignment in the first week of class, was designed to be easy and confidence-building but to yield a very fun, graphical result. The other was a capstone on the skills learned over the entire semester, applied to one of the canonical problems in computer science. For many of my students, it was the first time they were given a chance to apply their skills to a problem that felt real rather than contrived.

Similarly, in CS 3339, I substantially expanded an existing project, resulting in a complex assignment built over six milestones from both student programming and significant amounts of my own program code that students must read, understand, and extend. This makes the project's scope feasible while still enabling students to create a real-world piece of software by the end of the semester. It also gives students practice at inheriting a large codebase written by someone else, a critical skill required in industry but at which college classes rarely provide experience.

In addition, unlike many CS faculty, I strongly encourage students to collaborate on projects. While each student must submit his/her own assignment, I recommend students work together to plan, problem-solve, program, and debug. I believe the group discussions and student explanations that result from coding collaboration drive much better learning outcomes for everyone. I have also implemented classroom policies aimed at facilitating group discussion (see Prompt #3).