Teaching Statement
Gennady Pekhimenko
gpekhime@cs.cmu.edu

Since my high school years, I have been involved in teaching and mentoring students, and I have found my experience very rewarding. I view teaching and working with students as an important complementary process to performing cutting-edge research. The opportunity to teach and work with the best students is one of the major reasons I would like to join academia.

Teaching Experience
As a PhD student at Carnegie Mellon University, I had the pleasure to serve as a teaching assistant twice. In Fall 2011, I served as the teaching assistant for the Introduction to Computer Systems (15-213) class, a core undergraduate class in computer systems for both CS and ECE departments. In this class, I was responsible for creating new assignments/homeworks, holding weekly recitations, and grading exams and homeworks. My favorite part was the recitation session, where I was able to answer students questions and had an opportunity to engage the audience with many real life examples from my full-time industry experience. I greatly enjoyed this experience, and was happy to see my students getting more and more interested in the area of computer systems as we progressed in this class. In Spring 2012, I served as the teaching assistant for Optimizing Compilers (15-745) class, the most advanced graduate-level class at CMU related to the topics of compilers and optimization. My responsibility in this class was to cover all the homeworks, support the labs, supervise several research projects, and to answer student questions through Piazza (our forum system for student questions). I also gave lectures in this course on several occasions.

Apart from being a teaching assistant at CMU, I also gave several guest lectures in computer architecture courses (both undergraduate and graduate) on the topics related to my research. It was amazing to see how quickly even undergraduate students understood the key insights behind my work when state-of-the-art ideas and mechanisms are presented in an effective manner.

As a Masters student at the University of Toronto in 2006–2008, I served four times as a teaching assistant: twice for the Operating Systems class, once for Software Engineering, and once for Computer Programming (all undergraduate classes). I enjoyed teaching these classes, because not only did I help my students learn more about the subject, but I also learned a lot myself, both about the right methodology of teaching these classes and how to engage students from many different backgrounds.

Teaching Philosophy. During my academic life I was fortunate to work closely with many amazing teachers and educators in U.S., Canada and Russia. Not only did I learn from them the specific subjects they were teaching in their classes, I also learned about different ways of explaining complex ideas, techniques to engage students, and how to generate excitement for the subject being taught. This experience shaped my own teaching philosophy, which consists of several major elements.

First, I am a strong proponent of active classes with many live discussions and conversations. It is very important to keep students excited about the topic, and my way to achieve this is by engaging students in active discussions, making them comfortable with sharing their thoughts and questions openly, and forcing them to look beyond simple whats taught in lectures.

Second, I like to add an element of friendly competition in both my lectures and homeworks. I think students are willing to invest more time in learning about the problem, and looking beyond the solutions proposed in class, when they have to compete with their peers, or even against TAs and instructors. For example, in Optimizing Compilers class it is possible to compete for the best average runtime on a set of applications. This automatically forces students to not only implement the optimizations discussed in class, but also make their own optimizations, based on the performance analysis of the target applications.

Mentoring Experience
I have been actively involved in mentoring students, both for projects directly related to my research and beyond. Overall, as of now, I have mentored (or am still mentoring) thirteen students of all levels: (i) five undergraduate students, (ii) four masters students, and (iii) four junior PhDs from both CS and ECE departments. Each mentorship experience was different, ranging from very close collaboration on the research topics that are a part of my PhD thesis to cases where the student (especially a young PhD student) was driving his/her own research topic, and I primarily helped in shaping the research direction.

This mentoring not only helped me progress faster in my research and learn many useful skills on how to manage and work with students of different seniority and backgrounds, but also helped students learn what research is about, and in many cases, even publish paper in top-tier conferences. For example, with undergraduate students Tyler Huberty and Rui Cai, we worked on compression-aware replacement policies for on-chip caches. This project started with the idea I suggested for the project in the advanced computer architecture class at CMU, and with tremendous help from both of these students, it became a publication at a top-tier conference (HPCA’15). This research experience turned to be very useful for the future career paths of both these
students. My CV has more details on each mentoring experience and its outcome.

Mentoring Philosophy. From my experience as a mentor, I learned several important things. First, every student is unique, and hence he or she requires a special attitude – there is no “one size fits all” approach that works for everyone. For example, some students are much more effective if you give them very precise directions both in terms of ideas, and evaluation methodology. On the other hand, there are students who are much more comfortable when you give them more freedom on what to explore, and how to go about it, as they are excited to discover things on their own.

Second, it is important to meet frequently, especially at the beginning of the project. For all the successful projects where I mentored students, regular exchange of ideas, thoughts, questions, and preliminary results were key factors in the success of the project.

Third, I see it as my job as a mentor to set proper goals for the project and scope it well. In my experience, even very smart and talented junior students have limited idea on what research is like. And, unfortunately, many of the students might not even realize that despite all difficulties, they are capable of doing research that is worth being published. Hence it is important to encourage students to target ambitious goals in their research projects, but also show them how to scope what can realistically be done to be successful in the project.

Course Coverage

In my opinion, a deep understanding of the fundamental principles that form the basis of modern computing systems is critical for a good education in both computer science and computer engineering. Computer architecture is an important layer between software and hardware that has to be efficient in order for the whole system to operate efficiently. In my undergraduate classes, I would like to emphasize its importance, and show the key principles and ideas that computer architecture area has to offer.

I am excited to teach a variety of classes in the systems spectrum: computer architecture, compilers, operating systems, and software engineering. I would also be happy to teach any introductory class in computer science, as I was always happy to work with junior students and invoke enthusiasm and interest for the general area of modern computing. I am also interested in organizing new classes for both graduate and undergraduate students that would address interesting interdisciplinary problems, e.g., software/hardware interfaces, data center designs, interaction between programming languages, OS, and hardware.