Teaching Statement

Yoongu Kim
yoongukim@cmu.edu

I derive a great deal of personal satisfaction from giving clear explanations. I strive to achieve that special expression of wonderment on the listener’s face, which happens only when a new idea has been installed satisfyingly into their mind. I find this to be one of the most rewarding experiences. My approach to teaching starts by decomposing a complex notion into the smallest number of non-trivial components. I then employ personal anecdotes, real-world analogies, and concrete artifacts to engage the audience. I am a strong proponent of minimum working examples and thought experiments as powerful tools for driving the point finally home. In this teaching statement, I present a brief summary of my previous experience in teaching and mentorship, followed by my future course offerings.

Teaching Experience

As a teaching assistant for the undergraduate-level computer architecture course at CMU, I interacted very closely with the students throughout the entire semester. Every week, I spent three hours in the computer laboratory to help students make progress on a MIPS processor that they were modeling as part of a semester-long project. This required them to write a C-based simulator, as well as to implement the processor in Verilog. A lesson that I quickly learned is that any ambiguity in the project’s specification is a major source of confusion for the students, which detracts from their overall learning experience. So, I took it upon myself to overhaul most of the documentation, tutorials, and scripts that are provided to the students, and optimized them for the utmost clarity. This had the immediate effect of vastly increasing the quality of questions that were asked by students. Additionally, I gave two guest lectures in class, where I taught virtual memory by gradually introducing its key aspects based on when they were first adopted by Intel processors. Such a historical approach gave students the proper context to understand why specific design decisions were made in the past as they were. After the lectures, many students expressed appreciation at the attention to detail that I had put into the lecture slides. Lastly, I was heavily involved in crafting homework/exam questions and grading them.

I also served as a teaching assistant for a graduate-level computer architecture course, which required the students to complete an original work of research. I met regularly with the students to brainstorm on open-ended research questions, and to help them set up the necessary research infrastructure. I am proud to say that one of the projects eventually went on to become a well-received publication at ISCA 2012.

Previously, throughout my undergraduate years in Korea, I accumulated hundreds of hours in tutoring high school students on mathematics, physics, and chemistry. It was immensely gratifying to see the students graduate and advance to college after the many years that I spent instructing them.

Research Mentorship

As the first student of my academic advisor, I have always been the most senior student in our research group. Over the years, I have actively helped the younger Ph.D. students on many aspects of research: topic selection, execution, and writing/presentation. Furthermore, I have also mentored six B.S. and M.S. students, who were instrumental in my own research. Many of my co-authored publications are a result of this mentorship and collaboration.
Course Offerings

I would be happy to teach computer architecture courses at both the undergraduate and graduate levels. I can also teach introductory digital logic design, and operating systems. In addition, I intend to design a graduate level curriculum that is focused on the memory system, which provides a comprehensive understanding of how data is managed across the entire memory hierarchy between the cache and storage. Using my industry contacts, I plan to procure prototypes of emerging memory technologies (e.g., 3D die-stacked memory, and non-volatile memory) and offer them to students as a research substrate for devising new ways to accelerate file-systems, key-value stores, databases, data-mining, graphics processing, etc.