Over the past years, my experiences as a student, lecturer, and mentor have shaped my own approach to teaching and learning. In brief, my teaching style emphasizes interactive synthesis of fundamental concepts in the classroom and hands-on projects outside of it. Understanding how classroom materials are used in practice to solve real-world problems, students will find it interesting to learn new ideas and skills, and motivate themselves for study.

Teaching Experience and Philosophy. During my undergraduate years, I had an opportunity to take an intensive Windows developer course for a year. The purpose of the course was training a team of fifty students to have competitive C and C++ programming skills and understand the internal mechanisms of the Windows operating system. As a project team, we taught and learned from each other interactively reviewing the course materials and working on many pilot projects together. From this experience, I learned that competition like settings do not make students perform very well. In contrast, I realized that students perform much better through active and close collaborations as a team sharing knowledge and giving feedbacks to each other. Such knowledge sharing process forms a great collaborative synergy for study: students with more knowledge strengthen their understanding by teaching other students while less experienced students gain new knowledge. In addition, I learned that by turning gained knowledge into code, students understand the real-world impact of what they learn and this inspires them to self-motivate to participate in the class. A pilot project with specific goals and tasks is a great way for students to experience this. While solving small problems step by step, students feel interested and gain confidence about their understanding of course materials and skills. With these valuable experiences, which I wish to share with future students, I was selected to receive the Learning Excellence Award from the course.

As a Ph.D. student at Purdue, I had privileges to give a guest lecture to a software security class and work as a private tutor of an undergraduate student. From these experiences, I found that students appreciate my teaching style and philosophy showing their interests about the course materials, actively asking questions, and having student-to-student discussions voluntarily.

Courses and Teaching Plans. I am interested and have confidence in teaching operating systems, systems security, information security, and software engineering at the undergraduate and graduate levels. Having experienced many different undergraduate and graduate level operating systems classes, I believe that one of the most effective ways to learn computer systems is through carefully-designed programming assignments under a principled guidance. I am going to provide a base infrastructure for the assignments with clean and simple interfaces that all students can easily understand. Each assignment will have a number of small individual problems on which students can follow with reasonable efforts. By solving each of the small problems one by one, students will be able to build a full-stack system gradually implementing core mechanisms and understand how the mechanisms interwork to provide key system capabilities. Similarly, students in security-related classes will gain hands-on experience implementing various attack and defense mechanisms, to explore offensive and defensive techniques used in the wild. I am also interested in designing more specialized courses for graduate students who have a strong interest in systems security and program analysis. Examples of such courses include “Virtualization security,” to help students understand virtualization systems and related offensive and defensive techniques; “Cyber-physical systems security,” to have students learn about newly emerging attacks against cyber-physical systems and the current state-of-the-art defense mechanisms; and “Program analysis for security,” to cover current and future topics in cyber security in which dynamic and static program analyses could play an important role.

Mentoring. I have been fortunate to mentor a talented senior undergraduate student from Purdue, who worked on a project to improve the security of cyber-physical systems with me. My mentoring was mostly about helping him to approach a problem as a researcher, assisting to resolve technical problems, and providing advice on his future research career. From the experience, I realized that it is
important for a graduate student to understand the big picture of research, in order to have strong self-motivation. A graduate student should first understand the entire research process in advance to conducting any meaningful research, from problem searching to research evaluation. I also learned that one of the main roles of an advisor is to have the student lead his/her own work and research. As an academic advisor, I am going to show respects to students’ opinions and encourage them to work on their own to become an independent researcher.