My first goal in every classroom will be to engage students in an ongoing dialogue with me, as well as other students. Making students comfortable enough to enter into dialogue may be a challenging task for someone in a position of authority in the classroom, yet a very important one. This dialogue will permit me to be aware of every student’s grasp of the material and learn from students a more effective way to present the subject.

In my past experiences as a hockey coach, as well as a Taekwon-do instructor, I found humor to be an effective way to break the ice and help establish communication with my audience. I can still recall my first experience as a Taekwon-do instructor near Chula Vista, California, just a few miles from the US-Mexican border. The group of kids, who were predominantly of Mexican descent, looked in fear at the new black belt in the school who grew up in a place very far away. After a silly joke and an awkward misstep, the discomfort with me was gone; they were ready to learn.

I believe that once students are at ease with me and their peers, many kinds of communication and collaboration will benefit the learning experience. I will try to stimulate discussions and group assignments in class, as well as collaborative projects outside it. Learning from others in class (and, at times, from their mistakes) will reinforce the concepts covered in the classroom.

I feel that learning should be continuously reinforced (rather than only occasionally tested in exams). I believe in supplementing lectures with frequent small assignments.

Additionally, substantial term-long class projects can lay the foundation for working in teams; such projects can teach students to plan, design, and execute their goals, as well as collaborate with others. These skills are essential for undergraduates preparing for future software-engineering careers and graduates preparing for research.

Tell me and I forget. Show me and I remember. Involve me and I understand. (Chinese proverb.)

One method of teaching that I experienced while an undergraduate at the University of Texas stands out in my mind. It is the Moore method, named for the late Professor of Mathematics, Robert Lee Moore. The method places an emphasis on discovery: the instructor helps students discover the answer by guiding them to it rather than lecturing about a known solution. In Mathematics, this can be done by dividing the proof of an old theorem into smaller pieces that the students could tackle on their own. I want to experiment with the application of the Moore method in my teaching. For instance, I can ask students to present pseudo-code for a data-structure algorithm or an optimization before covering it in class. For graduate students, a simple application of the method is to ask them to read a paper and send a thorough review before discussing the paper in class. Finding ways to employ this method to solve complex problems in class will be an exciting challenge.

At the cost of spending extra time on a complex topic, I like to present several different explanations and routes to the answer, when possible. A student who has not grasped the material may simply need a different way to view the problem or the solution. In my experience as a private tutor of Computer Science, Mathematics, and French, I met many students who fell behind in class. A fresh approach to the material made the difference for those I helped. The abilities and preparation of students at every level differ. Alternative approaches may allow a diverse group of individuals to be successful without hampering the progress of any.

I will support the students when they struggle, yet always demand the best from them. During my two years of coaching a hockey team of adult women with vastly different abilities and backgrounds, I found it immensely satisfying to see a weaker skater make a breakthrough because of the extra effort she put forth for the coach who refused to let her fall behind. Moments like these make the life of a teacher a happy one.

With enrollment in Computer Science classes dropping in many universities, I want to do my part in making the subject exciting and accessible.

I have the depth and breadth to teach most undergraduate computer science courses, and I look forward to the opportunity to share my knowledge of the subject with younger people. Courses on Introductory Programming, Data Structures and Algorithms, Programming Languages, Compilers, as well as seminars on Program Analysis and Program Verification, are among those that I would be particularly interested in.