It is always fascinating to me that without any magic like mind reading or knowledge inheritance, we can pass on the knowledge between generations and build a civilization through verbal communication and writing. Nowadays even with all the slides, video, or even virtual reality, to me the art of teaching still boils down to the direct and sophisticated interaction between the teacher and the student, with exchanges of questions and answers. Most importantly, the roles are not static, but constantly changing – the student becomes a teacher when answering peer's question, and the teacher could also learn from the students. I believe capturing this evolving dynamics is the key to bringing an inspiring, inclusive and attractive education to the next generation.

From my past teaching experience in the past decade, I found myself truly enjoying this charming interaction: taking interesting questions, writing on the blackboard with the chalk, discussing research questions with the mentee, etc. And I gradually developed my teaching philosophy, which can be precisely summarized by an old Chinese phrase: better teach a man how to fish than give him fish (**授人以鱼不如授人以渔**). I believe teaching is more about enabling them to ask important questions, inspire their own understanding, and develop a curiosity that lasts for a lifetime. Here I will give my two cents with my own experience.

As a High School Student: Learning by "Making" Homework

As one of the top students in math and physics during my high school, I got a lot of questions from my classmates about how to understand a concept or solve a homework problem. After answering their question, I usually tried to find a similar problem to help them better digest the content. Soon I started to make my own "extra" homework for my friends to practice. It is so challenging and rewarding to jump out of the box and think from the teacher's angle – what are the obstacles to understanding the theorem, how this problem reinforces the concept, etc. Gradually my teachers let me substitute for them when there was an emergency. Besides, this never interfered with my other studies, but actually improved my grades. Eventually, I was admitted to Tsinghua University, which is one of the top universities in China.

This experience made me realize that **teaching is the best way to learn**. It forces you to first have a systematic and thorough understanding of the subject before trying to teach others. Thinking and practicing like a teacher also helped me improve my presentation skills. Also, as a Chinese adage goes, the truth will emerge clearer from debates. During this process, you are constantly challenged by the audience, which helps you clear some misunderstandings and fill the gaps in your knowledge system. This is such a valuable experience for me, and if I had the opportunity to teach the next generation, I would encourage them to form study groups and learn from each other. They could also take turns to lead the discussion session with me firing questions and as a backup. I believe this will help them to have a deep understanding of the material, and they could benefit from this learning method for their entire life.

As a Teaching Assistant: Becoming a "Questioning" Machine

As a senior Ph.D. candidate, I finally got the opportunity to work as a teaching assistant for an undergraduate course: CS33 Introduction to Computer Organizations. This is a hard course as it covers low-level details in hardware and assembly code, and some tricks to manipulate bits are even challenging for an expert like me. I still vividly remember during my first office hour, a student came with a new problem that even I had no clue about how to start. Instead of getting back to him later, I started to work with him together. About halfway I figured it out, but kept guiding him with more and more questions, and the sense of accomplishment when he finally solved the problem is priceless.

Since then I realized that as a teaching assistant, I should not be an answering machine that simply spits out the solution, but a "questioning" machine that guides and inspires the student to develop their own understanding. During my office hours, instead of giving away the answer, I preferred to give them hints by asking "what if" questions, and it brought so much joy to me *and* the student when they figured it out on their own. I also pay attention to the frequently asked questions and fired them back to the students during my discussion session to strengthen their impression.

These approaches worked really well, and I got *8 out of 9 points* for my TA evaluation. One of my students wrote in the evaluation: *"Very knowledgeable, helpful, and encouraging. He ensured that I understood the content."* I can not emphasize enough how much I enjoyed this experience, which made me realize that **teaching is to inspire everyone**. By asking questions, students gradually develop their own understanding. This may look less efficient than simply feeding and remembering everything, but every student is different and has different difficulties. Only by asking questions can we find and clarify the misunderstanding, which is crucial to *inspire everyone*. In my future teaching, I would continue this style and try my best to build an inspiring and inclusive class for everyone.

As a Research Mentor: Ask the Important Question

As I spent more years in graduate school, I got the chance to mentor a few talented undergraduate and master students on various research projects. Such research mentorship is very different compared to regular teaching, but more like collaboration. Since it was usually the first time for the mentee to conduct a research project, at the beginning I had to offer a lot of direct help on setting up the environment, debugging, or even implementation. However, as the project went on, I gradually started to let them independently investigate and solve the problem. During this process, they showed great efforts and contributed a lot to some of my work.

Research projects can be hard and intimidating, especially for those undergraduates. As a mentor, my responsibility is not telling the answer (cause I am not sure either), but offering some possible solutions, guiding them on the right track, and most importantly, **keeping them interested and motivated**. To achieve this, I believe the core is **teaching them to ask the important question**. It is not uncommon that the mentee gets distracted by other implementation details and loses interest when having little progress, and it is important for me to help them ask the important research question and push the project forward. And only through this process, they learn how to conduct a research project independently in the future. For my future research, I would continue to offer my help to my future students.

Classes to Teach

Based on my expertise, I could teach courses related to computer architecture and organization, including undergraduate-level introduction to computer organization, computer system architectures, and graduate-level advanced computer architectures. Since my research heavily studies parallel high-performance workloads, I could also teach parallel computing class about how to write a high-performance race-free parallel program. Introduction to algorithm and data structure, compiler construction, and introduction to operating system are also candidates since I am familiar with these technologies during my research and have first-hand experiences building or extending them.

Classes to Develop

Since my research focuses on the intersection between the data and computation in modern architectures, I believe it is important to develop new courses to educate the students on the latest advance in the industry and academia. Besides the classical computer architecture courses, if hired as a faculty member, I would also like to develop two graduate-level courses: a seminar on near-data computing, and a course on non-conventional computing platforms.

For the seminar, it aims to cover the history, the motivation and the latest development in near-data acceleration. I will organize a hybrid class in which I first give some high-level introduction, followed by a discussion on classic and latest publications led by students. One week before the class I would distribute the paper with some key questions for them to think about while reading it. Also, I will also incorporate the angle from the industry: what are the tradeoffs for chip companies like Intel, Nvidia, etc. to adopt near-data computing, and what are the recent technology advancements?

For the non-conventional computing platforms, I will focus on emerging programmable platforms, including FPGA, GPU, and spatial accelerators. Each platform employs a different programming paradigm, and students will get first-hand experience programming them to accelerate important and widely applied kernels, e.g. machine learning, graph processing, linear algebra, etc. Through this practice, students will compare the performance of different platforms, and better understand the low-level hardware details and write more efficient code in their future research and career.