ZHIYAO DUAN TEACHING STATEMENT

In the past eight years, I have served as a lab instructor of two computer programming courses; a teaching assistant and guest lecturer of an artificial intelligence course and three machine learning courses; a research mentor of two master's and two undergraduate students; and a euphonium tutor in a wind symphony ensemble. These different experiences gave me insight into teaching, and helped shape my teaching philosophy.

1. **Foster passion**: I think fostering passion is the most important duty of a teacher. I have seen many talented peers in my college who lost passion for a subject and finally gave up learning. I have also seen how students were inspired by my advisor in his class and furthered their study into independent research.
2. **Promote active learning**: I deeply agree with the theory of "the purpose of teaching is not to teach"; instead, I think students' active learning is the ultimate goal.
3. **Tailor to students' aptitude**: Each student is unique. They come from different backgrounds and have different strength and weakness. I think a good teacher should interact with students, pay attention to their feedback, and design teaching/learning agenda in accordance with their aptitude.

TEACHING EXPERIENCE

I apply my philosophy to my teaching activities. I was the weekly lab instructor of two undergraduate courses on fundamentals of computer programming using C++ at Tsinghua University. Since practice is the only way to learn computer programming, I set up my goal as making sure they grasped the key knowledge points through lab practice. To achieve my goal, in each lab session I always reviewed the key knowledge points taught in class before assigning new lab tasks. Furthermore, I taught one debugging skill (e.g. setting a breakpoint or tracing a variable) in each session using previous lab examples. I knew that with these skills, they would be able to develop their programs independently and experiment with new techniques actively. My instructions especially those debugging skills were appreciated by students, as they fostered students' passion to actively explore computer programming.

While students' understanding of basic knowledge can be assessed through well-formulated lab assignments, their capability of integrating and applying the knowledge can only be assessed through projects. The course instructors and I decided to challenge the students to solve their favorite real-world problems as the final project. I reviewed students' proposals, discussed their ideas and technical issues with them, and graded their final projects. Students' passion was further fostered through the project as they could see the real impact of studying the subject. The outcome was beyond our (and students') expectations. A number of interesting projects, e.g. a computer game Othello, an instant messager, a calculator, etc., were accomplished. Not only were their basic programming skills practiced, but more advanced topics such as data structures and algorithms were explored through their active learning.

Three times during my PhD study at Northwestern University, I was a teaching assistant of the machine learning course, working with two professors. This course was an advanced undergraduate/graduate course for those who wish to apply machine learning to their research. Over the years, I have designed and given multiple lectures on different topics, including ensemble learning, memory-based learning, Gaussian mixture models and expectation maximization. I believe that lectures should be tailored to students' aptitude. Take the Gaussian mixture models lecture as an example, the idea of modeling data distribution with a mixture of Gaussians is intuitive, but a thorough understanding of how the algorithm was derived requires solid math skills. Since students in this course came from diverse backgrounds (e.g. engineering, social sciences, math), when I designed my lecture, I kept intuitive ideas as the main thread and dived into deep math from time to time. In this way, all students were able to follow the lecture and see the big picture, even if some were lost in some math derivations. In grading their homework, I gave detailed comments to each student on each mistake he/she made. In this way, students' learning efforts were respected and their unique problems were addressed.

I believe that group discussions are effective to foster passion and promote active learning. Together with another teaching assistant, this Fall I set up an online study community for students to ask/answer questions and exchange ideas. On average each week there were about 10 new threads and 30 follow-up posts. Some posts involve discussions of materials beyond the scope of the course, showing students' passion in active learning. The online community was also a nice place to form groups for the final project, which was research oriented and required students to find, formulate and solve a real-world problem. Students used the online community to advertise ideas and recruit team members. Students' ideas were also inspired by each other.

MENTORSHIP

During my PhD study, I have had the pleasure of mentoring two master's students and one undergraduate student. I believe that a key issue in mentoring students is to foster their passion by giving them enough freedom for exploration. Take the master's student as an example; he worked on an independent research project about designing a real-time multi-pitch estimator for guitars. He came to me with a very intense interest and an ambitious goal. He quickly proposed an idea and wanted to test it directly on complicated guitar recordings. As an experienced researcher in this field, I clearly knew that his idea needed to be improved by first testing on some simple examples such as isolated chords. However, I did not stop him, but just told him my thought. After two weeks, he realized that his idea was naïve. I then encouraged him that even experts often start with a naïve idea and then rigorously test and improve the idea. After nine months' hard work, our efforts paid off. This project resulted in a successful computer program and a technical report.

EXAMPLE COURSES

Here are several courses that I feel I would be strongest in teaching. Courses 1-3 are commonly provided in many universities. Courses 4-6 are new courses that I would like to design.

1. *Principles of Programming with C++:* (undergraduate) A practice-oriented introductory course. Students practice programming skills through weekly programming problems and a final project.
2. *Digital Signal Processing:* (undergraduate) Introduces basic concepts and techniques for processing digital signals. Emphasizes mathematical derivations and practical implementations. Covers discrete-time signals and systems, discrete-time Fourier transform, z-transform, discrete Fourier transform, digital filters.
3. *Machine Learning:* (graduate) Introduces theory, algorithms, and applications of machine learning techniques. Weekly problem sets with programming assignments. Students also work in groups on a final multi-week design and implementation project to exercise course topics with real-world problems.
4. *Machine Perception of Music:* (undergraduate) A project-based course that studies machine extraction of musical structure in audio, MIDI and score files. Topics include music representation, analysis, synthesis, effects, pattern extraction, source separation, and music retrieval.
5. *Computer Audition:* (graduate) Fundamentals on computational models for analyzing auditory scenes. Topics include human auditory system, perceptual cues, auditory scene segmentation and grouping, computational models for pitch perception, source separation, streaming, and localization. Students work on a final project to exercise course topics.
6. *Machine Learning for Signal Processing:* (seminar) Recent advances in data driven approaches and machine learning techniques for a variety of audio, speech and image processing problems.