TEACHING STATEMENT
Du Li (duli@cs.cmu.edu)

I believe the value of knowledge is multiplied when we share and pass on knowledge from one generation to
the next. This is the main reason that motivates me to seek an academic career. My mission as an educator is not
only conveying knowledge to the students, but training them to obtain new skills, developing their ability to solve
problems, and inspiring them to explore the unknown. Teaching is challenging, requiring very good capabilities
to organize and present information in a creative and comprehensible way. Teaching is mutually beneficial as the
interaction with students can lead me to advance the understanding of the subject and to discover new insights.

1 Teaching Philosophy

1.1 Teaching students in accordance with their aptitudes.

I served, for one semester, as a teaching assistant for the Data Structures and Algorithms (CSCE310) course
when I was a PhD student at the University of Nebraska-Lincoln. My responsibility was to grade students’
homework and discuss the results with them afterward. CSCE310 is an introductory course, so students taking
the course can have diverse backgrounds (e.g., from recently arrived international students to students who
competed in ACM Programming Contests). For many, this is their first computer science course. Given such
diverse backgrounds, I realized that out-of-classroom support must be tailored to fit the student’s needs. During
my office hour, I encouraged students with advanced knowledge about programming to optimize their solutions
in terms of better time and space complexity to get extra points. For students who were new to the subject, I
carefully explained the lectures in detail and guided them to finish their homework. My goal was to ensure every
student can learn as much as they can during the class.

Globalization is changing the background of college students in many ways. Currently, the majority of
computer science graduate students in US institutions are from other parts of the world. Recently, this trend has
extended to undergraduate programs. The students come here with different knowledge, skills, aspirations, and
cultures. It is important for educators to appreciate the diversity and differentiate their instructional methods
and contents to benefit most if not all students. For instance, Asian students tend to be more reserved and
passive even when they have questions. During the course of CSCE310, I kept encouraging several international
students to speak up. This helped them to overcome the culture shock very quickly so that they could succeed
in the course. Another student in the class had 10 years of industrial experience. He was having a difficult time
understanding the theory portions of the courses due to insufficient mathematical background. However, I found
that an effective way to refresh his mathematical knowledge was express those algorithms in terms of real world
programming examples that he was familiar with. In doing so, the theoretical aspects of the course became much
more accessible for him. My experience with this course has taught me the importance of providing differentiated
education for students.

1.2 Learning by doing.

Besides introductory courses, I also co-taught a graduate level course, Advanced Runtime Systems (CSCE990),
with my PhD advisor, Dr. Witawas Srisa-an. This course covered several runtime systems including memory
management, just-in-time compilation, and virtual machine implementation. In the beginning, I introduced the
basic concepts by assigning landmark papers for students to read as part of their assignments. I then asked them
to lead the discussion of those papers in class. I realized that reading alone is insufficient for learning as often

http://www.cs.cmu.edu/~duli
times, students do not have a clear understanding of how some of the concepts actually work (e.g., read/write barriers).

To make learning more effective, hands-on experience is needed. As such, I came up with a project that asks students to implement a dynamic escape analysis within the Jikes RVM, a research Java Virtual Machine. This project actually incorporated most of the concepts we taught in the class like garbage collectors, dynamic compilers, and so on. It also involved substantial hacking of JVM internals. The students finished the project surprisingly well within only three weeks. More importantly, the students all felt they had concrete understanding of runtime systems based on the hands-on experience with a real world JVM.

As they had already gained basic knowledge, students could decide the topics of their final team projects. Students were advised to find topics that could connect with their research backgrounds. One student chose a project that exploits the interplay between memory management and computer architecture to improve cache locality. Later on, this project developed into his dissertation work and I am still collaborating with him.

Plenty of studies have demonstrated that learning-by-doing is a most efficient way to assimilate new knowledge, which is also confirmed in my teaching experience for course CSCE990. Well-designed course projects can help students to grasp key concepts of a course, and enhance students’ practical problem solving ability in ways that will help them in their future careers. Learning-by-doing is particularly useful for computer science education.

2 Teaching Interests

I am interested in teaching in various areas of computer science. Given my teaching experience and knowledge of computer science, I believe I am qualified to teach software as well as systems related required courses at the undergraduate level including Automata and Formal Languages, Compiler Construction, and Operating Systems. In particular, as the demand for software engineers is on the rise, CS and CE students are expected to have software engineering as part of their undergraduate training. I have studied and worked in two nationally recognized top software engineering programs at the Carnegie Mellon University and the University of Nebraska-Lincoln. I would like to further strengthen the software engineering curriculum by teaching a series of advanced courses on Principles of Software Design, Software Quality and Testing, and Software Process Management.

In addition to developing undergraduate courses, I would like to develop new graduate level courses in the areas of programming languages, runtime systems, and software engineering that are closely related to my research. I would like to create a course on concurrency and parallelism to uncover the challenges at the programming language level due to the prevalence of multi-core systems, discuss cutting-edge research and present the underlying theoretical foundations. I would also like to create a new course in runtime systems in the setting of mobile and cloud computing as they are becoming mainstream. I can also design a course on program analysis that covers both static and dynamic analyses, and how to use program analyses to address software dependability and security vulnerability issues. I expect this course can lead to interesting research projects and dissertation topics.

http://www.cs.cmu.edu/~duli