TEACHING STATEMENT
LEEN-KIAT SOH

1. INTRODUCTION
I have devoted considerable time and efforts to teaching, undergraduate mentoring, graduate mentoring, and curricular development since joining the UNL in August 2001. I continue to improve the undergraduate and graduate courses that I teach, recruit UCARE—the Undergraduate Creative Activities and Research Experience program at UNL—students and undergraduate research students, and mentor graduate research students. I participate in several curricular revision activities, pedagogical activities, and CS education workshops and committees. I have been recognized by the department and the university for my teaching and contributions to students:

- UNL Parents Association Certificate of Recognition for Contributions to Students (2014)
- College of Engineering Distinguished Teaching Award (2012)
- Holling Family Master Teacher Award/UNL University-Wide Teaching Award (2012)
- College of Arts and Sciences Distinguished Teaching Award (2006)
- two CSE Department Student Choice Outstanding Teaching Awards (Lower Division Courses) (2008, 2004)
- Certificate of Recognition for Outstanding Contribution to Undergraduate Research (2008)

Note that in the following, I have separated my work in CS education research from this statement. Please refer to Appendix B for letters from students and faculty on my teaching and Appendix E for a course listing and course evaluations.

2. IMPACT HIGHLIGHTS
Here is a summary of the impact of my teaching activities:

- I have offered 100-level courses 15 times with an average of 4.15 (average enrollments = 41.4, maximum enrollments = 94), 200-level courses 6 times with an average of 4.31 (average enrollments = 42.7, maximum enrollments = 83), 400/800-level courses 14 times with an average of 4.38 (average enrollments = 14.1, maximum enrollments = 29) and 900-level courses 2 times with an average of 4.53 (average enrollments = 6.5, maximum enrollments = 8). The department averages, since Spring 2007, are 3.44 for 100-level courses, 4.06 for 200-level courses, 4.39 for 400/800-level courses, and 4.72 for 900-level courses.
- I have worked with twenty-two (22) undergraduate students in total and received 25 UCARE awards from UNL. I have also worked with two NSF REU undergraduates for my research projects. Five of my undergraduate advisees wrote and successfully defended their undergraduate senior thesis.
- I have graduated seven Ph.D. students (six in Computer Science and one in Instructional Technology)—four of which as the main advisor, and three as the co-advisor—and twenty-three (23) M.S. in Computer Science students (as the main advisor).
- One of my Ph.D. students, Adam Eck, received the prestigious NSF Graduate Research Fellowship award for his research. The fellowship provided more than $123,000 in funding over three years for Adam to do research in resource-aware multiagent sensing. Adam Eck also won the CSE Department Outstanding Ph.D. Student Award in 2015.
- I have published more than 110 papers (31 journal papers, 67 conference papers, and 10 workshop papers, plus book chapters and technical reports) with my advisees and more than 25 papers with graduate students advised by other faculty working on the same projects, including publications in premiere journal and conference venues.
- I have developed and prepared for several introductory core courses such as CS1 in different contexts, CS2, and CS3, three upper division courses, and one advanced graduate course. I have improved

---

1 100-level courses: freshmen courses, 200-level: sophomore; 400/800-level: senior and graduate; 900-level: advanced graduate
each course that I have taught. For the learning objects that we built to support CS1 courses—under the project titled iLOG, Professor Gilliean Lee of Lander University, Greenwood, SC, first adopted the learning objects for his class for Fall 2011, and e-mailed us the following:

“In the last semester, for the first time in 5 years, I had all the students received passing grades, except one who didn’t attend lectures in the last month. I believe the iLOG was important part of the success, I am truly thankful to you for that.”

- As a key member of the Re-Inventing CS Curriculum Project, I helped define the innovative curriculum changes in CS0, CS1, CS2, and CS3, having been heavily involved 2002 – 2006. I was instrumental in the design of the placement examinations, learning objects, and laboratories for this project. We were able to implement the changes department-wide in Fall 2003. Though the laboratories have undergone changes over the past 12 years, the same basic problem-based and inquiry-driven approach is still ingrained in all our laboratory activities.

- In 2008, as leader of the Renaissance Computing project, I successfully overhauled our introductory CS courses by upgrading CS0 courses—and creating new ones—to a suite of CS1 courses flavored in engineering, sciences, arts, and humanities, after working with a dozen departments across campus to understand their needs and challenges. This also has helped create the interdisciplinary Computational Biology and Bioinformatics minor program. The CS1 courses, which are still being used today, have undergone several revisions.

- All these have helped increase the number of non-CSE students taking CS courses, as reported by Stephen Scott (our Department Vice Chair) Fall 2014, with the number of non-CSE students enrollments up by 54% in 400-level courses, 22% in 800-level courses, and 29% in all other courses. My efforts in curricular development have helped improve our curriculum for CSE students and increase our footprint and services campus wide.

- In 2012, we started the IC2Think project, integrating Computational Thinking and Creative Thinking into our CS1 courses. Our suite of IC2Think exercises have been shared at the NSF-sponsored Ensemble portal for computing and adopted for K-12 students on Google’s Exploring Computational Thinking website. These course materials have the potential of changing the way we teach CS and other STEM subjects, as well as broadening participation in computing because of the non-programming creative problem solving exercises.

- I was an invited participant to the Future Directions in Computing Education Summit 2014 where a selected group of about 30 researchers met to discuss ideas, challenges, and opportunities in future research and curricular development in computing education.

3. PHILOSOPHY

3.1. Teaching and Undergraduate Mentoring

A. My Overall Goals for Student Learning. My goal as a teacher is to transfer my knowledge and experience effectively and enthusiastically to my students. I want to share with them what I have learned. I want to first guide them in solving problems and later let them be independent and enjoy tackling other problems themselves. I want to equip them with insights so that they can apply them to whatever areas and domains pertinent to their careers down the road. I want my students to be confident in what they have learned specifically, as well as to be confident in the student and professional careers in general. I want them to be passionate about what they learn, and hopefully that transfers to their individual lives. I want to motivate them to not only want to learn, but to enjoy learning! My goals for student learning are: (1) to prepare my students as engineers or scientists, (2) to get my students excited and eager to learn, (3) to encourage my students to realize the importance of both fundamentals and creativity, (4) to design classes so that my students can learn to work both as a team and as an independent problem solver, (5) to present and discuss current, published works of other researchers in related areas so that my students will be kept up-to-date with the latest technologies and research interests, (6) to introduce new classes that involve emerging research disciplines, and (7) to enhance my classes using real-world applications and examples from my research experience.

B. Skills I Aim to Teach. The skills that I aim to teach the students are skills that enable them to be successful in their student and professional careers and other endeavors later in life. I aim to teach them
problem solving skills and learning skills. I want the students to be able to identify where a problem is, describe what the problem is, understand why the problem occurs, and know how to solve the problem. This involves pattern recognition and identification of interconnections. I also want the students to be able to analyze a piece of knowledge critically, to be able to praise or criticize a particular solution. Only then, they will be able to refine their knowledge. I also want the students to be able to figure out when their knowledge needs to be refined. I also point out how we can appraise a particular solution: advantages and disadvantages. Finally, I want to teach them to be able to be resourceful by being inquisitive and tenacious in problem solving. Often times, students give up at the first sign of impediments in their problem solving process. I want to show them the process is not intimidating if one is resourceful. And one can be resourceful in many ways, e.g., by thinking from another angle, by being creative, by thinking outside the box, decomposing a problem into smaller subproblems, trial-by-error, solving a smaller problem and generalizing the solution, and so on. And I do that through my lectures, assignments, examples, and handouts.

C. Methods that I Use. I have used several methods, with the underlying approach of making the subject topic more interesting, more “real-world”, and easier to grasp. I also give a lot of examples, a lot of feedback, and a lot of questions to make students think. My lectures are highly interactive—I prefer writing on the whiteboard, and interacting with the students as I go along. I also give a lot of handouts to summarize key points or concepts, or issues that the textbook does not cover or does not cover well. I also give quite a bit of homework assignments to make sure that students get a lot of practice conceptually and in programming. I also tend to my e-mail very responsively. Usually, I respond to their questions immediately, within 24 hours. These are my basic methods. In addition, for different courses, I also have unique methods depending on the number of students and the makeup of the students. Here I will list some of the methods that I have used in my courses: (1) game days – where students act as “software agents” participating in a multiagent environment and learn about how agents negotiate, compete, and cooperate among themselves to achieve global goals (my Multiagent Systems course). (2) ask-your-neighbor – where students have to convince their classmates that their solutions to in-class short problems are correct (my CS1 course). (3) forums – where students break into groups to come up with solutions (my CS1 course), (4) seminars – where students need to present papers and other students in the audience have to ask questions (my Information Retrieval and Multiagent Systems courses), (5) interesting assignments such as writing a tutorial on permutation (my Discrete Structures course) and building themed programming solutions (e.g., Lord of the Rings, Star Wars, etc., for my CS1 course), and (6) test programs – short program modules that allow students to experiment with (my CS1 course).

Further, for all my courses, I require students to pick up their exams personally from me. That is, if they do not pick up their exams from me, their exam scores will be recorded as a zero. During each individual one-on-one meeting, I go through each problem with the student, identifying what he or she did wrong and why and how. For students who did well, I commend them and point out other things that they could do to further improve. For students who did not do well, I discuss with them about their course schedule, their use of time, their other activities, trying to identify the aspects that they could do to help improve their scores in the next exams. For large courses such as CS1 and Discrete Structures, since I wanted to return all the graded exams within one week, I held additional office hours for the entire week after each exam so that students can visit me in my office to pick up their exams.

D. My Views that Encompass the Whole of My Teaching Assignment. Overall, I have five major views that encompass the whole of my teaching assignment: (1) Different Students Have Different Needs: Some students need to be pushed more; some need to be encouraged. For example, there are very motivated students whom I tend to give more advanced comments in their assignments, hoping to encourage them to probe further. There are students who are not motivated whom I tend to give more examples in their assignments, to make things convenient for them; (2) Fairness: This translates to my grading of the assignments, and to my teaching and treatment of students. Students will be rewarded fairly based on their effort and understanding of the subject topics; (3) Tough Love and Integrity: I strongly believe in tough love—no easy grades and no easy classes. I believe in assignments that challenge the students. However, they should not be overly difficult that they become unfair. I believe likewise in personal integrity and course integrity. This translates into quick turnarounds in returning homework assignments, providing ample information for final projects, specifying requirements, etc. This also translates into my
expectation of the students: no cheating, good effort, plenty of interactions between them and me, and good initiatives; (4) Fun and Games: I strongly believe that lectures should be fun and should incorporate games if possible. For my Discrete Structures class, I used the idea of Lightbulbs—examples ranging from Chevy Chase’s National Lampoon to Monty Python’s Quest for the Holy Grail, from triathlons to basketball. For my Multiagent Systems class, I have designated Game Days for role acting, and designed my final project as a Fox-and-Hound game. For my CSI classes, I use forums – we act out sorting algorithms, exception handling processes, etc., and we also have group activities to come up with conceptual solutions for disaster relief and building smart homes; and (5) Research and Applications: Incorporating research activities into teaching has many benefits, ranging from the increased confidence of the instructor to a sense of purpose in students’ work. It shows students why people are interested in the subject matter, why and how it is challenging, and what we can do about it. This gives them something concrete and more relatable to grasp.

E. How I Assess Student Learning. Since I focus on reasoning and problem solving and how a student arrives at a solution, I do not give multiple-choice questions. All my questions are open-ended with a majority of the questions in the level of application, analysis, synthesis, and evaluation, instead of pure recall and comprehension. I use quizzes, exams, presentations, game days, group activities, topic summaries (e.g., “stupid questions”), term papers, programming assignments and labs to assess student learning. Whenever I grade their solutions, I also look at their reasoning process. I give partial credit to logically sound reasoning.

3.2. Graduate Mentoring

A. Goals. My goal as a graduate mentor is two-fold: (1) to shape and mold my advisees in particular and other graduate students in general to become an independent, dedicated, and resourceful researcher and problem solver, and (2) to give them the best opportunities to succeed in their career during and beyond their graduate studies. I want them to learn about themselves— their strengths and weaknesses, their skills and how to improve them, their outlooks and aspirations. I want them to be able to chart their own career paths, graduate, and excel.

Under Goal #1, specific objectives are (1) to teach them about doing research and the skills needed (communication, critical thinking, abstraction, etc.), (2) to manage their priorities and time well and the skills needed (organization, time management, etc.), and (3) to have a well-balanced experience as a graduate student. Under Goal #2, specific objectives are (1) to support student travels to conferences to present their work and interact with other researchers in their areas, (2) to strengthen their qualifications by involving them in opportunities such as proposal writing, workshop organization, serving on conference program committees, (3) to help them prepare for their job search.

B. Methods. My graduate mentoring methods vary depending on individual students, their needs, aptitudes, and personalities. However, common characteristics of my methods include:

- **Leading by example**, giving them ample examples to observe and learn about the research process as well as commitment and priority management; for example, working in the trenches with them on writing technical papers, teaching them how to articulate thoughts into coherent writing;
- **Leading by trusting them**, letting them assume responsibilities; for example, with matured graduate students, regarding them as collaborators and encouraging peer-like discussions and idea sharing;
- **Establishing a friendly atmosphere and positive environment**, recognizing and appreciating their contributions and research progress; for example, having off-campus get-togethers and recreational activities, and facilitating research groups so that students can enjoy peer support;
- **Setting high expectations, pushing students towards achieving their goals, while providing support**, e.g., meeting regularly and having an open-door policy, giving my students ample access on week days and nights and also weekends;
- **Training them in time management**, e.g., planning and scheduling milestones and deadlines for regular checkpoints, allowing students to be guided and structured in open-ended research activities, and teaching them about prioritization;
- **Most importantly, caring about them and about their lives and dreams and their families**; to me, my advisees are more than just students, they are friends and colleagues.