# TEACHING STATEMENT

**Ashish Sabharwal**
**Cornell University**

My goal as an educator is to introduce students to the fascinating concepts and challenges in the field of Computer Science and to prepare them to use their knowledge to solve practical problems in traditional application areas as well as newly emerging ones such as computational sustainability.

Free flow of ideas is a cornerstone of academics. It appears in various forms during peer-to-peer communications, in advisor-advisee relationships, and most pervasively during student-teacher interactions. In a constantly evolving field at the forefront of technology, the task of an educator demands an active interest in ensuring that the concepts and the ways of imparting them stay fresh. To be a successful teacher one must understand and go through *a constant cycle of preparation, implementation, and evaluation*. I share below my experiences with these, drawing upon several refreshing and enlightening teaching opportunities, particularly as a *lecturer* at the University of Washington (2003) and as a *tutorial presenter* at the Conferences on Artificial Intelligence (AAAI-07/08), the Defense Advanced Research Projects Agency (DARPA, 2007), the Kavli Institute for Theoretical Physics in China (KITPC, 2008), and the Banff International Research Station for Mathematical Innovation and Discovery (2009). My views also draw upon the years that I have spent at Cornell University as a researcher interacting with several *Ph.D. students* in a unique role that delicately combines collaboration and guidance, and several invited *book chapters and surveys* that I have written in order to clearly and comprehensively introduce the state of the art in my research area to other researchers and students.

**Preparation**: Preparation for teaching in a natural and clear manner begins by developing a deep understanding of the subject matter. Thankfully, both the graduate curriculum at the University of Washington as well as the rigorous coursework at the Indian Institutes of Technology emphasize *breadth of knowledge*. They require courses from a large number of diverse and important fields not necessarily directly related to a student's specific research interests, and have thus prepared me to comfortably teach a broad range of classes at the undergraduate level. My own research is in Artificial Intelligence, with strong connections to Theoretical Computer Science. Teaching standard as well as newly designed courses in these areas will be a delightful opportunity for me to share my knowledge and enthusiasm. At the same time, I also look forward to developing and teaching an introductory course on applications of computer science techniques to addressing pressing environmental and sustainability concerns of our time.

As another important step in long-term preparation, I have strived to *learn how to teach* because sufficient knowledge doesn't necessarily make one a good teacher. I have frequently worked with the Center for Instructional Development and Research (CIDR) at the University of Washington. I have participated in training programs and discussions on how to be effective in a classroom where students have varied backgrounds, expectations, and skill levels.

**Implementation**: As a Ph.D. candidate, I took up a rare opportunity for hands on experience: teaching an undergraduate Data Structures course as a lecturer. The task involved lecturing and organizing a class of nearly fifty students and managing workload with two teaching assistants. My goal was to make this foundational course stimulating, exciting, and challenging for the students. I achieved this by using a new *tablet-PC based interactive slide projection system* being developed at the UW, along with partial handouts that we filled together as each lecture progressed. This encouraged interaction, at the same time providing students concrete course material to take home. Having *lively programming goals* such as playing songs in reverse, solving mazes, and using word patterns to identify playwrights made traditional data structures projects fun to work on. Group programming projects and joint quizzes were my tools to make students learn to work productively as a team, an asset for any professional. Pop quizzes at the beginning of lecture were a lesson in punctuality for some. I broke the monotony of the class by having guest speakers give a glimpse of intriguing new topics such as the concepts behind Google's search engine and zero-knowledge data structures.

In retrospect, this was an invaluable experience. It helped me develop my own teaching style and

philosophy, and evaluate its efficacy. Prior to this assignment, I have been a teaching assistant for a variety of undergraduate and graduate courses, where I enjoyed the challenge of presenting lecture material in alternative ways during tutorial sections and office hours. I have volunteered for a one-on-one tutoring program for undergraduate students in need, and have substituted for lectures as occasions arose. During the past few years, I have prepared and delivered extensive tutorials on subjects related to my own research area, namely, the latest in automated reasoning technology. These tutorials have been very well received by seasoned researchers and new graduate students alike. The experience, along with several book chapters and surveys I have co-authored in the past few years, has taught me how to bring together scattered research ideas into comprehensive introductory teaching material on a subject.

**Evaluation**: The ability to mold one's teaching approach based on continuous feedback from students about their grasp and needs is crucial for effective teaching. Students' interest and understanding can be perceived indirectly from homework, attendance, and quality of questions raised in class. I personally like, in addition, the more active approach of feedback forms designed specifically for this purpose. One challenge in the latter method is to convince students that they are free to critique and that their comments will receive due attention. It is important, especially for early career educators such as myself, to understand that we are in the classroom to teach and, at the same time, to continuously learn and evolve ourselves as a teacher. For the Data Structures course mentioned above that I taught, I utilized the services of CIDR at the UW to obtain concrete comments from students in the middle of the quarter, and was able to make some immediate changes based on their needs. The final survey at the end of the quarter revealed that the students had liked aspects such as *instructor's enthusiasm and knowledge, relevance of course content, and availability of help* when needed. Their overall evaluation rating was quite high at 4.2 out of 5.0.

In addition to student feedback, one must also evaluate over time the course content itself. As computer science rapidly progresses, it is critical to scrutinize the relevance and comprehensiveness of the course material. I believe this is best done by incorporating into it *a flavor of recent research results* and by looking at the way the course is taught by other lecturers and at other places.

While the challenges and the approach to teaching vary to a great extent from one discipline to another, certain key aspects are invaluable irrespective of what one teaches and in what context. My own teaching philosophy has been influenced enormously by a seemingly unrelated subject—the practice of martial arts, which frequently requires me to lead groups of beginner and moderately advanced students. Over the years, this has consistently emphasized *leadership, clarity, patience, and the ability to inspire passion* for the subject—qualities that are in my opinion essential for the success of any teacher in any field. Advances in human society rely critically on the transfer of various kinds of knowledge and wisdom from one generation to another. To accomplish this goal is the responsibility of the teacher. I believe that I have prepared myself well for this task in the field of computer science, and I look forward to it.