Teaching Statement

Ashish Mishra

Three main goals can best describe my teaching or pedagogical method. These goals are based on my experiences, first as a student and later during my experience while TA'ing, teaching and working with students as a postdoctoral researcher. These goals are:

- Developing sufficient intrigue in students to follow the reading material for the class.
- Evolving the available course structures and teaching methodologies to meet the changing research and job market requirements.
- Creating an environment for learning where students feel comfortable asking questions and giving meaningful feedback.

Following are some of the most relevant teaching and mentoring experiences I had during my years as a PhD candidate and later as a postdoctoral researcher.

Teaching Assistant: Programming Languages Design and Implementation, Fall 2016: During my PhD at IISc, I had several chances to assist professors in their courses. Fortunately, the TA'ing involved not only assessing the assignments and helping in designing exam questions but also occasionally involved designing and teaching crucial topics I learned during my research.

For instance, I was the TA for the course "Programming Language Design and Implementation" in Fall of 2016. It was a graduate-level course offered by Prof. Y. N. Srikant at IISc. The course had an ambitious and novel goal; evolving a traditional "Compiler Design course" to the changing research directions in the PL community worldwide. This included adding programming languages and program verification topics, introducing functional programming basics, etc.

It was one of the first such attempts in IISc, and I was fortunate to be a part of designing/teaching some crucial topics. As a concrete example, since most modern programming languages' front-end includes a type-checking phase which varies according to the language paradigms, we included topics like *type systems* and *language paradigms* like *functional*, *imperative*, *dynamic*, *static*, *typed*, *untyped*, etc. I taught around 1-2 lectures covering these basics, followed by two full classes to cover *an introduction to functional programming* using Haskell.

To access students' learning, I helped Prof. Srikant design programming assignments (besides mid-term and full-term examinations) using interesting programming problems from several standard Haskell resources. One of the satisfactory experiences for me came from the fact that at least two students went on to work in programming languages and related fields for their PhD thesis.

Instructor: Lambda Calculus at IISs, CSA Summer School 2017: Another enjoyable teaching experience I had involved teaching an undergraduate course on *Lambda calculus* at the yearly CSA Summer School (2017). Approximately 100 undergraduate students attended this summer school from across the country. It was a unique experience as I had more freedom to design the two-class course. The task was challenging as students generally had no prior experience with programming language courses and came from varied backgrounds. It was in contrast to the traditional courses at IISc, which have students selected through the GATE examination and thus, the instructor can assume a baseline background. To address this variance, I took some critical steps to make the content accessible to everyone: Firstly, I decided to define a course on lambda calculus rather than a high-level course like functional programming; this obviated the requirement of a common programming language. Furthermore, to make the calculus concrete, I showed instances of Lambda Calculus in their favourite programming languages by actually building a few basic programs and showing live examples showcasing the power of lambda and finally relating it to Haskell and JS.

Courses I would like to offer for the changing times: I have gained some crucial insights regarding the courses and the teaching methodologies throughout my experience, first in the Indian academic setting at IISc and later in the US Universities at Northeastern and Purdue. For instance, I experienced that it is quintessential to have a good understanding of *logic* and its relation to *computation* for good research in program verification. Unfortunately, such a course is commonly unavailable for students in India at both undergraduate and graduate levels.
Given opportunities, I would like to offer such courses which are foundational and useful for research, along with other traditional courses like Discrete Structures, Data structure and Algorithms, Software Engineering, etc., at the Undergraduate level. At the Graduate level, I will like to mix more cutting-edge courses like Programming Languages and Program Synthesis with more foundational courses like Software Foundations and Principles of Type Theory.

Working with students: At Purdue, I gained valuable experience in working with graduate students. This came in two forms; first, I implicitly learned from Prof. Suresh Jagannathan through his approach to mentoring other lab students and me. There are several important things I picked up this way. For instance, to give helpful guidance, you need to develop good listening skills; Secondly, it is important to enabling students to take charge of their research while still being present to give high-level directions. This learning later helped me when I started working more closely with a few students in the lab.
I worked on two main problems with a few students in the lab. One of the big challenges and hence a significant learning opportunity in the process was ensuring that the student was making progress. It is challenging because there is no single rule which fits all students. Some students require a more hands-on approach with a closer look at their problems, while others may thrive more in a hands-off setting. The key, which I again learnt from Prof. Suresh, is to keep the communication with the student open and let the student candidly explain their challenges, possibly through concrete examples.
In summary, I have learned many concrete skills that will benefit both teaching and mentoring students. I have encountered several key limitations and challenges in these tasks and have learnt from some of the best mentors and teachers, both implicitly by seeing them work and explicitly by working with them. I plan to apply these skills and learning to my core philosophy and principle for teaching to achieve the three pedagogical goals I am striving to achieve.