

Teaching Statement

Mukesh Tiwari

July 3, 2020

Growing up in a small community of farmers in a remote village, without any electricity and road, of India and having just one school 4 KM away from my village, I never thought I would go for PhD in computer science and subsequently, for an academic position. I always wanted to be a farmer, but the situation changed overnight when I moved to a small town Mughalsarai, roughly 150 KM away from my village. In Mughalsarai, I started class 5 in a new school, and there was a teacher, who taught mathematics and moral science, had a profound influence on me. One thing she really taught me was how to think. At that time, I had no idea why I liked her teaching, and I just like the way she taught. Now that I have tools to analyse her techniques, one thing which I really liked is her clarity in teaching. Firstly, for every problem, she would passionately explain us the real life implication of solving this problem, e.g. given certain speed of car how much time it will take to visit the nearby city. Once we are motivated, then she would ask us to break down the problems, based on the material present in the textbook and what we studied in the previous classes, into small chunk, and then go for solution. All she did, like a skilled moderator, was help us in discovering the material by ourselves.

I used these techniques when I first started teaching at Institute of Information Technology, Bhubanesware, India (IIIT-BBSR) or at the Australian National University, Canberra. During the teaching, I faced numerous challenge in explaining the concepts to the students, coming from different backgrounds, and I would like share three incidents:

1. One course I taught at IIIT was C programming to first year student. In general, they have no programming background; however, they have good background in mathematics. I found teaching C programming to first year students extremely challenging for me and confusing of them, given their understanding of variables in mathematics. When I first used the word *variable*, they immediately connected it to mathematical variable. However, equations like $x = x + 1$ did not make any sense to them (some even went further to simplify the equation as $x - x = 1$ and exclaimed: how is it possible to have $0 = 1$). I was not surprise because when I was learning C programming in my first year programming course, I also had a similar realisation, and my professor was not able to clear my doubt (unfortunately, he was not familiar with Haskell). It ultimately led me to search a programming language which treats *variable* as a mathematical variable, and that is how I found Haskell. I took this as a opportunity to clarify the decade old misconception that variables in C are not variables in mathematical sense, and it is just a misnomer (I also encouraged them to explore Haskell). More importantly, I told them that their thinking is perfectly fine, and C *variables* are not variable (in mathematical sense), but they are simply memory location. To make it crystal clear, I told them that you can

think of C *variables* as a slate where you can write things, erase it, and then rewrite again. It was one, amongst many, such instance where my own curiosity (search of truth!) and learning helped me in paving a more informed path for the future generations.

2. The other problem I faced during teaching was engaging some students, who were not motivated to take the course. In the beginning of each semester, before teaching any class, I will memorize every students' name, so that I can connect with the whole class individually. Then I will tell them in advance that during this semester, after teaching some concepts, we will do practise problem, but the catch is that I will write some of the answers wrong on the blackboard for those practise problems (I like using chalk and board), and your job is to catch me red-handed. If you did, then you would awarded class participation marks, which would add to your final grades, and if not, then I will tell you next moment that the solution I explained was wrong (but in this case, no one would get participation marks). Not only it made the class more interactive, but it also improved the class participation. In the end of class feedback, they said that it was an excellent step to engage them in the material. Moreover, the felt very empowered.
3. In my PhD, I tutored the first year Logic Course at the Australian National University. It was very different experience than teaching in India. The class size was very small, which facilitated the individual interaction. Nevertheless, many students coming from countries, e.g. China, Vietnam, Malaysia, India, whose primary language is not English, were facing the usual problem with moving to a new country for the first time. I noticed that most of these students were not participating actively in the discussion, while the native (Australian) students were very active and prompt in the class. In order to increase the participation of the whole class, I encouraged every native speaker to sit with a non-native speaker and discuss the problem. Once, they feel that they have good understanding of problem, then discuss the solution. Although, I do not have scientific backing, I felt that collaborating with each other helped a lot in engaging the material.

During my time at IIIT, I taught C programming, Cryptography, and Algorithm Design and took C programming Lab. Because of my competitive programming background, I redesigned the C programming Lab and introduced them to competitive programming, which led to first ACM-ICPC team participation from IIIT. Moreover, it started a culture of programming, which ultimately led to participation in GSoC (Google Summer of Code).

While I would also be happy to teach outside of this list, current subjects I could teach include ¹: CSE1300: Reasoning and Logic, CSE1305: Algorithms and Data Structures, TI1316TH: Algorithms & Data Structures, CS4135: Software Verification.

Finally, my ultimate goal, as a teacher, during any class was not to immediately reach the solution, but develop the thinking process (problem solving mindset) in the students, and most of these ideas I got by observing my teachers, taking the Dr. Barbara Oakley's online course *Learning How to Learn*², and many videos from YouTube about how to teach effectively.

¹<http://pl.ewi.tudelft.nl/education/>

²<https://www.coursera.org/learn/learning-how-to-learn>