Teaching Statement

Michael D. Adams

Early in my graduate career at Indiana University I was supported by research positions and thus did not teach, but as I knew I wanted to pursue an academic career of which teaching is an important component, I made a conscious choice in my last two years as a graduate student to take teaching appointments instead of research appointments. I specifically requested courses in which the associate instructor[1] position involved lecturing and interacting with students instead of just grading behind the scenes. I also sought out courses that were early in the undergraduate curriculum as I felt that they would be particularly valuable experiences.

At Indiana University, I was an associate instructor for the honors introduction to computer science (H211, Fall 2010), both the honors (H212, Spring 2011) and non-honors (C212, Spring 2009) second-semester computer science courses, and the data-structures course (C343/A594, Fall 2009 and Spring 2010). For my work in the data-structures course, I received the department's Associate Instructor of the Year award. My responsibilities in these courses included preparing assignments and lab materials, running lab sessions, delivering lectures in discussion sections, grading assignments and tests, holding office hours, and mentoring students on semester-long group projects. The lab and discussion sessions that I ran had around 25–30 students each. Of course, as I was not the primary instructor, I was not the primary person responsible for the overall course design. I did, however, invent, design and implement several new assignments and labs for those courses. In the H211 course, I had a lot of input regarding the direction and design of the course as it was being revamped that year and the instructor collaborated closely with me on the course design. The students enjoyed the new course so much that they gave us thank-you letters at the end of the course. In the spring of 2011, I was a mentor for a student team in the Undergraduate Research Opportunities in Computing program at Indiana University. Next quarter there are plans for me to be the primary instructor for a graduate-level course in linear logic at Portland State University.

A strong influence on my teaching style is a belief in the importance of helping students see and understand both the practical application and the broader context of what is being taught. An example of this is an assignment that I designed for the data structures class (C343/A594). In lecture, students were learning about set and graph representations, and earlier in the semester they had learned about and implemented stacks. For this assignment, I wanted to

[1] At Indiana University, they use the title "associate instructor" where most other institutions would use the title "teaching assistant".

integrate these and give students an understanding of how these data structures fit into the design of a complete program. The assignment was to create a pattern matcher for regular expressions (i.e., something like "grep") that used a regular-expression syntax that was postfix notation (i.e. reverse polish notation) and was modeled using a non-deterministic finite automaton (NFA). Students implemented the assignment in two stages. In the first stage, the students constructed the graph for the NFA by parsing the regular expression using the stack implementations they had written earlier in the course. In the second stage, they implemented an execution engine for the NFA graph that matched the NFA against a string. They used a set to track which nodes in the NFA were active. This assignment allowed students to see how the stack, graph, and set representations they were learning in lecture applied to a realistic programming problem. At each stage, we discussed the different representations of these structures as well as the advantages and disadvantages of each. Afterwards, several students made a point of telling me how much they enjoyed that particular assignment as it helped them better understand the practical application of the techniques being learned. By seeing the context in which these techniques are used, students could more easily assimilate the knowledge being taught in the course.

I would be comfortable teaching any undergraduate course in the standard CS curriculum. Based on my training and background, I would be particularly well suited to teaching courses in programming languages and paradigms, data structures, algorithms, theory of computing, systems-level programming, and introductory computer science, as well as advanced courses in compilers, type systems, static analysis, and other topics classes in programming languages.

Quotes from Student Letters and Evaluations

"Thank you for being an outstanding AI this semester." (Alex Liby)

"It was a pleasure being taught by you." (Andy Spillman)

"The world needs more teachers like you." (Jordan Tritell)

"Although I am an art major, this class was definitely my favorite of the semester." (Brittany Keilly)

"Thanks for being the best AI that I've ever had." (Kate Sanders)

"Michael is absolutely fantastic." (Anonymous evaluation)

"He knows how to make students think through problems by giving hints and pointers in the right direction." (Anonymous evaluation)

"Good pace, easy to understand, good at interpreting questions & explaining." (Anonymous evaluation)

"Increadably helpful and interested in student's development." (Anonymous evaluation)

"Michael was very helpful, knowledgeable, and insightful ... he inspired and educated." (Anonymous evaluation)