Teaching Statement

Makayla Moster

My Experience

In middle and early high school, I was adamant that I did not want to become a teacher; however, I spent my senior year tutoring an elementary school student in English and Math; this experience provided me with my very first taste of teaching and sparked an interest in teaching later on in life. I then went on to college with aspirations of becoming a software developer upon graduation. My first experience helping others learn computer science concepts was during my junior year. My classmates asked me to organize a final exam study session for our Formal Languages and Computability class, which I ended up leading due to my understanding of the material. We commandeered a couple of whiteboards and an entire corner of the library, and I spent many hours answering questions and re-explaining the course materials to help my fellow classmates study for our final. This study session sparked an interest in helping others learn computer science, leading to a renewed interest in teaching as a career (specifically at the collegiate level) and my broad range of teaching experiences, from tutoring to being a graduate teaching assistant.

As an undergraduate student at the University of North Carolina Wilmington (UNCW), I served as a tutor for the computer science department, where I was one of three tutors who provided one-on-one help for the freshman and sophomore-level computer science courses. I was also an instructor for UNCW's Engineering Expectations summer and weekend camps for two years, where I helped students from elementary to high school learn basic programming and web development concepts in multiple week-long camps.

As a graduate student at Clemson University, I have been a graduate teaching assistant (GTA) for six years now and have sought out many opportunities to expand my teaching repertoire. I have been a GTA for two lab courses, an intro to C programming course (2 semesters) and a capstone software development course (5 semesters). I also helped develop content for a new graduate-level software engineering course, which I will be a GTA for during Fall 2024. I also organized and led a graduate seminar course that provided graduate students with faculty presentations on pertinent topics such as grant writing, academic job hunting, personal finances, and more. In addition, my research group hosts a free video game development summer camp for autistic high schoolers every summer, where I have developed content and led multiple days of instruction for four years. Lastly, I completed a Graduate Certificate in Engineering and Science Education (ESED), where I was prepared for an academic career by learning best practices for teaching STEM courses and conducting classroom-based research.

My Teaching Ideology

These experiences, especially my time as a summer camp instructor, GTA, and as a student in the ESED certificate program, have greatly influenced how I want to teach in the classroom. As a summer camp instructor, I taught a large variety of students, from kids with privileged backgrounds to those without and everything in between. As an ESED certificate student, I learned that not everyone learns in the same way and that there are proven teaching and assessment strategies to help ensure everyone is learning the content. As seen during my time as a GTA, everyone who walks through my classroom door has a different background and set of skills, so to accommodate this variety, I aspire to incorporate practical examples and active learning principles to keep students interested and engaged in the content.

I believe that computer science classrooms should be engaging for students through the use of concrete examples, hands-on programming experiences, and ample opportunities for discussion and digestion of concepts taught. Using concrete, practical examples in the classroom will help students make the connection between the concepts taught in the classroom and how those same concepts can be applied to a real-world situation within industry. Utilizing interdisciplinary examples of computer science concepts can also help maintain student interest and engagement with course content by showcasing how computer science concepts are important to other fields.

I plan to use active learning within my classroom to increase engagement and discussion surrounding concepts taught. It is important for students to come up with solutions to problems themselves; however, students can also benefit from discussing their solutions with others to identify the benefits and drawbacks of solutions similar to and different from their own. This discussion with their peers is important for both students' hard and soft skills. Students may learn how to implement better solutions from discussions with their peers, but they also are increasing their communication skills, which have been known to be lacking in college graduates [1, 2, 3]. Increasing students' communication and collaboration skills has been the focus of my dissertation research, and I would like to continue to help students develop these skills within the classroom by providing ample opportunities for students to present and discuss their work.

In summary, I believe that every student who walks through my classroom door, no matter where they come from, deserves a chance to learn computer science in an engaging and inclusive environment. Through my previous experiences in the classroom both as a student and an instructor, I plan to utilize active learning concepts and concrete examples to make my classroom an environment where students want to participate in learning, even if they may not be interested in the content.

References

[1] Vahid Garousi, Gorkem Giray, Eray Tuzun, Cagatay Catal, and Michael Felderer. 2019. Closing the gap between software engineering education and industrial needs. *IEEE Software* 37, 2 (2019), 68–77.

[2] Wouter Groeneveld, Joost Vennekens, and Kris Aerts. 2019. Software engineering education beyond the technical: A systematic literature review. *arXiv preprint arXiv:1910.09865* (2019).

[3] Wouter Groeneveld, Joost Vennekens, and Kris Aerts. 2021. Identifying Non-Technical Skill Gaps in Software Engineering Education: What Experts Expect But Students Don't Learn. *ACM Transactions on Computing Education (TOCE)* 22, 1 (2021), 1–21.