Teaching Statement

Dwayne "Chris" Brown, Jr.
North Carolina State University

Teaching Approach

My teaching approach is shaped by my teaching, research, and industry experiences. My education philosophy focuses on empowering students to pursue computer science and technology careers by providing real-world examples and creating diverse and inclusive learning environments.

Real-World Examples: I believe the best way to teach concepts and skills necessary to succeed for a computer science career is through experience. This includes creating and presenting lectures that include material based on real-world software engineering practices and research. I also envision tying in my own personal industry experiences at Bank of America, Blackbaud, and Red Hat into course lectures and materials. To make learning effective for students, I would emphasize project-based learning in my courses with original, collaborative, and hands-on coding assignments. For instance, one example of a way to provide real-world experience for students in a course is to have them to contribute to open source repositories for class assignments.

Diversity and Inclusion: I believe in a classroom that fosters an environment for everyone to learn and succeed no matter their experiences or backgrounds. As the instructor for a course, I believe it is important to be an active listener support the needs of students from various backgrounds in a classroom and one-on-one setting. This includes making course materials accessible for all students, accommodating different learning styles in lectures, making course requirements and expectations clear, diversifying project groups, providing opportunities for computer science majors who may not want to pursue primarily coding careers, and ensuring all students have the opportunity to participate and succeed in class activities no matter their race, gender, age, sexual orientation, programming experience, etc.

Experience

Teaching and Communication Certificate: I am currently enrolled in this graduate certificate program to provide training in teaching and communication to graduate students at North Carolina State University. The Teaching and Communication Certificate involves completing 100 hours of approved activities and creating a professional development portfolio. Through this program so far, I have taken useful courses that have impacted my teaching approach such as *Accessibility in the Classroom*, a course about creating content and materials accessible to all students in courses.

Teaching Assistant: My primary teaching experiences come from teaching assistantship positions as a Ph.D. student at North Carolina State University. I was able to TA for two undergraduate computer science courses: *Software Engineering* (CSC 326) in Fall 2015 and *Programming Concepts - Java* (CSC 216) in Spring 2016. For CSC 326, I was expected to teach two lab sections emphasizing concepts taught in lectures, act as the Scrum Master for teams during the final project, grade programming assignments and exams, attend lectures, answer student inquiries via email and Piazza, and hold office hours. For CSC 216, I was able to aid students during in-lecture activities, grade assignments and exams, and hold office hours. Both of these teaching assistant opportunities were valuable in helping me gain experience interacting with students, conveying introductory and more advanced programming concepts to them, and performing other necessary activities for computer science professors such as grading, office hours, etc.

Research: Through my research interests, I have gained further teaching and education experiences. As an undergraduate researcher at Duke University, my work primarily focused on integrating computational thinking and programming concepts into K-12 education. Through this work, I was able to teach basic programming concepts using Alice, a 3-D block-based programming language, to varied groups of students including three courses at Oak Grove Middle School in Winston-Salem, NC, a Duke FEMMES (Females Excelling More in Math, Engineering, and Science) event for middle school girls, Duke Alice Activity Day sessions for local sixth graders, and to K-12 teachers at two Adventures in Alice summer workshops for educators seeking to integrate programming into their courses. These opportunities allowed me to gain experience teaching basic computer science concepts to a wide variety of audiences.

For collegiate computer science education, I am currently designing a study to determine the impact of bots to nudge software engineering students to adopt better development behaviors for team projects. Additionally, for an independent study and graduate course project I developed a grading algorithm incorporating automated program repair concepts to improve student feedback and reduce instructor effort with the goal of decreasing attrition in introductory programming classes. Compared to a baseline approach of grading by test cases, we found that our algorithm increased student grades and took slightly longer to run on a benchmark suite. While these projects do not explicitly involve teaching, both explore automating and improving typical teaching activities performed by professors.

Tutoring: I have also gained teaching experience through various one-on-one tutoring opportunities. Over the summer I participated in The Coding School codeConnects program which focuses on increasing STEM participation for underrepresented groups. Through this 10-week mentorship and tutoring program, I was able to teach a high school student Python. Additionally, I have been able to tutor C++ to a Central Piedmont Community College and Java to an AP Computer Science at a local high school in Raleigh. These 1-on-1 tutoring experiences helped strengthen my knowledge in the basic concepts for these programming languages and helped my gain practice teaching computer science concepts to students.

Service: Finally, I have participated in many service and outreach opportunities focused on computer science education. With the NC State Minority Engineering Graduate Student Association (MEGSA), I was able to volunteer at an INTech Mini-Camp to help teach HTML and website design to African American middle school girls. Additionally, through the NC State Students and Technology in Academia, Research and Service (STARS) Computing Corps I tutored students in the *Introduction to Computing - Java* course in addition to volunteering at an outreach on campus to help teach Scratch, a 2D block-based programming language, to underserved local middle school students from the Math/Science Education Network Pre-College program.

Example Courses

Below is a sample list of courses that I would be excited to teach based on my interests and computer science curricula:

Software Engineering: An undergraduate or graduate level class educating students on the processes used by software engineers in industry. This would be a project-based course designed to help students build the knowledge and coding skills necessary to build and maintain software systems using realistic professional development team roles and requirements.

DevOps: A graduate level course that focuses on a specific area of software engineering to provide an overview of DevOps concepts in industry practices such as continuous integration, continuous deployment, build configuration, tools, and more. The course would involve completing a programming project to automate software development practices.

Open Source: Another software engineering undergraduate or graduate level course focusing on contributing to open source software (OSS). This class would go over topics such as the importance of OSS, licensing, open data, and civic tech, and would require students to contribute to real-world open source repositories throughout the semester.

Software Engineering Research Seminar: An undergraduate or graduate level course that explores current topics in software engineering and computer science research. This would be more of a discussion-based course that involves having students read and present on academic research publications, learn about general research practices, and design their own study to complete and write a publication-quality paper for the final project.

Introductory Programming Courses: A lower-level undergraduate class to coding basics to CS and non-CS major students with little or no programming experience. I feel that I can be most effective and it is more relevant to teach this type of course in Python or Java, but I am also willing to teach comparable classes in other programming languages.

Overall, I believe I can excel as an educator in software engineering and programming classes focused on helping undergraduate and graduate students gain knowledge and real-world experience developing and maintaining software. Using my teaching approach, prior experiences, and sample classes, I hope to create relevant and inclusive classroom environments to help students succeed in their future careers.