

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 integrating active learning, providing real-world examples, and creating diverse and inclusive learning environments.

Active Learning: I plan to implement active learning activities into classes for students to better understand and practice course materials. Rather than teaching lectures just for students to take in information, I plan to make learning effective for students by emphasizing project-based learning in courses I teach with original, collaborative, and hands-on programming assignments and class activities. I also believe it’s important to encourage students to participate in productive development practices that are often taught in computer science courses, including pair programming, requirements engineering, code reviews, and more.

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. Furthermore, I believe it is important to include relevant research findings into computer science courses to increase awareness of important research concepts to students, and plan to integrate my own work on improving developer behavior and productivity into courses I teach. Lastly, I plan to provide realistic projects and assignments for students to practice skills necessary to develop and maintain production-quality software. 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 plan to create classroom environments that foster equal opportunities for everyone to learn and succeed no matter their experience or background. As the instructor for a course, I believe it is important to be an active listener and support the needs of students from various backgrounds in classroom and one-on-one settings. This includes making course materials accessible for all students, accommodating different learning styles in lectures, making course requirements and expectations clear, diversifying project groups, and providing interdisciplinary materials. Additionally, I would aim to provide multifaceted active learning opportunities for computer science majors and non-majors who may not want to pursue predominately 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 North Carolina State University graduate certificate program to gain additional experience and training for effective communication skills. The Teaching and Communication Certificate requires completing 100 hours of approved activities and creating a professional development portfolio. Through this program, I have taken useful courses to improve my teaching approach such as *Accessibility in the Classroom*, a course providing information and techniques to create accessible content and materials for students in a class.

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 Software Engineering, 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 Java Programming Concepts, I was able to aid students during in-lecture activities, grade assignments and exams, and hold office hours. These teaching assistant opportunities were valuable in helping me gain experience interacting with students, communicating introductory and advanced concepts, and performing activities required by professors such as grading, holding 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 middle school courses, a FEMMES (Females Excelling More in Math, Engineering, and Science) event for middle school girls, local sixth graders, and K-12 teachers at summer workshops for educators seeking to integrate programming into their courses. These opportunities allowed me to gain experience teaching 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 mentorship and tutoring program, I was able to spend 10 weeks over the summer teaching

a high school student Python in weekly 2 hour sessions online. The curriculum involved teaching the basics of the Python programming language and helping the student complete various coding activities. Additionally, through the NC State Students and Technology in Academia, Research and Service (STARS) Computing Corps, I was able to tutor *Introduction to Computing - Java* (CSC 116) students. I have also tutored C++ to a community college student and Java to a high school AP Computer Science student. 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, I have volunteered at outreach events through STARS to help teach Scratch, a 2D block-based programming language, to underserved local middle school students. These volunteer opportunities have emphasized the importance of computer science education.

Example Courses

Based on my experiences and pedagogical interests, I believe I am equipped to teach undergraduate or graduate level *software engineering* courses for students to learn and practice processes necessary to build and maintain software systems. I am also able to lead *introductory programming* classes to teach basic coding concepts to undergraduates with little or no programming experience. While I feel I can be most effective teaching these types of courses in Python or Java, I am also capable of teaching comparable classes in other programming languages. Below is a sample list of additional courses that I would be excited to teach based on my interests, experiences, and current topics within the computer science field:

- **DevOps:** A graduate or advanced undergraduate level course that focuses on providing an overview of DevOps concepts. In the course, students would learn about development and operations industry practices and tools for continuous integration, deployment, project builds, and more. The course would involve learning about these processes as well as completing a final project to automate software development tasks.
- **Open Source Software:** 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. Students would be required 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 discussion-based course would involve students reading publications, learning research practices, and designing a study to complete for a publication-quality paper.

Overall, I believe I can excel as a computer science educator by helping undergraduate and graduate students gain real-world knowledge and skills through course lectures, assignments, and projects. With my teaching approach, prior experiences, and interests, I aim to create engaging and inclusive classroom environments with relevant materials for students to prepare them to succeed in the class and in their future careers as computer scientists.