Teaching Statement

Lina Battestilli, July 2023

I joined the NC State's Department of Computer Science as teaching track faculty in 2012. I have taught ~6000 undergraduate students in ~98 course sections and have managed ~300 teaching assistants (TAs). I have learned a lot on the job from my students, TAs and colleagues. The more experience I get, the more I get reflective about how I teach and design my courses. I have become more curious about different learning theories as I am always searching for research-based better ways to teach my courses. In this document, I highlight my teaching values and briefly describe the teaching methods that I have incorporated into my courses. I value course design based on equity, organization, clarity of expectations and learning outcomes.

Inclusive Practices and Mutual Respect: My goal is to create a welcoming, equitable and respectful learning environment for my students. Students may forget some of the details that they learned in my courses, but they will remember how my course made them feel. My syllabus starts with a clause on class community where every student is a valued member. I am committed to respecting the unique perspectives of different communities, cultures, races, ethnicities, and nationalities. I continually train my TAs to be respectful and to avoid unconscious bias or microaggressions when they are interacting with the students or grading their work. I set high standards and clear ground rules which highlight the positive behaviors expected from the TAs and students in my courses. Students and TAs have reached out to me when they felt disrespected or not supported and I have followed on discussing and working through these situations.

Compassion, Flexibility and Equity: I make efforts to teach my classes in a student centered way and to provide my students with compassion, flexibility and equity. I tell my students on the first day of class that I will be "kind and firm" and that their well-being is very important to me. I want my students to feel comfortable to ask questions and get help when they need it. I monitor students' attendance and along with my TAs reach out to students that are absent or not turning assignments. Students can request extensions (before or after a deadline) and I honor their needs for extra time but also keep them working at the pace of the course. Not all students have equal resources outside of the classroom (some work or take care of their families, etc). For equity, I don't penalize the students' grades if they ask for an extension. I assume that my students are doing their best and often my students thank me for reaching out and for this provided flexibility. In 2021, I started teaching part of my courses in HyFlex (Hybrid-Flexible) where the students have a choice to either come in-person, join via a synchronous zoom session or asynchronously view a recorded video of the class. My students have been successful in learning in the HyFlex classroom and have expressed gratitude for this extra flexibility.

Course Orientation & Developing Self Regulation Skills: Modern computer courses have high cognitive demands and use a variety of tools for programming and learning. Courses should be clearly organized and students should be oriented as to how to be successful in a course. The first week of a course should be guidance through the learning management system, the learning resources, the expectations/deadlines and how to ask for help. I often advertise the help opportunities in my course (online forum, office hours, small-group tutoring) and *invite* the students to join. I have clear deadlines for my courses that don't change weekly. Multiple times through the semester, my students are asked to reflect on their course performance and their study habits. This is a way to develop their self-regulation skills and to identify successful learning strategies and study habits that may need to be improved.

Provide Instrumental Help instead of Expedient Help: My research over the past few years has made me realize that students often ask for expedient help, i.e., help just to get them to complete an assignment. This is especially problematic in programming courses where the topics build on each other. If a student moves expediently through the completion of assignments but doesn't actually learn the concepts then they will struggle with the next topics of the course. So I have restructured the help in my courses so that the teaching staff can also provide instrumental help, where students are given the amount of help they need to complete a task on their own. In my programming courses: 1) I teach the students to use the debugger from day one, 2) train my TAs to ask questions and guide the students instead of just providing the answers and 3) provide lots of variability in learning resources (autograded multiple choice practice questions, atograded small programming questions, videos on each concept with worked out examples, weekly at-a-glance reference guides,etc.)

Support Students with Variable Preparation: I believe that there are no bad students and that everyone can learn to their full potential if provided with the right type of support. Therefore, I often reflect not only on the average performance of the whole class but I also look at the performance of individual students. To do this in my large introduction to programming course, I have restructured the TA team so that we have Computer Science (CS) Mentors. The role of the CS Mentors is to reach out directly to students who received lower grades on the first few major assignments, invite them to small-groups tutoring and lead study sessions prior to tests. I also create individualized Catch-Up Plans for lower-performing students which involves: 1) requires them to attend a

certain number of help sessions with the CS Mentors/TAs and 2) allows them to retake tests so they can show that they have mastered the material. Students learn at different paces and courses should be designed to accommodate for these differences!

Broadening Participation in STEM: I am a strong believer in broadening participation in computing and most broadly in engineering. Technology is part of every part of our lives and diverse teams need to be involved in creating this technology. This is also an issue of equity as computing jobs are secure, well-paid and in high demand. So in addition to just teaching the technical concepts in my courses, I often focus on wider societal issues, ethics and the impact of software engineering decisions. I encourage my students to persevere in getting a computing degree, to continue to work hard and not to doubt their innate abilities. For more than 10 years, I have been the faculty adviser for the Women in Computer Science (ACM-W) student group and I am highly involved in managing this group, fostering their community, providing direction on their events, and connecting them with industry sponsors for career opportunities. Since 2018, I have also done research on a lightweight intervention that provides my students with additional encouraging feedback when they receive their grades on major assignments. This intervention has increased the students intentions to persist in computing, their self-assessment of computing abilities and will hopefully impact their career choice. I am also a Cultural Competence in Computing (3C) Fellow where I learned about social science topics (such as identity, intersectionality, racism, bias, and discrimination) and how these topics impact computing environments (as well as the developed technologies).

Foster Critical and Creative Thinking: As students graduate, it is very important that they are able to apply critical and creative thinking when solving new problems, evaluating the work of others and reflecting on their own thinking. This remains true if they get a job in industry or if they continue their academic studies. Students should be introduced to the breadth of topics in their major but also develop the necessary skills to be able to dive deeper and explore specific topics in detail. This will ensure that they are successful, innovative and well prepared to tackle new problems as they begin their careers. Typical assignments in Computer Science courses require students to be critical thinkers and show precision, accuracy and logic. My goal, however, is for the students to also be creative, original, adaptable and flexible in their thoughts. In my classes I incorporate open-ended projects, reflections, peer reviews and collaboration. I use Bloom's Taxonomy to create assignments and assessments so that my students can practice higher order thinking. Factual knowledge is definitely not enough and students should be asked to evaluate, analyze and even generate new ideas.

Active Learning leads to Engaged Learners: Students have different learning styles, personalities and academic backgrounds. Some students thrive on lectures and theory but most students learn much better with lots of examples, visual materials and hands-on activities. Some students learn well in groups while others prefer to work alone. Some students have had experience in the subject while others are completely new to the topic. Students also have different motivations for taking courses, some want to get proficient in the topic while others simply focus on grades and passing the course. There are two ways to look at this problem. A very superficial way is to categorize students as "bad" and "good" students and thus place all the responsibility on the students. This, however, does not solve the problem of disengaged students. A much better solution is to focus on how the teaching is done in the course and what the students are doing in the course and to evaluate their actual learning. I believe that no class should be taught by just lecturing as research tells us that people can concentrate on listening only for a few minutes before their minds wander off to other things. Active learning engages the students with the content, helps them stay focused so that they are able to connect the new material to things they have previously learned.

Project-Based Learning -- Relevant to the Learner: The best way of learning is by doing. According to Edgar Dale, people generally remember only 10% of what they read, 50% of what they see and hear but 90% of what they do. In my classes I always incorporate practical hands-on projects. I look for interesting project topics that have larger societal impact and/or are relevant to the learner. I have created projects that use more serious datasets such as election data, health data, poverty data, hurricanes data, social security baby names, high performance cars, twitter sentiments, financial stocks, truss/frames engineering design, etc. However, I also want my students to be able to relate to the topics and have fun while learning so I have also created projects around games such as Battleship, Mastermind, Connect Four, Adventure Games, etc. I want my students to see the impact of Computer Science but also be able to personally relate to the topics and have fun while learning.

Future of Education -- Digital Learners: Technology is changing our society in so many ways and undoubtedly, many of these changes are coming to higher education as well. Using technology, we can replace traditional in-class lecturing courses with new types of courses such as flipped, fully-online, hybrid-flexible, using AI, etc. This transition is necessary because it allows the instructor to spend their time with the students on discussion and active learning activities. I envision courses being divided into two phases, where the 1st phase is fully online and self-paced and the 2nd phase is project-based and instructor mentored. The new role of the instructor will be to create/manage the online learning, mentor the students and to facilitate discussions and active problem solving sessions. I am an early adopter of educational technology because I believe that it brings efficiency, innovation and organization to my courses. I also think that this is helping my students develop the necessary 21-century skills in an increasingly digital and connected age. I always have an active online forum, I use learning management systems, my tests/assignments are online and paperless as much as possible. I also use learning tools, videos, interactive online textbooks, etc.

Innovative Pedagogy that I have used my courses:

- **HyFlex Instruction:** Since fall 2021, students can attend my courses either in-person, synchronously over zoom or watch asynchronous classroom recordings. Students appreciate this flexibility and on average I have observed that about 40% chose in-person, 40% chose zoom and the rest either don't attend or watch asynchronous recordings. I am currently doing research to better understand the impact of HyFlex on student learning.
- **Peer-Instruction:** Peer Instruction is a specific pedagogical practice defined by Eric Mazur of the Department of Physics at Harvard University in the early 2000s. It centers around a method for asking questions during class. The students have to follow a three step process to answer: 1) individually respond, 2) discuss the question in small groups, and 3) respond again based on their new understanding. [peerinstruction4cs.com]
- **Flipped Classrooms:** where initially students are introduced to new topics outside of the classroom. In-class time is used to explore the topics at greater depth and to create an active learning environment. I received a DELTA Critical Path Course Redesign Grant in 2018-2020 and have fully flipped CSC113 (a large course on introductory programming, ~280 students). Initial evaluation (via surveys), shows that the flip is indeed effective.
- **Growth Mindset vs. Fixed Mindset Theory:** In this theory, a person with a Growth Mindset believes that they can get smarter, their intelligence is not fixed and that with enough effort they can learn. Students with a Growth Mindset tend to put in more effort, be more persistent, and in turn, have better grades. I led the effort to add Growth Mindset video modules into two computer science courses: 5 videos for CSC116(2015) and 3 videos for CSC113(2019)
- **"Micro-classes" in a large class:** I have developed successful techniques to teach large classes (~280 students) and manage large groups of TAs (~15 each semester). I divide the students into "micro-classes", where students develop a sense of community within a large class.
- **Self-Regulation Skills:** refers to the self-directive process through which learners transform their mental abilities into task related skills. I have adopted techniques to help my students self-reflect on their study habits and learning goals.
- **Grades with Context:** STEM courses tend to have lower grades than other courses which makes students doubt their STEM abilities. Instead of just numeric grades, I give additional feedback to help my students understand their performance.
- **Open Ended Assignments, Code Reviews and Reflections:** From 2015 to 2017, I participated in NC State's TH!NK program. I created assignments and activities for my courses to improve the students' critical and creative thinking.
- **Guest Speakers:** It is a tradition that in all of my courses, I invite a guest speaker. Typically, I invite folks from industry that have a successful career using knowledge related to the course. My students always comment on how much they like these guest lectures because it helps them build important connections between what they are learning and its applications in the real world.
- **Organized Study Hours:** In 2017-2018, I organized study hours led by students currently taking the course (selected via application/resume 3rd week of the semester) as an additional resource to TA office hours. Study Hour leaders organize and hold weekly 1-hour sessions in the NC State libraries available to all students in the course.