SARAH MORRISON-SMITH | TEACHING STATEMENT

Teaching Philosophy

My background in human-centered computing has shaped my goal as an educator to foster an environment that guides students towards interdisciplinary thinking. I have had a range of teaching experiences, from acting as the instructor for undergraduate courses at Hamilton College, Barnard College, and the University of Florida to preparing community-oriented workshops. I have also had the pleasure of mentoring a number of undergraduate students over the course of my career. I also enjoy connecting with students outside the classroom through extracurriculars that expand the boundaries of their interest in computer science. My teaching philosophy focuses on three key strategies in the classroom and mentorship setting: 1) active learning, 2) active listening, and 3) providing practical examples.

Active Learning: I have found that when introducing new material in class it is particularly important to actively accommodate different ways of learning among students. As a result, I strive to present and assess the information that I teach in multiple ways. For example, when I taught students how to perform hierarchical task analysis in COMS BC3364: Introduction to Contextual Design for Technology, after orally presenting the concept I visually show two examples of how the method can be applied to everyday tasks, and then provide students with an opportunity to independently practice the technique, compare answers in small groups, and then finally go over the content together as a class. When I teach programming-heavy courses, I make sure to present information in multiple ways, such as live coding in class, providing demo code to be run and analyzed outside of the classroom, and written documentation for reference. I also deliberately incorporate different activities so that class participation can happen in different ways. I recognize that the diversity in student cultures or personalities may affect participation and learning preferences, such as engaging in critical class discussion. As such, I incorporate a variety of activities in classrooms to address this range of cultural norms and participation preferences.

Active Listening: For me, being an effective educator means not only communicating new information to students but also listening to my students and tailoring my classroom to their needs. It is common for my students to have different backgrounds and levels of familiarity with the material that I present. To best support this diversity in experience, I have found that being accessible for one-on-one instruction with students before or after class can effectively allow students to communicate their concerns about grasping material. I have also employed reflective methods to help me discern which aspects of my instruction is confusing or effective to help me revise how I teach and assess knowledge. For example, I have recently moved to a weekly reflection model where students individually or in small groups identify what aspects of the week's lectures they understood well, what they have questions about, and what has inspired them to seek further knowledge. I use the responses to answer questions that students have otherwise been too shy to ask in front of the class and refine my materials for the next time I teach the course. I have also successfully employed mid-semester reflections to re-assess teaching methods during the COVID-19 pandemic. To reach as many students as possible while teaching virtually during the pandemic, I tested a variety of methods to foster discussion and facilitate asynchronous learning. Student responses to the mid-semester reflection revealed which techniques were successful in the classroom so that I could tailor the course to suit the students' unique virtual learning experience.

As a faculty member, I feel that it is my responsibility to help all students understand the concepts to the best of their ability, which requires active listening on my part. I also strive to meet students where they are with regards to their daily lives, passions, and diverse backgrounds and abilities. For example, in the Introduction to Contextual Design course, I noticed that students responded thoughtfully to concepts that align with their everyday experiences. As a result, I designed examples during lecture grounded in students' experiences with technology when explaining design concepts. As a researcher, this means tailoring my mentorship style to fit the needs of the student, such as varying the level of structure or emphasizing development of research vs. programming skills.

Providing Practical Examples: I believe that students learn best when they understand how the skills they develop in the classroom are applied in practice. For computer science students, this involves many concrete examples, hands-on coding experiences, and collaborative projects that utilize the skills students will require for post-graduation success. For example, in my Developing Accessible User Interfaces course, I have found that the most effective learning regarding accessible design includes examples of accessible and inaccessible interfaces in the wild. This class also features a semester-long project where students collaborate in a team to develop a new assistive technology from idea inception to full implementation. This gives students hands-on practical experience applying the concepts learned in the classroom to a real product that can be further developed. I also invite alumnae to share their experiences with the class regarding the user-driven development process in the tech industry. This grounds the material presented to students. In my Developing Accessible User Interfaces class, students apply the principles they learn in class regarding accessibility to develop Chrome extensions that provide equal access to the web to users who otherwise are barred by disability. Several students have contacted me post-graduation to tell me that the topics covered in this course immediately benefited them when starting their first tech job.

Teaching Experience

Lecturing: As a PhD student at the University of Florida, I taught *COP3502: Introduction to Programming Fundamentals I*, which is the first course in a two-semester introductory sequence for students with no prior programming experience. This course explored major concepts of computer science and the process of computer programming, including object-oriented programming, procedural and data abstraction, and program modularity.

Later, at Barnard College, I developed and taught two undergraduate-level electives courses, *COMS BC3364: Introduction to Contextual Design for Technology* and *COMS BC3162 Developing Accessible User Interfaces*. Intro to Contextual Design introduced students to the methods and tools used in Contextual Inquiry (CI), User Experience (UX), and User Interface (UI) design and was designed to be an interdisciplinary course suitable for students with a broad range of majors, not just computer scientists. In contrast, Developing Accessible User Interfaces is a more technical elective targeted towards upper class undergraduate students and focuses on access technology and the development of accessible systems by teaching students to build and evaluate various access technologies. Both Barnard classes have also been adapted for a summer seminar targeted towards high school students as part of Barnard's Pre-College-Programs. These seminars span two to three weeks and act as an appetizer for the course where students complete a portion of the college-level coursework.

At Hamilton College, I've taught *CPSCI 101 CS For All*, the introductory course to programming open to all majors that features creative projects in a number of areas including: natural language processing, image manipulation, and audio synthesis. During the fall of my first year at Hamilton, I taught *CPSCI 220 Principles of Programming Languages*, a mid-level course that introduces students to five programming languages (Bash, Smalltalk, Java, Scheme, and Prolog) as well as the fundamentals of programming language design. I've also taught *CPSCI 360 Developing Accessible User Interfaces* as a refined version of *COMS BC3162 Developing Accessible User Interfaces*. Finally, this semester I am pioneering *CPSCI 410 Senior Seminar in HCI for Social Good*, a course that pairs small groups of senior students with local non-profits to create bespoke software.

Service: At Barnard, I co-organized BEARS: Better, Enhance, and Advance Research Series. This program has featured monthly talks from women faculty in computer science, regular workshops on research skills, and a mentorship program connecting undergraduate students at Barnard to women Ph.D. students at Columbia. As a part of this program, I have conducted two workshops on topics related to computer science research careers. I also provided materials for Barnard's adapted version of Kode with Klossy, a program developed to create learning experiences and opportunities for young women that increase their confidence and inspire them to pursue their passions in a technology driven world. This course was taught to high school students in the greater New York City area in the summer of 2022. To do this this, I developed supplementary curriculum that introduces Kode with Klossy students to the human-centered-design aspect to developing technology. This includes modified lectures from my Barnard classes alongside new content developed especially for Kode with Klossy. Finally, I have been engaged as a mentor at Columbia University's Design at Columbia program. As a mentor, I have guided student projects and participated in user studies. I have also given a workshop on how to quickly and effectively improve the accessibility of new technology. I further expanded and developed this workshop which was given during Columbia University's 2021 DivHacks Hackathon.

Mentoring: Throughout my time as a graduate student and faculty fellow, I have had the pleasure of mentoring twenty-seven undergraduate research assistants, all from demographics traditionally underrepresented in computer science. Four of these students have gone on to pursue a graduate degree in computer science. My proudest moments have been watching my students transition from following my lead to taking initiative and charting their own direction. Mentorship has given me the opportunity to teach both general topics in computer science (e.g., intermediate programming, using source control) and research topics (e.g., experiment design, paper writing), as well as help navigate the world of academia in a one-on-one context. Several of these projects have resulted in an ACM publications.

Future Teaching Plans

I have found teaching to be both a pleasurable and rewarding experience for me. I consider the role of an instructor to be important and as such, I am unceasingly looking for ways to improve as an educator. My teaching and research experiences make me particularly well-suited to teaching classes and courses in human-computer interaction, qualitative and quantitative research methods, and user interface design and construction as well as traditional Computer Science classes and courses. Below is a list of courses that may complement the current curriculum along with existing classes that I am excited to teach:

- **Introduction to Computer Science:** The first course in computer science is often an introduction to algorithmic problem solving in Python.
- **Social and Ethical Issues in Computer Science:** Social and ethical issues as they relate to computing, including privacy, risks, computer abuse, commerce, professionalism, free speech, intellectual property, social justice, and current issues.
- **Developing Accessible User Interfaces:** Introduction to access technology and the development of accessible systems.
- **Introduction to Contextual Design for Technology**: Introduction to the methods and tools used in Contextual Inquiry (CI), User Experience (UX), and User Interface (UI) design.
- **App Development for Android:** A primer on how to develop and deploy mobile applications on Android. Students will learn and utilize java to develop an application through a semester-long project. Additional topics may include data management, animations, and some recent developments in the Android operating systems.