Teaching Statement

Yasra Chandio
yasrachandio.github.io

My teaching philosophy focuses on transforming conceptual understanding into practical, tangible applications, reflecting my research approach. In my work, I apply theoretical frameworks from psychology to understand subjective human behavior and map these insights into system inputs that drive adaptive technologies. Similarly, in teaching, I guide students in connecting high-level concepts to real-world applications. **I am motivated by the power of scientific and mathematical frameworks to model and address real-world problems and aim to inspire students to see how abstract theories can lead to concrete solutions.**

Teaching, Mentorship and Outreach Experience

Connecting Theory and Practice Across Courses: I worked with students from varied backgrounds and learning stages, from introductory undergraduate-level courses to advanced, graduate-level project-based courses. Regardless of level, my teaching emphasizes the progression from theoretical knowledge to hands-on applications.

As a teaching assistant (TA) for four years for the *Introduction to Embedded Systems*, a core course for approximately 230 sophomores, I taught fundamental C programming for Linux-based hardware platforms such as BeagleBone Black and Raspberry Pi. Many students encountered programming on these boards for the first time, and they come in with diverse skill levels. Over the years, I have developed an approach that focuses on breaking down complex programming tasks into a sequence of clear, logical steps, beginning with high-level concepts and gradually moving to specific, hands-on tasks. I have found that structuring labs to scaffold these principles enables students to progress from foundational understanding to seeing the real-world effects of their work, even if that effect is as simple as getting a temperature sensor reading by pressing a physical button. **I adapted my teaching methods depending on each student's needs. Some students benefited from a high-level sketch of the task, where I used the high-level objectives and real-world relevance of the task, helping students see why each step is meaningful.** For example, I might first introduce memory allocation in the hardware and then show how low-level processes, like memory management, directly impact hardware performance. For those requiring additional support, I divided the lab into smaller, manageable chunks and guided them step-by-step through each lab section, from setting up the code to interpreting outputs and troubleshooting. This process often included live debugging, where I modeled troubleshooting techniques, turning challenges into real-time learning moments. Through this progressive structure, students followed a clear path from theory to application at their own pace, making complex concepts more accessible, building confidence, and maintaining curiosity.

I further adapted my teaching approach to meet the distinct requirements of other advanced courses, combining structure with the flexibility to explore complex ideas in *System Programming* and *Programming Languages* courses, which served 120 juniors and 60 graduate students, respectively. I guided students through in-depth topics that demanded a strong understanding of low-level programming and system mechanics. These courses required students to develop a deep technical understanding in a structured way, and I emphasized how specific programming choices impact overall system functionality. Here, I focused on developing students' critical thinking and technical depth, preparing them with a solid foundation in advanced systems design.

In the *Networked Embedded Systems* course, where semester-long, collaborative projects are more open-ended, I guided undergrad seniors and graduate students through developing their own solutions from theoretical foundations. I encouraged students to structure their projects independently, set collaborative goals, and manage workflows while I provided targeted feedback at each project stage. In the short term, this allowed students to explore complex ideas in a supportive framework, balancing guidance with the freedom to pursue their own solutions. In the long term, my approach cultivated learning independent problem-solving skills and teamwork, equipping students with the adaptability to handle open-ended, real-world challenges.

Mentorship and Outreach: **I have mentored dozens of senior undergraduate, graduate, and REU students as part of course projects and independent study.** This mentorship involves guiding them through complex project lifecycles, from defining requirements to final implementation. I mentored a group of students who applied concepts from a project on systematically measuring multi-user collaboration to develop a secure framework for managing shared data on digital platforms. Their work became their senior design project and was recognized as the best project in the graduating ECE class. Mentoring students through such projects has shown me how hands-on guidance can empower them to take ownership of their skills and push boundaries, translating their learning into meaningful, innovative solutions.

My commitment to teaching extends beyond traditional classrooms. Through K-12 outreach, I adapted my approach to meet younger students' needs; I designed a 3-week 3-course to introduce high school seniors to programming concepts and guided them in building simple systems using platforms like Raspberry Pi. I also conduct interactive "show and tell" sessions for elementary students, demonstrating the potential of technology in an accessible way. I have organized and participated in panels and events connecting students, researchers, and industry professionals, focusing on careers in STEM and graduate education to support students in navigating academic pathways and exploring career opportunities. All these experiences have reinforced my belief in the importance of nurturing interest and curiosity at all learning stages, which I plan to continue throughout my academic career.

Apart from teaching and mentoring, I actively participate in academic services. **As an editor for CRA-Undergraduate research**

highlights, I showcased exceptional undergraduate research, highlighting non-traditional paths to graduate education and research careers[1]**.** I have contributed as a technical program committee (TPC) chair for Sensors S&P@Sensys; TPC member for DATA@BuildSys and ACM S3@MobiCom, Grace Hopper Celebration AR/VR/MR tracks; and reviewed for top venues, including IEEE VR, CHI, ICRA, and ICLR.

Looking Ahead

Inspired by the structure I experienced as a student in my graduate-level Operating Systems course during my Masters, where the instructor maintained a balance between content delivery, discussions, and questions, I aim to apply a similar balance in my own teaching and mentorship, making space for exploration within a structured framework. Moving forward, **I aim to bring this integrated teaching philosophy to each course that I teach and the students I will mentor, using a structured yet flexible approach to meet the diverse needs of my students.** For instance, an observer in my class would see a mix of structured lessons, hands-on labs, and real-time feedback. My goal is to cultivate a learning environment where students understand the theoretical knowledge of their field and feel equipped to apply their knowledge in innovative and impactful ways.

Core Learning Goals: In engineering and computing, students often learn concepts in isolation, with a limited understanding of how they function within a larger system. My teaching approach will begin with high-level objectives, helping students understand how individual components contribute to overall functionality. My top-down teaching approach will clarify the relevance of each concept, linking foundational knowledge to real-world applications. Across all courses, my goals are to build a strong grasp of core principles, encourage structured problem-solving, and instill confidence in bridging theory and practice. Whether students are working in low-level programming, writing firmware for hardware, exploring system architecture, building real-world system prototypes, or creating tools to analyze user interactions with systems, **I aim for my students to connect skills across contexts, preparing them to address real-world challenges with confidence, adaptability, and purpose.**

Inclusive and Adaptive Teaching: I believe an inclusive classroom is one where all students, regardless of background or experience, feel supported and confident in their learning. Creating this environment will require cultivating openness, where students feel comfortable asking questions and exploring ideas without judgment. In my teaching, I will provide an adaptable framework that meets students at their respective skill levels, allowing each individual to engage meaningfully and progress. My approach will consider the diverse backgrounds students bring to the classroom. For those with a strong technical foundation, I will introduce optional advanced tasks and extensions to standard assignments, challenging them to apply concepts in more complex scenarios without disrupting the primary course flow. For students with less exposure or fewer resources, I will focus on creating accessible pathways to foundational skills, breaking down complex concepts into clear, incremental steps. I will use office hours and teaching support to provide additional guidance, allowing students to build confidence in core skills at their own pace. By addressing individual questions in a supportive, non-judgmental setting, I will ensure the students receive the help they need without feeling self-conscious or left behind. Knowing the time and resource constraints, I will actively promote peer-to-peer learning in my classroom, encouraging students to draw on each other's strengths to provide additional support. For instance, in the Networked Embedded Systems class, I asked students to regularly share project updates, which improved their technical communication skills and created an environment where they could learn from one another. This practice has proven effective in building a supportive, collaborative space where students benefit from diverse perspectives and complementary skill sets. These experiences reflect my commitment to adapting my teaching to meet diverse student needs and skill levels, using structured guidance and practical applications to help students with hands-on experiences.

Developing and teaching courses **With my intersectional research and teaching background in systems, human-computer interaction (HCI), and machine learning (ML), an undergraduate degree in software engineering, and work experience as an embedded system research engineer, I am comfortable teaching a wide range of courses.**

These include traditional core courses like Introduction to Computer Systems, Programming Fundamentals, Systems Programming, Introduction to Embedded Systems, Human-Computer Interaction, Networked and Real-Time Embedded Systems, Multimedia Systems, Wireless Networking and Mobile Computing, Sensing and Pervasive Computing, Internet of Things, Software Engineering, Object-Oriented Programming, Software Architecture and Design, and Software Reliability and Testing.

I am also well-prepared to design and teach specialized courses such as Human-Centric System Design, Immersive Computing Systems, Interactive Systems, Mobile and Cyber-physical Systems, Modeling and Learning from Real-Time Human Feedback, Wearable and Ubiquitous Systems, Topics in Safety and Privacy Impact of Sensing and Immersive Systems, Theory and Practice of Prototyping Human Behavior with Sensing Systems, Sustainable Systems with Sensors.

[1] https://sparc.cra.org/stories/