I have had the opportunity to work with students from all levels with the courses that I develop, ranging from professional master’s students, pre-engineering undergraduates, and international students through MOOCs. I am a strong believer that the best way to learn engineering is through a practicum-based curriculum, a vision that I have contributed to and will continue to contribute to through course and curriculum development. My teaching philosophy is grounded in five main pedagogies: Build it/ Break it/ Own it; Leave room for problem identification; Leave room for creativity; Teaching is learning; and Learn how to learn.

** Build it / Break it / Own it –** Project-based classes have the power to draw students out of problem sets and into that surge of excitement derived from seeing their code run for the first time on a smartphone. In the process of building, students will be challenged with the process of breaking. Through the iterative process of building and breaking, a student truly begins to find ownership. Ownership of an idea is core to someone finding enjoyment in what they do. If you don’t own it, you won’t bother to build and break it.

**Leave room for problem identification –** One of the biggest weaknesses in many students is the ability to understand what it is that they are actually trying to solve. I believe this is a core issue with students being fed specifications to achieve without determining the specs for themselves. Identifying the problem that needs to be solved is a core competency that must be developed before a student can critically think about how to solve the problem. Using my philosophy, students will design their own experiments and not just follow a protocol. Problem sets can be divided into specification design and then calculate. This skill is quintessential and can be fostered not only with project-based teaching but through well-designed problem sets.

**Leave room for creativity –** An environment that fosters creativity leaves room for each student to generate their own ideas on how to look at the problem, what are potential solutions, and how they approach executing the project. When creativity thrives, we can truly leverage the diverse background of each individual in the classroom, research project, and even outside the classroom. This hinges on creating a collaborative and welcoming work ethic in all students. Part of being creative is to be exposed to diverse ways of thinking. My research in global health technologies has forced me to challenge my own assumptions as I try to understand different cultures and the challenges faced by those in different socio-economic backgrounds. In my teaching, I want to challenge my students and push them to go beyond their current worldview by introducing them to the diversity of ways people think, lives of people from different backgrounds, and instill students with the desire to challenge their own assumptions and understand the needs of others.

**Teaching is learning –** This is one that is as much for me as it is for the students. By teaching, you are forced to learn the material more than just in passing, you must know it well enough to make someone else understand. It also forces reassessment of your understanding and assumptions. In my class, students often assume the role of teachers either through discussion driven by students or through pairing of students with different skill sets to foster teaching between students. I also learn through teaching the courses I develop. I was the founding instructor for the Biosignals Processing Laboratory at Harvey Mudd College. As the semester progressed, I continued to adjust and iterate on the material for the course based on student feedback and observing the cohort’s learning progress.

**Learn how to learn –** Classes I design motivate students to explore the world of embedded systems, signal processing, and machine learning by directly using these concepts in projects that solve real world problems. However, the skills they learn through these projects are only secondary to the transferable skill of knowing how to learn. The main objective in my courses is to open up the door for students to learn how they can learn more about the topic because they feel confident that they have been trained to pull apart any concept, tutorial, or problem statement in order to generate their own insights.

**Teaching and Course Development Experience**

My main passion for teaching is designing project-based courses that help students understand the entry point to current industry and research topics. I had been a lead course developer / instructor for a laboratory course in biosignals processing at Harvey Mudd College during my time as an undergraduate student and a co-lead course developer for a Ubiquitous Computing Professional Masters Program course at the University of Washington.

The **Biosignals Processing Lab** is a course that I developed during my Junior to Senior year summer at Harvey Mudd College, and subsequently taught in Fall 2011 and Fall 2012. The course was designed for pre-engineering students with little to no knowledge about signal processing, let alone bio-signal processing. The purpose of the course was to promote general experimental design while gaining competency in quickly learning to use the tools available. I was responsible for deciding what the students would learn and developing the corresponding content, such as pre-lab material to teach core MATLAB skills or conceptual readings on physiology, biophysics, and basics of signal processing. Through the course of a summer I came up with 4 experiments involving the eye, the skeletal muscle, the heart, and the brain, each designed to teach the student a different concept of signal processing. In each of the experiments, the students were introduced to the physiological basis for each signal in their pre-labs and during their in-lab sessions. For example, students were presented with a series of questions such as "how do the placement of electrodes affect signal quality" and "how does the frequency content of the EMG change with respect to fatigue." The students were not given the experimental protocol, but instead were instructed to draft their own in order to answer these questions.
One of my favorite labs I designed involved the whole class going to the gym where each team was tasked with measuring the electrocardiogram (ECG) of a person when they were flipped up-side-down. The goal of this experiment is for students to observe how the ECG measurement is actually an electrical potential measurement that corresponds to mechanical change to the body. When the human body is flipped up-side-down, gravity causes the heart to rotate out of the natural resting angle. This causes the ECG measurement to also change in amplitude measured by each projected axis. It is a weird phenomenon and forces you to make the connection that physical systems are often not neatly separated into mechanical and electrical components.

In our course review, students noted that this was an eye-opening experiment that not only challenged their thinking, but also started making them think more deeply about every signal they measure.

During my graduate career, I again had an opportunity to be a co-course developer, this time for the Ubiquitous Computing course of the UW PMP course offered in Spring 2018. The course is centered on giving professional masters students already working in industry at companies such as Microsoft, Boeing, and Amazon the opportunity to become acquainted with hot topics in technology. As such, each time this course is offered, the content is revamped to reflect the newest and coolest techniques and gadgets. This year, we tailored the course to introduce students to Android programming, machine learning, signal processing, embedded device programming, Rapid 3D prototyping, AR and VR interactions, and Edge Computing. The course centered around a project that builds upon itself through the quarter. The final creation was an embedded system, encapsulated in a 3D printed housing, that dynamically interacts with an Android program that performs on-board machine learning and signal processing.

Beyond teaching in and designing for the classroom, I am very passionate about mentorship of students. In the course of my PhD, I have had my own master/undergraduate/highschool student sub-group, where I directly mentor and advise the students from things like their thesis project, classes to take, and incorporate students into my own research projects. I now even have a student who is leading his own research project in developing a phone-based osteoporosis screening app. I have graduated students who have gone on to join the PhD program at MIT CSAIL, full-time at Apple Inc., and won awards like the Mary Gates Research Scholarship, the Washington State Research Foundation Fellowship, and placed 3rd Overall in the International Science Fair. Through our lab’s summer high school student program, I have also worked with underrepresented students from a variety of backgrounds in the Greater Seattle Region. My mentorship style is very personal. I share my personal story, my own struggles in my career, and what paths I have taken and why. As a research adviser and course instructor, I intend to continue bringing my personal story to the table for others to learn from.

Beyond learning about how to teach courses at all levels and designing new courses to promote more “Learn how to learn” curriculum, I am generally interested in engaging in programs that are meant to increase computer literacy in the public. At the University of Washington, I have participated in many outreach programs focusing on showing what computers can do beyond playing videos to children as young as five years old. One aspect of outreach I would like to contribute to is engaging with GED curriculums and policy makers to hold seminars that promote more appreciation and understanding in computational thinking.

**Future Teaching Plans**

One of the courses that I believe is missing in most information and computing related curriculum is a **Mobile Computing Debugging course**. A course like this would blend topics from embedded programming and mobile application programming. One of the biggest leaps for a student beginning to program is transitioning from nicely contained environments to an application platform. However, it is often the case that when students program their first application on a phone or Arduino, they start to see the point beyond the theory. A large hurdle is that fundamental programming classes only teach students to program while courses dedicated to Hardware Software Interfaces or Mobile Programming only focus on developing programs to get through canned projects. Once you are out of the confines of the class, one of the biggest challenges is figuring out what is going wrong with your program. For embedded programming, the issue could be entirely not in your program at all! The hardware version you are using might not have the module you are trying to address. Your circuit might be connected in the wrong way. How do you know if you have a hardware or software issue? For mobile programming, error messages are often hard to unpack and debugging strategies can be utilized to more systematically understand the issues. Online resources are abundant, but how do you sift through all that content to actually know which is the right one without just copy and pasting code until it works? A course dedicated to teaching students debugging techniques instead of focusing on programming can help bridge this gap between theory and practice, ultimately improving their ability to navigate new opportunities at internships, graduate research programs, and even in their ability to help others as they learn to program.

Through my graduate and undergraduate career, I have had the opportunity to work with students through mentorship and teaching, and I am excited to continue this as part of the faculty. Although my main experience has been in designing and teaching my own course, I also have the interest and training to take on courses already being offered. I am confident teaching both undergraduate and graduate level courses in core electrical and computer engineering, including: programming, circuit theory, digital logs, embedded systems, signals and systems, and digital signal processing. In addition to ECE fundamentals, I strongly believe in providing students with opportunities to learn about active research and practical engineering skills that I bring from my research. In particular, I would like to teach courses on ubiquitous computing, physical computing, prototyping hardware/software interfaced systems, debugging, and effective communication for engineers. I hope that through teaching, I can find ways to instill my own teaching pedagogies into the curriculum and help improve the curriculum as a whole.