JULIAN RAMOS

(+1) 412-7088874 | ing.julianr@gmail.com | www.julian-ramos.com | Julian Ramos

Teaching Statement

One of the most rewarding experiences in my Ph.D. has been contributing to my students' success, whether it was completing a challenging project, getting into grad school, or getting their dream job. I have had the fortune to teach and mentor students with many different interests (UX, machine learning, technical HCI, data science) and at all levels (undergraduate, masters). I was a teaching assistant for two different courses Programming usable interfaces (PUI) class, an introductory web programming and design class, and User-Centered Research and Evaluation, a human-computer interaction systems design and analysis class. In both classes, I had to create and teach a 90 minutes recitation once a week, design and evaluate homework, midterms, and finals. I have also mentored students as part of their independent study courses, research projects, and lately as a mentor in the CMU-AI-Mentoring program. These experiences have allowed me to develop a teaching philosophy grounded on those interactions and my own experiences as a student.

Something I always do as a teacher is to **level the field**. My goal is to bring all students to the same level of knowledge to give them the same opportunity to grow. I accomplish this by paying close attention to quizzes, homework, and interactions to gauge the concepts students are struggling with quickly. Then, usually, at the beginning of my next lecture, I go through these concepts trying to dispel any confusion, and later I re-iterate them at any possible chance during lectures. Another strategy I use to level the field is to demystify experts: "A knows it all that does not make mistakes". This was very important for my PUI recitations, where I had students with no programming experience and experts. I noticed that the novice students were initially ashamed of writing code that did not do what they expected. My approach to helping them was to provide an in-class coding activity, give them some time to solve it, and then I went through it myself to show the process. While solving the activity by myself, I would make mistakes to show them that it was OK not to get your code right the first time and the value of testing. I also show them my approach to debugging because this is a skill that is difficult to acquire as a beginner. Also, to keep advanced students engaged, I provided them with optional challenges and exposure to advanced concepts and APIs. By the end of PUI, it was gratifying that none of my students dropped the class, novice students produced very polished final projects, and the more advanced students also felt very satisfied with the course.

Another crucial part of my teaching philosophy is to **reinforce** what they have learned. For example, during PUI recitation, usually, right after introducing a new programming concept, I modified the code used as an example. I then asked: "What do you think will happen now and why?". Usually, even the more advanced students would engage in the discussion since often, the changes I chose, although looked trivial, were not. During this process, I like to always acknowledge their approach because although they could be wrong, this will motivate them to keep trying and helped me understand their mental model to address any confusion.

As a teacher of courses on designing and implementing systems interacting with people, a fundamental part of my philosophy is **inclusion**. I encourage students to think deeply about their solutions, the assumptions necessary to hold, and how those solutions would be received or affect different people. My goal is to guide students towards humane and inclusive solutions, which means to think beyond just solving the problem in a cost effective way. For example, a group of students during UCRE proposed a smartphone app citizens could use to detect potholes from their phones' accelerometer data automatically. I highlighted to the students that this solution assumed that smartphones have the same capabilities (internet access and sensors) and are used by most of the population. On their own, the students realized how their solution would exclude reports from the most impoverished areas, leading to worsening those roads' state. They then set out to create alternatives to make their solution more inclusive.

Last, I like to highlight to students **the value of failure**. Whether it is on a homework or project, sometimes the final solution does not produce the desired outcome. In those cases I encourage students to summarize what they have learned and then evaluate whether they should keep on trying to improve their current approach or if instead they need to go back to the basics and re-think the problem to generate a better solution. Later on as they succeed I like to show them how those initial failures were fundamental at identifying crucial parts of the problem that were key to succeed.

Teaching experience

During my Ph.D. program I have participated in the design and teaching of two different courses at the Human-computer Interaction Department: User Centered Research and Evaluation (UCRE) and Programming Usable Interfaces (PUI). During UCRE, I was responsible for designing and testing the homework and exams for the class and teaching labs which includes preparing the teaching material and in-lab exercises. During sections I would usually cover in more detail the methods introduced in the main lecture and engage students in group exercises applying it. This was a very valuable experience because students got to use the different methods I introduce them directly in the lab exercises where they would get feedback from their peers and myself.

For PUI, I was responsible for preparing the teaching material for labs including slides, in-lab exercises and homework. This course was much more challenging than UCRE because the students background knowledge in programming varied from none to professional. This created some challenges however peer support and learning by demonstration sessions proved to be very valuable. The most rewarding part of this course was having some of the students that had no programming experience to create very interesting websites that were at the level of students with more programming experience. Likewise, students with programming experience also learned to use advanced APIs and libraries that I referred them to for their projects. I have also delivered guest lectures presenting my work in mobile health interventions using artificial intelligence and human-feedback on courses like Human-AI interaction and Programming Usable Interfaces.

Mentoring

My mentoring style depends on the students skills set and experience doing research. For example, one of my mentees already had experience applying machine learning in the finance sector, and he worked with me developing activity recognition classifiers for the elderly. With this student I was very hands off and guided his work from a very high level. With students that have less technical experience I walk them through research basics and the specific techniques they need to use, over time as they gain more confidence I become more hands off and at some point they become very independent. I deeply enjoyed working with my mentees and it has been very rewarding to see that their experience with me contributed towards their long term goals.

Future courses

Besides typical Human-computer interaction courses, as part of my course work, I took courses that were pre-requisite and part of the concentration areas for Ph.D. students in the machine learning and robotics program like Machine Learning (10701), Intermediate Statistics (10705), Deep Reinforcement Learning (10703) and Statistical Techniques in Robotics (16831). Together, these courses and over ten years of experience applying machine learning give me the qualifications necessary to teach Artificial intelligence and Data Science courses. The next are the courses I would be very excited to teach in the following areas (besides any core CS courses):

Human-computer interaction

- **User Centered Research and Evaluation:** In this course will be covered the basics of how to investigate users interactions to uncover needs, issues, behaviors and motivations with the goal of designing or modifying an existing digital service or product.
- **Interactive data science**. In this course will be covered the basics of data science for exploratory data analysis and visualization of human behavior as collected from interactions with computing systems like websites, mobile phones or IoT devices.
- **Mobile health**. In this course, I will cover the most fundamental machine learning pipelines for sensing health-related measures (e.g., Breathing rate from light absorption, heart rate from accelerometer data, eating episodes from sound data). I would also cover the latest advances in health interventions with technology components that adapt treatments using data feedback from sensors.

Artificial intelligence

- **Applied machine learning and signal processing** In this course, I will introduce classical machine learning methods, and modern approaches like deep learning architectures with a focus on applications in health and interaction techniques. Projects could range from hand-tracking using a webcam to manipulate the computer pointer and detecting eating episodes using a microphone near the throat.
- **Applied Data Science:** This course will cover the basics of data pre-processing, exploration, visualization, and statistical modeling. The course focus is on methods actively used in industrial applications like spam filtering, movie recommendation, website add placement, anomaly detection, among others.