Teaching Statement

Dr. Luís Gabriel Ganchinho de Pina

I have a passion for teaching and I look forward to develop it further by lecturing courses at several levels. In my teaching experience, I was involved in teaching a number of one-off guest lectures, a full course; and I took an intensive one-day teaching training. I have always found teaching rewarding and I have noticed increased student involvement as my teaching experience has grown. My teaching goal is to maximize the success of each individual student in my class. Classes should be an inclusive environment where all students take an active role in their learning, receive frequent feedback on their performance, and have many opportunities to voice their questions and concerns.

Teaching Approach

Active Learning. I believe students should take an active role in learning, for instance by solving exercises about the material just presented, either individually (replying through a clicker-style interaction) or in small groups. This allows to create "pauses" to break up the contents of one long lecture into two or three smaller parts. Treating students as such an active entity also provides a mutually-beneficial feedback loop, in which students notice questions about their understanding of the material, and in which I understand what techniques work for each class and the correct pace to present each topic in a lecture.

Listening to students. In class, I take a *"there are no wrong answers"* approach and I make a point to always have the floor open to questions from the students. I find that a motivated student interrupting a lecture with a pertinent question benefits the whole class. For most students, who prefer offline interactions, I make myself available in many different ways: I arrive to class at least 5 minutes before the start time and I stay after class ends for at least 10 minutes, so that students can meet me right before or after class; and I encourage students to attend office hours to meet me one-on-one, which I schedule to maximize availability to students. I value meeting students in person, but I also make myself available electronically through forums associated with the course.

Inclusivity. Each student has a different background, abilities, and identity. As such, I take inclusivity as a main goal so that no particular student feels left behind. For instance, to allow shy students to engage in active learning, I prepare multiple-answer questions that students can reply anonymously on the material just presented (*e.g.,* through clicker-style apps on their phone). I create grade incentives for students to participate, even if they do not get the expected answer. This incentive for students to attend class allows to record lectures and then make them available to students (*e.g.,* using *Panopto*), so they can review classes at their pace. As another example, I use gender-neutral pronouns in all hypothetical examples.

Frequent feedback. In class, my style of active learning provides students with several opportunities for feedback. I believe assignments should be broken into two or three parts, so that students can receive frequent feedback as they make progress.

Teaching Experience

My first experience with teaching was through guest lectures on my advisor's classes. In this setting, I taught about: serialization and linearizability, the Rust programming language (two lectures), OS linkers and loaders, and network name services/service discovery. This gave me the chance to design a whole lecture on my own, figure out what amount of material is adequate, how to split the material through multiple lectures, and how to connect my material with the broader scope of the class in which it is presented.

I was involved in teaching a whole course about Software Reliability as Course Support Leader. My direct teaching duties involved designing exercise sheets for each lecture; presenting them to the students in weekly tutorial lectures, in which we solve a sample of the exercises; and managing the Piazza forum for the class. My indirect teaching duties involved shaping the curriculum around new topics that were introduced this year: SAT solving, undefined behaviour in C, and program analysis for security. This involved discussing with the lecturers the depth and breadth of the material, which parts would be presented when, and how the provided reading material (*e.g.,* slides) should look to match the exercise sheets.

Teaching Education

I attended a one-day course titled *Graduate-Teaching Assistant Training* to qualify to interact with students at Imperial College London during Software Reliability. This was an intense training that focused on the rudiments of teaching, student learning and motivation, small-group teaching, and assessing/evaluating students and providing them with feedback.

Mentoring

I mentored three students at Imperial College London. Daniel Grumberg, a second-year undergraduate student, implemented a research prototype during a summer, and then authored a paper published at a top-tier conference. This experience led Daniel to pursue a PhD in Computer Science. Andrei Brabete, a second-year undergraduate student, implemented a series of research prototypes which laid the groundwork for a funded grant. Finally, Karolis Mituzas, a final-year master's student, re-engineered a part of an existing research prototype to become familiar with low-level OS linking/loading and x86-64 assembly. Karolis considers his project to be very important for obtaining his current job.

I believe the key to mentoring successful students is:

- **Clarify the scope and outcome as a student starts a project.** Research projects are prone to unclear scopes, which leaves students confused. I actively address this with an initial meeting where I explain clearly the expected scope and outcome and break the project down into many milestones, with the more ambitious being considered optional if progress is fast on the preceding ones. This meeting provides a baseline of expectations throughout the project.
- **Frequent feedback.** I schedule frequent meetings with students (*e.g.,* at least weekly) with a formulaic structure: Progress made since last meeting, current questions about the project and its direction, and goals for the next meeting. Frequent feedback allows to adjust expectations in terms of scope and outputs, which in turn keeps students motivated and engaged.
- **Availability.** Besides the scheduled meetings, I make myself available via email and, when needed, extra meetings either physically or electronically. I understand that research sometimes leaves students blocked trying to fix a problem that is beyond their current ability. I encourage my students to contact me when that happens, so I can help them resume progress well before the next scheduled meeting.

Course preferences

I can teach beginning programming classes (in any language or paradigm), as well as classes in systems or programming languages. In particular, I believe I'm mostly suited to teach: operating systems, compilers, program analysis, concurrent/distributed systems/algorithms, computer architecture, and cybersecurity.