> *"Tell me and I forget, teach me and I may remember, involve me and I learn." — Benjamin Franklin*

My teaching philosophy is grounded in one core principle: students truly learn only when they can see a skill's practical value in their own lives and future work. When learners recognize how a concept connects to problems they care about, the motivation to engage deeply follows naturally. Conversely, if the relevance is unclear, even the most elegant exposition may fall flat. This belief shapes every aspect of my course design, delivery, and mentorship.

Educational Contributions

Since joining CU Boulder, I have participated in the educational mission of both the Programming Languages and Verification (PLV) and the Systems groups. I have taught graduate courses such as:

- **Fundamentals of Programming Languages** (CSCI 5535/ECEN 5533; Fall 2021, Fall 2024)
- **Advanced Functional Programming** (CSCI 7000; Spring 2023)
- **Distributed Systems Verification** (CSCI 7000; Spring 2021, Spring 2024)

At the undergraduate level, I have taught **Programming Languages** (CSCI 3155) to hundreds of students across multiple semesters (Spring 2021, Spring 2022, Fall 2023, Spring 2025).

In CSCI 3155, students explore topics such as functional programming, operational semantics, interpreters, type systems, continuations, object-oriented programming, and garbage collection. In my graduate courses, topics range from formal semantics, type theory, and program verification to advanced functional programming paradigms, distributed algorithms, and verification techniques. Detailed schedules and materials for all my courses are publicly available:

- CSCI 5535: https://csci5535.cs.colorado.edu/f24/
- CSCI 7000 Advanced Functional Programming: https://gowthamk.github.io/csci7000_pfp_s23/
- CSCI 7000 Distributed Systems Verification: https://gowthamk.github.io/csci7000_s21/

For each of these graduate-level courses, I designed the curriculum and developed all instructional materials myself. These websites not only provide students with lecture notes, assignments, and code but also serve as open resources for the broader academic community.

Making Utility Visible

In my classrooms, I strive to bridge theory with lived application. When introducing an abstract concept—be it continuations in functional programming or model checking in distributed systems—I begin by situating the idea in a context the students can relate to. This could mean:

- Demonstrating how type systems prevent security vulnerabilities in real-world software.
- Connecting garbage collection algorithms to memory management issues in industry-scale systems.

- Linking verification techniques to the reliability demands of distributed systems like blockchain or NASA flight software.

By framing concepts in terms of tangible utility, students not only engage but also begin to transfer their learning to new, unfamiliar domains.

Course Design and Delivery

My classes are highly interactive and often make use of live coding, collaborative problem-solving, and real-time polling tools to check understanding and adjust pacing. I integrate project-based learning wherever possible—asking students to build interpreters, experiment with language features, or verify non-trivial protocols. These projects give students the dual satisfaction of mastering theory while producing something functional and relevant. I also maintain public websites, which play a crucial role in my teaching. Beyond providing transparency and accessibility, they allow students to revisit materials at their own pace and serve as a portfolio of my evolving pedagogical approach.

Mentorship

Mentoring is where my teaching philosophy extends most naturally into research. Since joining CU, I have advised 4 Ph.D. students, 3 M.S. students, and 7 undergraduates. All 7 of my major publications at CU have been co-authored with CU students.

Highlights of my mentoring include:

- My senior Ph.D. student, Nicholas Lewchenko, is on track to graduate in Fall 2025.
- Undergraduate thesis student Pranav Subramanian defended his thesis and joined NASA Goddard Space Flight Center as a Software Engineer.
- Two of my undergraduate mentees came from front range community colleges through the SPUR internship program.
- Current undergraduates Noah Schwartz and Oscar Bender-Stone are applying to Ph.D. programs this year.

As part of the CUPLV group, my students benefit from a vibrant research community. I help organize a weekly student-led seminar and twice-weekly status meetings, in addition to one-on-one weekly advising sessions. These structures foster collaboration, peer learning, and sustained research momentum.

Mentoring, to me, is a reciprocal process: my students bring fresh perspectives that challenge me to re-examine familiar problems, while I provide the guidance, focus, and encouragement needed to turn promising ideas into polished contributions.

Closing Thoughts

Whether in the classroom or the research lab, my aim is to spark curiosity by showing students the practical power of the concepts they are learning. When they can see how an idea might shape their own future projects or careers, the drive to master it comes naturally—and that, in my view, is when real learning happens.