Cheng Zhang

Teaching Statement

✉ *czhang03@bu.edu*
cs-people.bu.edu/czhang03/
czhang03
Curriculum vitae

"Tell me and I forget, teach me and I may remember, involve me and I learn." – Benjamin Franklin

My undergraduate experience at a liberal arts college was transformative, thanks to the exceptional teaching and mentorship I received from my professors. Their guidance and wisdom not only shaped my academic pursuits but also had a profound impact on my personal growth and development. I have been fortunate to maintain lasting relationships with many of my mentors, who continue to inspire me with their counsel. Through my teaching, I aspire to emulate their success by making a lasting difference in the lives of my students. My goal is to provide principled and transferable skill that will empower my students throughout their life, regardless of the paths they choose. By engaging and inspiring my students, I aim to equip them with the skills, confidence, and critical thinking abilities necessary to succeed in this ever-changing world.

Teaching Philosophy And Interests

The wide availability of information has transformed the way we learn and works. Rather than viewing emerging technologies as a threat to education, I see them as an opportunity to shift teaching focus from mundane memorization to mastering abstract core concepts and apply them on realistic problems. While I don't pretend to have all the answers to the challenges facing education today, my experiences as both a student and a teacher have led me to identify several key strategies that can help achieve this goal.

- **Core Concepts:** Simplicity is the key to engaging and effective learning, thus my classes are always centered around several widely-applicable yet simple core concepts. While students may not remember every minor detail, they should be able to recall and apply these fundamental ideas for problem-solving.
- **Grounded Presentations:** Abstraction is essential, but it must be grounded in context to be meaningful. To provide such contexts, I use a combination of real-world examples, interesting connections, and hands-on practice to help students learn abstract concepts. In class, I would introduce abstract ideas through relevant examples, and then provide opportunities for students to practice and reinforce their understanding through homework, labs, and exams.
- **Engagement:** I've observed that students often resort to cheating when they feel overwhelmed by the material or anxious about their grades. Rather than relying strictly on cheat-detection systems, I focus on engaging these students and addressing the underlying issues. To accomplish this, I design the homework and exams to provide constructive feedback and support, rather than discouragement. To accomplish this goal, I identified two priciples with inspiration from game design [1, 2]: allow students to quickly recover from mistakes, and have a clear alternative path to success.
- **Exploration:** Besides homework, exam, and extra credit problems, I strive to also provide abundant opportunities to mentor student to further explore their interest. These include study/reading group, research seminars, and undergrad research opportunities.

Teaching Interest: I have a passion for teaching a wide variety of classes, as demonstrated by my diverse teaching experience. However, my academic background in Kleene algebra has equipped me with specialized knowledge facilitates teaching topics like functional programming, programming languages, and algebra. In general, I take great pleasure in breaking down complex topics into manageable components and linking them with a few key concepts; this approach helps students maintaining a clear mental image of the class materials and grasp these contents more effectively. My greatest joy comes from witnessing students not only succeed but also innovate by applying these foundational ideas to tackle complex real-world problems and uncover fascinating connections on their own. Ultimately, I aspire to impart not only specific knowledge but also the underlying ideas, thereby inspiring students' passion for the subject and their love for learning in general.

Teaching and Organizing Experiences

Throughout my academic journey at Boston University, I have extensive experience in event organization and teaching, with a proven track record of creating engaging learning experiences for both graduate and undergraduate students.

As the founder and organizer of several study groups and seminars, I have successfully fostered a culture of collaborative learning and intellectual exploration among students of all levels. Notably, I organized the long-running BU POPV seminar and also reading groups focused on Coq/SSReflect and category theory, which attracted researchers and students from within and beyond the university to learn and share innovative research ideas.

As a teaching assistant for a diverse range of courses, including introductory programming, algebraic algorithm (basic number theory and abstract algebra), geometric algorithm (linear algebra), and programming languages, I have honed my skills in classroom management, student support, and curriculum design. Serving as a TA in many classes have given me invaluable experience in leading different grading teams, developing course materials, and also shaped my teaching philosophies.

In addition to my teaching assistant roles, I have had the privilege of mentoring two talented undergraduate researchers through the UR2PhD program. Over the course of a year, we worked together on cutting-edge research in Kleene Algebra, culminating in a paper submission to LICS 2025.

Selected Course Evaluations: *"Cheng often does this thing where he creates a map and connects all of the concepts in Linear Algebra together which is the best thing ever[,] because so much of this class builds on top of previous material, so it really helps you understand concepts being taught."*

"Cheng Zhang is the most helpful instructor I've ever seen. He responds super fast in Piazza while being able to explain complicated concepts clearly. Also, he is patient and friendly. I rate him a 100 out of 10. Love you :)"

"Cheng was a great TF. He wanted us students to learn with a passion. I wish there were more TF's like Mark and Cheng."

"Helpful in office hours and willing to schedule time outside of those in order to help."

"Cheng Zhang's only weakness is his backache. Poor boy."

Case Study: Functional Programming Class With Haskell

To illustrate my teaching philosophy, I'd like to share my experience co-designing a Haskell course, "CS320 Concepts of Programming Language," in Spring 2019. I will walk through my thought process and highlight the key improvements I aim to incorporate into my future teaching practices.

We chose Haskell as the teaching language due to its clear and consistent syntax and semantics. In contrast to impure languages like Ocaml or Scala, Haskell's pure functional programming paradigm forces students to think functionally and avoid relying on impure constructs like references and loops.

To support students, teaching assistants, like myself, maintained an extensive online presence through Piazza and held sufficient office hours to help students with challenging questions. Notably, we intentionally scheduled office hours away from homework deadlines to encourage students to start assignments early and engage in collaborative discussions on Piazza after office hours. We also implemented a generous grace period for submissions, averaging around half a day, and allowed resubmissions for up to two assignments at the end of the semester.

To provide rapid feedback for homework, our grading system utilized QuickCheck which can provide immediate counter-examples to facilitate effective debugging. While this approach required significant effort to ensure that our tests were thorough yet did not reveal answers, we believe that the benefits of immediate offline feedback outweigh the costs. However, we acknowledge that mere counterexamples may not be sufficient for students with fundamental conceptual misunderstandings. To address this, we encouraged students to seek help online or in study groups. In the future, I am considering the development of misconception-based teaching tools, such as the LTL Tutor [3], to further support student learning. I am always eager to collaborate with social scientists, HCI experts, and students to design effective teaching systems that cater to diverse learning needs.

Finally, to develop the class material, we first identified four core concepts: functional programming, algebraic data types, type-driven development, and monads. We recognized that monads would be the most challenging concept to teach, so we decided to introduce them as a design pattern instead of mathematical concepts, while providing abundant examples to make the concept more accessible to students.

Our curriculum was then designed around these four core concepts, starting with an introduction to Haskell without prelude. We guided students through the implementation of familiar structures like boolean values, natural numbers, lists, and maybe types, allowing them to practice defining, destructing, and recursing on algebraic data types. Next, we had students implement and use functional primitives like "map", "filter", and "fold" to familiarize them with manipulating collection data structures. When implementing these primitives, we demonstrate type driven development by listing the type of all the available arguments, and show that function like "map" and "fold" can be implemented in one meaningful way.

We then introduced the data type of reader and state monad, without introducing monads, and had students work on small computations using these types. Once students were comfortable tracing the monadic computations, we introduced the concept of monads and do expressions. The final project, which is split into several assignments, asked students to implement a simple parser using the parser monad. Most students were able to successfully implement the parser monad using type-driven development, understand the semantics of the monad, and create a functional parser. Many were even able to handle associativity with the help of examples.

I believe that this class design reflects my teaching philosophy, which focus on few important abstract concepts while providing students with rapid feedback and support. As a result, students felt a sense of progress and accomplishment as they solved problems, but not punished when they fail. The success of our approach is evident in the final project, where most students were able to produce a competent parser in just a few weeks, with no prior experience, demonstrating effective utilization of the four core concepts.

References

[1] Geoffrey Engelstein. *Achievement Relocked: Loss Aversion and Game Design*. MIT Press, Feb. 2020. ISBN: 978-0-262-35705-0.

[2] Chris Lewis, Noah Wardrip-Fruin, and Jim Whitehead. “Motivational Game Design Patterns of ’ville Games”. In: *Proceedings of the International Conference on the Foundations of Digital Games*. Raleigh North Carolina: ACM, May 2012, pp. 172–179. ISBN: 978-1-4503-1333-9. DOI: 10.1145/2282338.2282373. (Visited on 10/03/2024).

[3] Brown PLT. *LTL Tutor*. https://blog.brownplt.org/2024/08/08/ltltutor.html. Blog. Aug. 2024. (Visited on 10/03/2024).