Kirshanthan Sundararajah
Teaching Statement

I believe that teaching strengthens one's understanding of the subject matter. Observing the growth in students' understanding when I teach makes it enjoyable and often gives me a new perspective on the subject matter. In my opinion, teaching is a communication medium for your understanding of a subject. Therefore, finding effective strategies to communicate understanding helps to achieve our teaching goals.

Teaching Experience

My teaching experiences are *diverse*. I have teaching experiences at universities in two different countries, and I have served as a teaching assistant and a primary instructor. These experiences have shaped my thoughts on both learning and teaching. My diverse teaching experiences largely contributed to developing a concrete teaching philosophy of *fairness, accessibility, and collaboration* to bring the best learning outcomes.

Teaching Assistantships. My very first teaching experience was as teaching assistant for **Programming Fundamentals (CS 1032)** at the University of Moratuwa. I was mainly responsible for conducting introductory programming labs in python for first-year engineering students. At that time, most of the students were first-time programmers, and my main challenge was to provide everyone with a fair chance to discuss the problems of programming labs and assignments. Through this experience, I have learned a very valuable lesson: Not everyone has the same resources and learns the same way. Therefore, I cannot teach everyone the same way and expect them to excel. Significantly, first-time programmers may have difficulty formulating their confusion, so I learned to help clarify their doubts based on their needs. I find students were particularly interested when solving a fun problem such as writing programs to draw shapes. At Purdue University, I served as teaching assistant for the **Introduction to Data Science (ECE 295)** class, and this was my first experience helping with an online course. I discovered that when I explain something in-person from the instant feedback I get, I was able to recalibrate myself and clear the student's misunderstanding quickly. But in the online setup, since I did not have that instant feedback, I had to think twice before explaining anything. I had to come up with detailed but simple examples to explain concepts. These teaching experiences helped me to come up with the tools to deliver strong understanding of the concepts.

Primary Instructor. I taught **Data Structures (ECE 308)** in the summers of 2021 and 2022. Both sessions had more than 60 students. Preparing course materials, delivering lectures, and evaluating the students were my core responsibilities. Usually, this is an in-person class, but due to the pandemic, I had to teach the class online. I designed my lectures sufficiently hands-on to keep a greater engagement of the class, thereby overcoming the challenges of asynchronous learning. In addition to the existing learning objectives, I wanted to equip students with skills needed for an *algorithmic programming interview*. I delivered the lectures via zoom and made the recorded video available for later viewing. I made lecture slides available before the lectures. I like to write on the slides (*e.g.,* explaining running examples) and released the annotated version after class. Even though there were allocated times for office hours, I had a virtual open-door policy (via email and discord), because there were students in different time zones, and a big chunk of the class was on summer internships.

Over the course, I found certain teaching practices are more effective than others to increase classroom engagement and improve students' understanding. For instance, students showed a greater engagement when I had *semi-improvised live programming sessions* in class and when given programming assignments based on real-world examples. In my opinion, you truly understand a complicated concept when you can explain it in simpler terms. One of my exams was akin to a *programming interview* where students were given a well-defined problem and asked for a solution in code. For the latter part of the exam, they were asked to solve a more challenging version of the previous problem, but they were not asked for a complete solution in code. Instead, they were asked to explain how they would solve the problem. When students explained their solutions, I observed that they were correcting mistakes in their own code. Most students found this experience exciting and valuable. Following are some of the feedback I received from students for my summer 2022 course evaluation.

> "*Kirshanthan is a great instructor who clearly conveys concepts, shows great concern towards the needs of his students (it making time to meet them or answering questions frequently) and is a fair grader that has made the class challenging for its conceptual value but not immensely difficult to make me frustrated at any point of time throughout the course.*"

> "*He does a really good job at explaining the architecture of our code in a way that can make sense. Even in the coding interview he seemed to care much less about a specific grade, and more about helping us to learn how to do specific things.*"

"*I liked the interview style exam. I haven't gotten many practical experiences like that exam in college classes. I also liked that we got 2 submissions and feedback on programming assignments.*"

Teaching Philosophy

Impactful teaching starts from the initiative to promote inclusivity. I always strive to maintain a fair, accessible, and collaborative classroom: so that everyone feels welcome to express their thoughts without hesitation. Underrepresented students may especially find it difficult to express themselves in front of everyone. Hence, I consciously try to keep an interactive environment by allocating time for discussion and encouraging questions from everyone by using polls and surveys for participation credits. An inclusive classroom helps every student achieve learning objectives while making teaching effective and efficient. In my view, effective teaching provides both understanding and knowledge of the subject matter. It takes significant effort to design and implement the material needed to understand the subject matter effectively, but it is also difficult to assess whether the students have acquired true understanding when it comes to feedback and evaluation. Therefore, I have developed my teaching philosophy around building the tools to deliver a solid understanding of the subject matter. I believe that the following elements in teaching help construct a strong understanding of the concepts.

Well-established big picture of the subject: I endeavor to first spend enough time explaining the bird's eye view of the subject. It helps to build an essential purpose for learning the material, and in turn, it strengthens the understanding.

Engaging and logically sequenced lectures: In my experience, cohesively presented subject matter has a high retention rate. I show a mind map of modules at the beginning of a course for students to understand how each module is connected to others. I revisit this mind map throughout the course to remind them of the big picture.

Fully encapsulated examples and analogies: When teaching complex concepts or recalling previous concepts, it is common to lose track. In some cases, too many details occlude the core concept and prevent students from understanding. Therefore, I structure my teaching materials around simplified examples and analogies whenever possible. Such examples makes teaching and studying easy and convenient to remind previously taught concepts.

Balanced assessment and evaluation: I find assessments that require regurgitating subject matter off-putting and skew the purpose of learning. Inspired from teaching the data structures class, I like to design a significant portion of the assessment focused on applying knowledge to well-structured practical problems. This makes assessment more enjoyable for the students, especially those who prefer a hands-on learning experience.

Courses

I am broadly interested in compilers, programming languages, and high-performance computing and my research is focuses on optimizing recursive programs. I anticipate being able to teach the following computer science and engineering courses:

Undergraduate-Level: Discrete Mathematics, Data Structures and Algorithms, Compilers, Programming Languages, Parallel Programming, and Operating Systems

Graduate-Level: Compilers, High-Performance Computing, and Performance Engineering

Advanced Topics Course: Optimizing Compilers. Given the opportunity, I would like to design my own advanced topics course for graduate students. A typical graduate-level course covers the inner workings of a compiler, but it is too complex to go deep into any particular analysis or transformation. The main learning objective of this course is providing the initial knowledge to conduct research in compilers. Modern compiler development has different styles with more and more functional programming concepts spilling over to the world of imperative programming. I would like to include the following topics focusing on optimizing compilers; Anatomy of a compiler pass, Static Single Assignment (SSA) form and Intermediate Representations (IRs), Advanced data-flow analysis (Partial Redundancy Elimination), Modeling loops, recursion and transformation frameworks, Automatic parallelization, Learning techniques in compiler optimization and autotuning, Domain Specific Languages (DSLs) and optimizations, Multi-stage programming and program generators, and Partial Evaluation.

Conclusion

My teaching philosophy is centered around strategies delivering the best understanding of the subject and making learning easy and enjoyable. Productive strategies to communicate our understanding make us better teachers, and I develop such strategies using my past diverse teaching experiences. In my humble opinion, teaching becomes impactful when we keep up inclusivity and accessibility.