Dominik Winterer — Teaching Statement

ETH Zurich, Switzerland

November 2024

My goal as a teacher is to spark students' enthusiasm to make learning the difficult subjects of computer science easier. To achieve this, my philosophy is based on three principles: (1) providing solid fundamentals, (2) combining theory and practice, and (3) eliminating obstacles to learning. I developed these principles in 8 years of teaching at ETH Zurich and the University of Freiburg, mentoring students and observing great teachers. **As a faculty, I am excited to teach graduate classes in Software Engineering, Formal Methods, and Programming Languages, as well as undergraduate classes in Algorithms and Theory.** Besides, I am well-qualified to teach classes in Programming, Databases, and Mathematical Logic. Besides teaching, I am passionate about mentoring and making a broader impact. Students of diverse backgrounds make remarkable achievements under my supervision and mentoring: top-tier publications, awards at student research competitions, landing Ph.D. offers at top places, etc.

1 Teaching

Teaching Philosophy

My teaching philosophy is based on three core principles: (1) providing solid fundamentals, (2) combining theory and practice, and (3) eliminating obstacles to learning.

Providing Solid Fundamentals Given the rapid evolution and specialization of computer science, developing *transferable skills* is becoming ever more critical. One way of helping students develop transferable skills is by insisting on solid fundamentals that are useful beyond the particular class at hand. As a head teaching assistant of the software engineering class at ETH Zurich, I established that each lecture and exercise session starts with a brief summary of the most important fundamentals. I further re-designed the assignments to expose students' knowledge gaps as early as possible using short, carefully chosen quiz questions requiring deep thinking; easily solvable when core concepts are understood but very hard if not. Our positive evaluations showed the effectiveness of this approach.

Combining Theory and Practice One of the key tasks of a computer scientist is to transform abstract ideas into working systems. To help students do that, we need to link rigorous theory with resolute practice. At ETH Zurich, I was instrumental in establishing and developing a new class on Compiler Design. I ensured that the lectures and exercises always contained both theoretical and practical parts, which were interconnected. For the practical part, which includes programming tasks, I believe it is important to be up to date with the latest technology stack. With the vast majority of students going to industry, this can facilitate their transition. During the five years I was involved in the Compiler Design class, I updated the programming assignments to work with recent compilers and hardware and gave students incentives to use the newest PL features.

Eliminating Obstacles to Learning Great teaching alone is not enough. A major part of the success of a class is determined by its structure and organization. For students to focus, it is crucial to eliminate any obstacle to learning. At ETH Zurich, I had a great influence on establishing the Compilers Design class, which every CS student has to take. During five years, I made a substantial effort to eliminate any obstacle to learning, especially when I was the head teaching assistant of the class. I studied thousands of feedback forms and forum discussions of various classes, and so eliminated many issues. I further tightened the organization around the teaching team and defined clear responsibilities for all parties: Lecturers, TAs, and students. This significantly improved the quality of the class, as a result. It has been complimented on its organization and the timely responses to students' questions, receiving high ratings from ETH students despite being one of the most labor-intensive classes at ETH Zurich's CS curriculum.

Teaching Experience

At ETH Zurich, I have been a teaching assistant for more than 5 years; almost every spring and fall semester. I was a head teaching assistant and guest lecturer for large courses including Compiler Design (200+ students, 9 TAs) and Software Engineering (100+ students, 5 TAs). The head teaching assistant at ETH is a role with great responsibility including guest lecturing, designing and grading exams, giving exercise sessions, and coordinating the teaching team of the course at hand. My experience covers multiple subareas of CS including Compiler Design, Data Modeling and Databases, Formal Methods and Functional Programming, Algorithms & Data Structures, Theoretical CS, and Model Checking. I was a member of the teaching committee of ETH Zurich for 1.5 years. In the teaching committee, I established awards for TAs which previously only existed for faculty. My idea behind this was to incentivize teaching assistants to excel.

Teaching Interests

My teaching experience covers various areas including Compilers, Software Engineering, Formal Methods, Programming Languages, Databases, Theory, Algorithms and Data Structures, Logic, and Symbolic AI. I am prepared to teach courses in all these areas. However, because of my research interests, I am particularly eager to teach classes in Software Engineering, Formal Methods, and Programming Languages. As a vision for the future, I aim to develop a class on Automated Software Testing similar to the one I have been co-teaching at ETH Zurich and a class on Formal Methods Engineering combining both the theory and practice of Formal Methods. As a long-term career goal, I aspire to make one of those courses (most probably Formal Methods Engineering) my signature class and develop an (open-access) book for it.

2 Mentoring & Broader Impact

Mentoring Approach

Mentoring students has been *the* most rewarding and meaningful aspect of my academic career. My aim for students working with me is to train them to become great researchers and computer scientists based on their individual goals, ambitions, and needs. To do this, I first and foremost try to lead by example. I encourage students to be ambitious and help them based on their individual needs. The best work is done if the interests of all parties meet in a welcoming environment. Such an environment incentivizes passionate, consistent, hard work but prevents unhealthy pressure, excessive working hours, and other toxic situations. I approach this job with a humility, gratitude, compassion, and responsibility.

I have been advising more than a dozen BSc. and MSc. students at ETH Zurich. The students have made remarkable achievements under my supervision. For example, the projects of my students Jiwon Park and Mauro Bringolf, have led to two top-tier conference papers, on which they are (co-)first authors. Moreover, my students Jiwon Park, Vince Szabo, and Dylan Wolff submitted abstracts to student research competitions (SRC). Jiwon won the second prize at the FSE '21 SRC and secured a Ph.D. position at UC Berkeley and Dylan was able to secure a Ph.D. position at the National University of Singapore. The works of Vince Szabo and Thea Ullegård Kjeldsmark are under submission and in preparation, respectively. **As a faculty, I look forward to continuing my great record as a supervisor and start advising Ph.D. students.**

Broader Impact

Besides supervision at university, I am an active mentor in the SIGPLAN-M longterm mentoring program by the programming language community since October 2021. As I had great mentors throughout my career, I am excited to participate in such a program, giving back, serving students needing advice. Currently, I am assigned to two students, mainly providing advice on on the Ph.D. job market. One of my mentees has recently secured a Ph.D. position at the University of Pennsylvania.

I like to organize and attend seminars. Great seminars can be a lot of fun and serve as a glue between research groups. One such example is the Future of Software Seminar at ETH Zurich which I organized. Students and faculty of many groups at ETH Zurich and other places came together to watch talks and discuss, to the enjoyment of everybody involved.