I find great fulfillment in teaching and mentoring students at different levels, as it allows me to contribute to the growth of future engineers and independent researchers. This role also keeps me engaged and up-to-date in my field. Over the years, I have taught industry courses, delivered academic lectures, and spoken at research events. I have also guided bachelor's, master's, and PhD students during my doctoral studies. My experience in the software industry, where I have worked as a developer, designer, architect, and team lead, brings practical perspectives that enhance the courses I teach.

Course Offering. During my Ph.D. and postdoc, my research focused on program synthesis and program analysis at the intersection of software architecture, software security, formal methods, and LLMs. As a result, I am well-prepared to teach both introductory and advanced courses for graduate and Ph.D. students, including software design and implementation, program analysis, , software security and quality, software evolution and maintenance, and algorithms. Additionally, I am equipped to develop and teach coursesboth introductory and advancedon program synthesis and auto-coding, software architecture, formal methods, and practical aspects of compilers. Furthermore, with a bachelor's degree in software engineering and 14 years of industry experience, I am capable of teaching fundamental software engineering courses for bachelor's (and upper-level) students, such as foundations of computer science and software engineering, software development, quality and security, introduction to software design and object oriented principals, and algorithms.

Teaching Philosophy, Methodology, and Experience. Teaching involves mastering complex concepts and effectively conveying them in a clear and engaging way. To keep students motivated and encourage active learning, I focus on teaching core principles while allowing them to explore specific aspects through practical applications. I adapt my teaching approach to their maturity and independence levels. For undergraduates, I typically cover most of the material in class and encourage exploration through assignments and semester-long projects. The assignments apply taught concepts to practical, relatable scenarios, helping students connect their learning to real-world contexts. This approach enhances their understanding and maintains their motivation. In graduate and Ph.D.-level courses, research-related activities are central. Depending on the class composition, teamsoften led by Ph.D. studentsundertake original research projects, aiming to produce preliminary results or proofs-of-concept by semester's end. This approach helps students become familiar with the field and inspires them to continue the research beyond the course. Additionally, I incorporate elements of flipped learning based on the course level. I may share pre-recorded lectures, key papers, or relevant book chapters in advance. For students who may not fully engage with these materials beforehand, I begin class with concise summaries to ensure everyone is on the same page before diving into the topic. This approach supports a deeper and more interactive learning experience.

I have accumulated teaching and lecturing experience in both academic and industry settings:

– **Guest instructor in academia:** At Virginia Tech, I delivered a lecture on LLM-based type recovery from binaries to graduate-level students. This session involved a lecture followed by problem-solving exercises and mentorship on research directions. Additionally, I taught an introduction to program synthesis to graduate students (Ph.D. and Master's) at Rochester Institute of Technology. This session included a lecture, interactive discussions, teamwork, and problem-solving activities. [Video Link: https://www.youtube.com/watch?v=wvbqBAWnMgY]

– **Teaching in industry:** I designed and taught courses on Java programming and software design patterns for software engineers, addressing both foundational and advanced topics. These sessions emphasized practical applications and real-world software development scenarios.

– **Invited speaker:** I have had the privilege of being invited to give talks at various academic research groups (e.g., CHAATS talks and the Ph.D. Colloquium at RIT) and industry forums (e.g., Palo-Alto Research Center).

Mentorship Philosophy, Methodology, and Experience. I define mentorship as guiding students to discover their own path to achieving their goals, rather than prescribing a direct,

pre-defined route. A skilled mentor helps mentees grow into the best version of themselves, not merely a reflection of the mentor. Effective mentorship requires flexibility and tailoring approaches to suit students' diverse backgrounds and needs. For undergraduates, detailed planning, task breakdowns, step-by-step interactions, and frequent office hours are essential. In contrast, Ph.D. students are mentored to become independent researchers by involving them in activities such as grant proposal writing, mentoring junior students, and engaging in light teaching tailored to their career goals. Recognizing that Ph.D. students are not all at the same level, I employ different strategies to keep them engaged. For instance, I may set a tangible goal, such as submitting a paper based on their current work to a targeted conference before their proposal exam, enabling the paper to contribute to their dissertation. Graduate students pursuing master's degrees receive mentorship tailored to their career preferences, whether it aligns more closely with undergraduate-style guidance, Ph.D.-level research training, or a blend of both. I also involve my studentsparticularly Ph.D. studentsin PI meeting-related activities, such as preparing reports and delivering oral presentations. This exposure helps them build professional connections with project managers and PIs from other institutions, putting their names on the map. Traveling with students to conferences is another avenue through which I mentor them indirectly. This experience allows students to develop critical skills, such as initiating or participating in collaborative research efforts, while expanding their professional networks.

During my Ph.D. and postdoc, I had the privilege of mentoring students from diverse backgrounds, spanning various academic levels and career aspirations:

– **Undergraduate:** I mentored Denis Zhenilov, a software developer from Budapest, Hungary, during his time as an undergraduate researcher in 2019. The focus was to guide him in effectively executing tasks and achieving tangible outcomes from his studies.

– **Masters:** I have mentored and am currently mentoring five Master's students, each pursuing unique career paths. At Virginia Tech, I am mentoring Priyatam Annambhotla, an M.Sc. student in Computer Engineering. At RIT, I mentored Chinmay Singh (now at Salesforce), Viral Parmar (now at Microsoft), Ishika Prasad (now at Motive), and Lorena Mendes (now at GoTo). My mentorship approach was tailored to their individual goals: for Priyatam, Chinmay, and Lorena, I focused on developing research-oriented critical thinking to enhance their ability to approach challenges analytically. For Viral and Ishika, who pursued careers in engineering, I emphasized strengthening their creativity, algorithmic planning, and practical execution skills.

– **Ph.D. student:** At Virginia Tech, I am mentoring Yuanzhuo Zhang, a Ph.D. student in ECE. Additionally, I played a significant role in mentoring Mohamad Fazelinia, a Ph.D. student at RIT. My mentorship included research-focused feedback, introducing innovative ideas in program analysis, synthesis, and fuzzing, and holding regular meetings to refine their research skills. These sessions also covered discussions on future academic and career opportunities, providing strategic guidance and encouragement to help them confidently achieve their goals.

Beyond academia, I mentored many talented software engineers as a team lead in the industry, providing guidance on their work, career development, and problem-solving skills.

Lastly, I have supported many researchers in the field of software engineering through my service as a program committee member for various tracks of top-ranked conferences, including USENIX'24, ECSA'24, SANER'24, ICST'24, MSR'23, ICSE'23, ASE'22, SCAM'22, ICSME'22, ISSTA'22, PLDI'22, ICPC'22, ECOOP'21, and ICSME'21. Additionally, I have served as a reviewer for prestigious journals such as IEEE Transactions on Software Engineering, IEEE Software, and the Journal of Systems & Software.