Shanto Rahman

Teaching Statement

I am excited to be a teacher, helping students to turn curiosity into capability and to apply knowledge to real problems. I am committed to creating an inclusive and stimulating learning environment where every student can engage confidently, think independently, and translate theory into practice.

Teaching Experience

I had several teaching opportunities in different roles: (1) I was a faculty member at Bangladesh University of Professionals (BUP), where I independently taught eight undergraduate courses (e.g., Software Engineering, Data Structures, Operating Systems, Artificial Intelligence), held office hours, and mentored students; (2) I was a teaching assistant (TA) at The University of Texas at Austin (UT Austin), leading weekly recitations, grading, and mentoring project teams; and (3) I gave guest lectures in a graduate-level course at UT Austin. Across these roles, I emphasize applied learning, meta-learning, feedback, and inclusive participation.

Applying Knowledge. It is important for students to apply classroom learning to real problems. In each course, I integrate hands-on projects with brief real examples so students can see the impact of their learning. As a TA for ECE 422C, *Software Design and Implementation II* at UT Austin, I helped design assignments that enable students to apply course concepts to concrete tasks. For example, after teaching graph algorithms such as BFS and DFS, students implemented a *Word Ladder* game that uses one algorithm (e.g., BFS) to find the shortest word transformation and then compared the result with another algorithm, such as DFS, to understand algorithmic choices for the problem. In a senior undergraduate course in ICE 4107, *Artificial Intelligence* at BUP, I used similar strategies. After covering core concepts such as regression, neural networks, and decision trees, students completed biweekly exercises and then undertook projects such as *Sentiment Analysis*. They implemented at least two concepts from the course on public datasets, and compared results using standard metrics. This structure strengthens conceptual understanding and builds practical skills in selecting, implementing, and justifying models on real data. To connect research and teaching, I guest lectured in ECE 382V, *Software Testing in the Era of Nondeterminism*, at UT Austin. Drawing on my research experiences at Amazon and Google, I presented step-by-step approaches to industry challenges such as test repair using generative AI (GenAI), so students could see emerging trends and build relevant skills. After the lecture, a student contacted me and began a research project under my supervision.

Meta-Learning. As programming languages, tools, and AI frameworks change quickly, students must learn how to learn. I therefore teach transferable learning strategies and disciplined debugging. In ECE 422C, I led a recitation that began with an interactive 30-minute mini-lecture, followed by a 60-minute work period in which students either implemented the new concept or used a tool to learn how to adapt and apply it. For example, after discussing the concept of multi-threading, I asked students to use a research tool, TSVD4J, that I developed to find concurrency bugs. During the work period, I guided them through the debugging process, asking them to inspect each reported location, reason about the underlying concurrency bug, and repair it. In ICE 4107 at BUP, students wrote brief postmortems after each assignment, tracing defects to root causes, recording the steps they took to repair them. In another course, ICE 3101, *Analysis and Design of Algorithms* at BUP, I ran a complementary session, each group presented a 10-minute demo of an algorithm that was not covered in class, using a small example. These processes make students independent problem solvers, so they can approach new problems systematically and require less help. In ICE 3101, *Software Engineering* at BUP, I used an open-book, time-boxed midterm with no communication to solve a realistic scenario (e.g., a library management system). Given the scenario, students identified the requirements, wrote use cases with acceptance criteria, sketched UML diagrams, justified a process model with a mini release plan, and outlined a test strategy. This format shows students' reasoning and ability to tackle new problems.

Feedback. I view feedback or assessment as a way to deepen learning, not just assign grades, and I align each task with clear learning objectives and a concise rubric (i.e., correctness, design quality, test coverage, and usability). For a small assignment such as *Word Ladder* in ECE 422C, I assessed technical precision, including correctness of logic and output, implementation design, and code coverage. For larger, open-ended projects, I used milestone-based evaluation with targeted feedback. For example, for the *Sentiment Analysis* project, students wrote a design specification, trained a model, and built a lightweight interface, refining

each step with a single rubric. I divided the project into multiple milestones and provided detailed feedback on each. Students submitted each milestone as a pull request with a brief, rubric-based code review. For quizzes, I often build a degree of flexibility or second chances. For example, if a student does poorly on an early quiz, I allow them to regain points by demonstrating improved understanding on a later exam covering the same material. This approach motivates students to continue to engage with challenging topics and to learn from mistakes instead of feeling discouraged by a setback.

Inclusive Education. I ensure every student has a comfortable way to participate and get help. I set clear norms that welcome any question and provide anonymous channels for help, such as a brief post-class Google Form and Ed Discussion for unclear points, so students can seek support without embarrassment. I also run quick check-ins to identify blockers early and adjust instructions accordingly. In the ICE 3101 course at BUP with 100+ students, I used live polls and short in-class quizzes to gauge understanding and adapt the pace. In ECE 422C recitations at UT Austin, I implemented pair programming to create an atmosphere where students articulate their thinking, learn from peers, and engage more confidently with challenging material. Together, these practices make the classroom accessible, collaborative and supportive, helping students engage confidently and contribute meaningfully. In ECE 422C course feedback, one student wrote: *I'm usually shy about asking questions, but she created an atmosphere where I felt comfortable speaking up. She kept an open-door approach to questions and made sure everyone understood the task before moving on.*

Mentoring Experience

Research Mentoring. My research mentoring focuses on developing independent researchers, helping students with their goals, and providing guidance and support based on their unique paths. I have guided 10 mentees, six undergraduates and four master's students, from diverse backgrounds. I help students break complex problems into clear milestones, meet regularly one-on-one to refine ideas and track progress. For students new to research, I start with concrete, well-scoped tasks to build core skills. I am explicit about the "why" behind each task so that students see how their work fits into the broader research direction and begin to develop their own research taste. This model has led to tangible outcomes. My mentees and I have had *five conference papers accepted*, and several students have begun graduate study or started full-time roles at major companies such as Amazon. Above all, I aim to cultivate independent thinkers who can communicate clearly, reason rigorously, and carry forward best practices in reproducible, collaborative research. One mentee shared: *I used to fear reading research papers. She (Shanto) taught me to read them critically, break problems down, and learn from each step. I'm grateful to her.*

Competitions and Outreach Mentoring. Beyond one-on-one research advising, I mentor students through workshops, competitions, and campus programs. At UT Austin, I advised undergraduate projects and served as a judge for the *Capital of Texas Undergraduate Research Conference*. At BUP, I mentored a team for *National Hackathon on Frontier Technologies* by helping them conceptualize and build an end-to-end system development which won the championship. As Moderator of the *BUP InfoTech Club*, I organized programming contests, trained students in algorithms and problem solving, and coached teams for national and international level programming contests such as *ACM ICPC, NCPC*, and *NGPC*. I also co-organized the *UT–Cornell Software Engineering Seminar*, coordinating speakers, scheduling, and post-talk discussions to foster cross-institutional collaboration and student engagement. I delivered two talks in the *ECE Outstanding Student Lecture Series*, aimed primarily at incoming PhD cohorts, to showcase ongoing ECE research and help students connect with labs and collaborations. I also mentored students on Amazon technologies and supported students with referrals for internships as Amazon UT Campus Ambassador.

Teaching Interests

At the undergraduate level, I am excited to teach *Software Engineering*, *Software Design*, *Object-Oriented Programming*, *Data Structures and Algorithms*, *Database Management Systems* and *Artificial Intelligence*, as well as other courses based on departmental needs. At the graduate level, I will offer a research-intensive course, *Software Testing in the Era of Artificial Intelligence*, focused on AI-guided test repair, test generation, and test selection or prioritization. Drawing on my work in AI-driven software testing, I also plan a course on Software Engineering for AI (SE4AI) that focuses on end-to-end testing of machine learning (ML) models, covering data quality checks, nondeterminism analysis, reliability under drift, robustness, security, and fairness testing, and corresponding repair strategies when these properties are violated.