Teaching Statement

Teaching Philosophy

My teaching philosophy draws from my experiences as a computer science engineering instructor and my training as a computing education researcher. As a teacher, I focus on using **evidence-based instructional practices**, i.e. practices which have shown to be effective for student learning, such as "Think-Pair-Share" and "Peer Instruction". These **learner-centric practices** aim to actively engage students with the content by enabling them to talk, write, reflect and express their thinking.

An advantage of being a computing education researcher is the **synergy it creates between teaching and research** - what I learn from my research can feed into my teaching, and intuitions and experiences I gain from teaching students can feed into future research directions. For example, my research statement describes future projects which I intend to undertake - of improving students' metacognitive skills in programming, and improving introductory computing experiences for CS and non-CS majors. To address these research goals, I will understand what difficulties CS and non-CS majors face while programming (based on findings from literature and reflections from teaching my own courses). I will then design learning strategies and use these strategies in my teaching. I can empirically validate the effectiveness of these strategies, which can lead to future research directions.

Courses I would like to teach

Based on my educational and research experiences in Computer Science and Educational Technology, and my experiences as a software developer and computer science engineering instructor, I am enthusiastic in teaching courses in the broad areas of Software Engineering and Educational Technology. I provide details below.

Courses in Software Engineering and Introductory Programming

- **Human-Centric Software Engineering:** This course will focus on the human aspects of software development, and provide an overview of skills and processes developers experience in their workplace. The course will focus on fundamental concepts in developing software, such as software design and architecture, and practices employed by software developers such as Requirement gathering, Software Comprehension, Debugging, Testing and Verification.

- **Empirical Methods in Software Engineering:** The aim of this course is to prepare computer science PhD students to conduct empirical studies as part of their research. The course will cover topics such as research paradigms (positivist, interpretive etc.), formulating research questions, study design, and introduce students to various quantitative and qualitative research methods which are used in software engineering research.

- **Introductory Programming and Data Structures:** These courses are the introductory courses for freshmen, which I intend to take for CS and non-CS majors as well. The course will teach the

fundamental paradigms and constructs of programming, such as variables, conditionals, loops, functions and relevant data structures.

I intend to follow a purpose-first programming approach, where I will introduce examples relevant to students of a particular major, without initially focusing on the syntax and semantics of a programming language *(more details of this approach can be found in my research statement)*.

The main goal of the introductory programming and software engineering courses is to prepare students develop the essential skills required to become effective software engineers. I believe that my knowledge of computing education literature, and my experience as a computer science instructor will help me effectively teach these subjects.

Courses in Educational Technology and Computing Education

- **Introduction to Educational Technology:** In this course, students will learn about the foundations of educational technology. The course will cover topics such as theories of learning (e.g. constructivist and situated learning), instructional design models, multimedia principles, and technology tools and environments for learning.

- **Designing Learning Environments:** This course builds on the "Introduction to Educational Technology" course, and will focus on foundational topics of learning environment design and its features, such as design models, scaffolding frameworks and technology supports for learning.

- **Introduction to Computing Education Research:** This course is primarily for graduate students who want to pursue research in computing education and allied fields. The course will primarily be based on reading seminal papers and book chapters of various computing education topics such as the history of computing education, learner difficulties and misconceptions in programming, research methods, pedagogic approaches, tools and learning environments for programming.

The courses in Educational Technology will have a major technology development component, whereby students will work in groups and develop a prototype of an educational tool or learning environment based on their interests. These courses can be electives which can be useful for students interested in working in EdTech companies.

I am excited to teach the educational technology and computing education courses, as I hope that these courses will inspire graduate (and even undergraduate) students towards doing research in the applied CS areas of computing education and educational technology.