Teaching Statement

Minseok Jeon

As an educator, my primary goal in teaching is to motivate students to use their acquired knowledge to solve their problems. Teaching is the most important academic activity that has a great impact not only on students but also on instructors, teaching assistants, and academia. In a course, students learn existing knowledge, and each of their understandings is assessed by exams or assignments. However, I believe that the goal of teaching is not just to transmit existing knowledge but to enable the next generation to use the acquired knowledge or create new knowledge to solve their problems. I have experienced that well-designed course materials play a pivotal role in achieving this goal. In particular, I will enable students to use programming language technologies to address their own problems.

1 Teaching Experience

When I was a teaching assistant (TA) in a programming language course, I designed a system that allowed students to improve it with their ideas. In the course, the students' grades were mainly determined by their programming assignments. Students submitted their codes, and the submissions were graded by scoring test cases. Because of eht characteristics of programming assignments, small mistakes (.e,g,m causing a compile error) often resulted in a poor score (e.g., 0). Students who made such mistakes were often deeply disappointed and requested opportunities to rescue their programs by fixing the minor errors. As I sympathized with the students and wanted to give them a chance, I designed a system named Rescue that automatically classifies whether a given code change is a minor revision that corrects only small mistakes. The students resubmitted their revised code, and I accepted the resubmitted code if our system classified the given code change as a minor revision.

A few weeks later, the students started to improve the system. The students proposed innovative ideas to improve the system and discussed its vulnerabilities. I would like to note that the system was an improvised one; improving or analyzing the system was not the main content of the course. The students, however, enjoyed using their knowledge to enhance or analyze the system. For example, some students leveraged their programming language skills, which they had learned in the course, to improve the system to capture minor revisions that corrected typos, which had not been possible before their idea. Other students used their security skills to analyze the system's vulnerabilities and proposed valuable ideas to remove them. I was truly impressed by the proposed ideas and actively discussed and applied them to the system. I believe that such activities are the essence of education, enabling students to leverage their knowledge and talents to address their problems. This experience has led me to believe that providing interesting content that motivates the students is the most important part of teaching.

2 Teaching Interests

My research experience covers a wide range of topics in programming languages and software engineering. I have conducted various research projects related to static program analysis. Additionally, I have developed machine learning methods based on programming language techniques. I want to teach undergraduate and graduate courses related to my research topics. This way, I can provide interesting content related to my research for students. Specifically, I want to teach *Programming Languages*, *Software Engineering*, *Compiler*, *Theory of Computation*, and *Program verification*. Also, if possible, I would like to develop a new course, *Program Synthesis*, that combines programming languages and machine learning techniques.

Programming Language. In this course, I will teach the essential concepts of programming languages. Students will first learn the basic concepts of programming languages, such as syntax and semantics. Then, students will learn advanced concepts such as program analysis and type systems. Students will implement an interpreter and a type system for small programming languages to understand these concepts. Additionally, I will provide content to motivate students to design and implement their own programming languages to solve their problems.

Software Engineering. This course aims to teach the process of developing software. Students will understand software requirements, software design, software development, software analysis, and software maintenance. I will introduce various recent research projects related to software engineering. A project will be assigned to the students, giving them the opportunity to enhance existing SE tools.

Compiler. This course aims to teach the key concepts and components of compilers. Students will learn how a high-level programming language is safely translated into a low-level programming language. Topics include lexical analysis, syntax analysis, semantic analysis, and optimization. Students will implement a compiler, including a lexer, parser, translator, and optimizer to understand these concepts. I will also introduce how static analysis can be used to optimize the compiler.

Theory of Computation. The goal of this course is to understand the Turing machine and computability. Students will first learn finite automata, regular expressions, and context-free grammar; they will understand their expressiveness and limitations. Then, students will learn about the Turing machine, which is a foundation of computer science. Finally, students will learn about undecidability.

Program Analysis. The goal of this course is to understand the basic concepts of program analysis and learn how to design program analysis tools. Students will learn various program analysis techniques, such as static analysis, dynamic analysis, and symbolic execution. They will understand how these techniques work. As assignments, students will implement program analysis tools and learn how to improve them. I will introduce how program analysis techniques are actually used in industry and applied to various fields. Students will also have a chance to improve the program analysis tools with their ideas.

Program Synthesis. The goal of this course is to learn how to design a domain-specific programming language for a given problem and how to synthesize the correct program from the designed language. I will introduce various practical domain-specific languages that are being researched and developed in academia and industry. Students will then learn how to synthesize the correct program from the designed language. As assignments, students will develop domain-specific programming languages and synthesis algorithms to address their own problems.

3 Advising

I believe that each student has a unique research topic, and the most important challenge in advising is exploring the talents and interests of the students. During my Ph.D. course, I advised two students, and they successfully published their projects as research papers. Currently, I am also advising three undergraduate students. When I was advising them, I observed that each student has different talents and interests. For example, one student is interested in the C++ language and has a talent for implementation. I introduced a topic related to analyzing C++ programs, and the student successfully finished the project. This work will be submitted to the premier software engineering conference ICSE 2025. Advising should aim to find and maximize such talent and interest. This will lead to outstanding results. To achieve this, I will have regular meetings with the students to establish effective communication and discussion. I will communicate with each student at least once a week to guide and discuss their research. I will also establish regular seminars to enable the students to present their ideas and discuss them with other students.