# Michael Carbin *Teaching Statement*

Teaching and advising are two of the most important activities of academia. I am excited by the opportunity to be a professor because it holds out the promise and privilege of engaging students in these activities.

## Teaching Approach

My teaching approach engages students' natural creativity with a challenging mix of theory and practice. I teach theoretical concepts with a commitment to connecting these concepts with concrete manipulable examples. For example, as a teaching assistant for *Computer Language Engineering* at MIT, I taught dataflow optimizations by explaining the effect of an optimization on a concrete program. I found that this approach helped students understand a concept's mechanics and also understand how the concept could be extended and applied elsewhere.

I also emphasize putting theory to practice by favoring large full-semester projects over multiple small disconnected projects. From my experience as a teaching assistant, I observed that giving large projects serves three purposes. The first purpose is to motivate students to learn all the material for the semester. I found that because future project components depended on past components, students often revisited old concepts as they reimplemented old parts of their projects throughout the semester. The second purpose is that a large project gives students their own world in which to map theoretical concepts. I found that students often arrived at justifications for concepts that were particularly suited to their own focus and talents as implemented in their project. The third purpose is that a project inspires a unique dedication to the class. I found that students rallied behind the shared goal of producing one of the largest systems that they had ever built. This created a positive environment in which students were excited by, motivated by, and proud of their work.

I also have a strong passion for helping students who are dedicated, but not necessarily performing well in class. As a teaching assistant, I found that while I was excited to work with high achieving students, it was also challenging and engaging to interact with students who found the material challenging. These are the students with which I found myself giving spontaneous lectures that distilled class material to its core concepts. The more I worked with these students, the more I identified that finding the elegance of an idea was a key skill that young students needed to be successful in the classroom.

## Lecturing Approach

I organize all of my lectures around a storyline. Each storyline starts with a fundamental research or teaching question (for example, "How do we verify a program?") and ends in a concrete example (for example, "How do we use Separation Logic to verify a linked list implementation?"). I teach to this storyline by taking moments to prompt students with the questions they should be thinking about as the storyline progresses. This approach helps students engage with the lecture material and also helps me understand how well students are digesting the concepts. As a teaching assistant, students connected with my approach; in their evaluations they reported that my lectures were clear, knowledgeable, and accessible.

## Teaching Interests

My teaching, research, and work experience has covered a wide variety of topics, including theory, operating systems, graphics, machine learning, computer architecture, and programming systems. Given the need, I am qualified to and would readily commit the effort required to effectively teach undergraduate classes in these subjects.

Because of the scope of my current research, I am specifically eager and excited to teach programming systems and software engineering classes at both the undergraduate and graduate levels. For example, I am particularly interested in teaching *Systems Programming*, *Programing Languages* (formal semantics and type systems), *Compilers*, *Programming Language Design*, *Software Engineering*, and *Program Analysis and Verification* (among others).

I am also interested in incorporating interactive theorem proving into my teaching approach. For example, I would enjoy teaching a programming languages course that pairs learning type systems with an interactive theorem prover. Using an interactive theorem prover will enable students to concretely build and execute type soundness proofs just as how I observed that they optimize and execute programs.

## Advising Approach

I am always eager to volunteer for opportunities to mentor students. I have helped advise two masters theses and I have formally and informally advised multiple high school students (two), undergraduate students (five), and younger graduate students (five) while at MIT. One of the most challenging aspects of mentoring has been identifying what kind of interaction works best with each student. Another challenge has been identifying how frequently to interact with a student; some students benefit from more hands-on encouragement than others. My approach has been to make myself readily available to students and make it clear that if they need something then I am available. To make sure that students are on track, I also encourage frequent casual meetings and occasional written proposals for new ideas and directions. I have found that giving thorough feedback on a student's proposal is one of the best ways to steer a project.

## Perspective

I am excited by the opportunity to teach Computer Science because Computer Science is the language of computation. Computation – whether it be as simple as manipulating personal data in a spreadsheet or as complicated as compiling a program – is something that anyone can benefit from. In the future, there will therefore be an incentive to teach everyone how to program. As both a researcher and teacher, I will have the unique opportunity to shape both the languages and curriculum of this future.