I believe that learning is, at the most fundamental level, a process of forming connections — connections that bridge the gaps between prior and new knowledge, between theory and application, between classes of problems and solution strategies. This process is dynamic and ongoing: just as athletes must regularly exercise their muscles in order to spark physiological adaptations that lead to increased strength, learners must continuously apply their knowledge and skills in order to catalyze the cognitive adaptations that enable them to achieve higher levels of mastery.

Within the context of computer science, I believe that the single most important adaptation is a change in the way that a student tackles problems. Successful computer scientists approach problems in a uniquely methodical way. They break down complex problems into logical sub-components, analyze tradeoffs in time and space, and design and implement elegant solutions. This mindset comes not from acquiring rote knowledge such as familiarity with a specific programming language, but from developing higher-order skills including creativity, critical thinking, and logical reasoning. With creativity, computer scientists can examine a multifaceted problem from all angles and imagine a diverse set of possible solutions. With critical thinking, they can assess the relative merits of these solutions on axes such as efficiency, flexibility, extensibility, and scalability. With logical reasoning, they can implement their chosen solution in a methodical way that leverages the power of computers.

As an educator, I believe that I can best support students as they develop these higher-order skills by influencing their attitudes, environments, and opportunities. Specifically, I believe in inspiring students to learn by providing motivation, enabling students to learn by creating a supportive and inclusive environment, and helping students to learn by constructing opportunities that actively engage them in application.

1. **Attitudes: Building Motivation through Context, Relevance, and Self-Efficacy**

   "Why should I care?" A simple question, but one whose answer can have a profound effect on motivation — and, by extension, on learning. Learning requires persistent effort, and the more motivated students are, the more excited they will be to put in that effort. Thus, I answer this question with context and relevance. First, I demonstrate how specific concepts further the broad underlying principles of computer science, and simultaneously emphasize that these underlying principles are paramount to success in the field. For example, when I introduce functions I explain how they can be used to abstract away complexity, resulting in a more readable, flexible, and maintainable solution. Second, I offer multiple real-world examples of theoretical concepts in action. For instance, when I introduce graphs, I show how they can model various structures and relationships ranging from airline routes to the spread of diseases. Later, I task students with using graphs to represent social networks and maps. In addition to providing motivation, such examples underscore the transferability of these concepts, better equipping students to apply them in new contexts. I also make it a priority to learn each individual student’s interests and goals, so that I can connect the material to their interests and show how it can help them achieve these goals.

Once students recognize the value of the material, they may still face other barriers such as low self-efficacy. I believe that self-efficacy has a huge impact on motivation, success, and happiness: it inspires students to attack challenges head on, makes them resilient to failure, and ultimately makes them stronger learners. Unfortunately, impostor syndrome is particularly prevalent in computer science due to the wide spectrum of backgrounds with which students enter the major. While some enter with the confidence of years of experience, many have never written a single line of code and immediately feel overwhelmed. In introductory courses, I combat this by emphasizing that the heart of the field is not programming but problem-solving. Code is simply a means of conveying a solution to a computer, and any gaps due to programming experience will quickly vanish; the true key to success lies in the creative design process.

To further build self-efficacy, I set high bars for my students but make it clear that my expectations are rooted in faith in their capability. When students seek my help, I am careful to guide them in a way that does not stifle their growth, but rather equips them to solve their own problems. For instance, I may ask probing questions to identify underlying misconceptions, sketch pictures and examples to clarify key concepts, or help them design experiments to diagnose faulty logic. Ultimately, my students consistently rise to the challenge, and when they do I can honestly affirm that their success was not because the class
Rebecca Smith

Teaching Statement

was easy, or because they leaned on me for support, but was a direct result of the hard work that they put in. This leaves them with a genuine sense of accomplishment that motivates them to persist in the face of future challenges.

2. Environment: Fostering a Supportive and Inclusive Community

Just as I strive to foster positive attitudes within my students, I also strive to create a positive learning environment around my students. A positive learning environment draws students in and allows them to thrive; to enable this, the environment must first be supportive. I believe that a supportive environment is characterized not by an absence of failure, but by a sense of security that empowers risk-taking. Such an environment enables students to speak their minds and inspires them to respond to failure with perseverance rather than defeat. To this end, I show my students that I care about them both as learners and as people by taking the time to understand their individual strengths, weaknesses, goals, and prior knowledge. This familiarity builds a crucial level of rapport and comfort, while also enabling me to better tailor my pedagogy to each individual. Further, I frame uncertainty and mistakes as valuable parts of the learning process. I assure students that there are no “stupid questions”, and seek to highlight positives even as I offer constructive feedback.

Second, the environment must be inclusive of diverse backgrounds and perspectives. As a female computer scientist, I am particularly cognizant of the many subtle ways that members of underrepresented groups become discouraged from continuing in the field, and work to eliminate these social and psychological barriers. First, I seek to broaden students’ view of the field, so that students of all backgrounds can self-identify as computer scientists. Pervasive stereotypes frequently act as deterrents, robbing individuals of opportunities and robbing the field of rich and diverse perspectives. I actively break down stereotypes by highlighting pioneers and leaders in the field who come from underrepresented backgrounds, painting computer scientists as problem-solvers who work in myriad domains such as art, music, biology, and economics, and noting that this problem-solving process typically involves teamwork and communication. Second, when I facilitate discussions, I emphasize that all ideas are worthy of consideration, and that melding diverse ideas ultimately leads to a better solution. Last, I take special care to engage, mentor, and encourage members of underrepresented groups, both at the undergraduate level and through community outreach.

3. Opportunities: Creating a Cycle of Application and Feedback

Computer science is an inherently applied field. Thus, when designing assignments and assessments, I prioritize exercises that require design, analysis, and application. I believe that class time should be used not only to introduce theoretical concepts, but to actively engage students in application coupled with two-way feedback. Practicing application outside the classroom is valuable but insufficient. Students benefit greatly from receiving feedback in a low-stakes environment; giving feedback early and often ensures that they students understand my expectations well before facing summative assessments, and have plenty of time to adapt their approach. Since I value not only the end result but the problem-solving process, it’s especially essential that I see students in action so that I can evaluate this process. To that end, I construct active learning exercises to engage students in targeted practice. As the students work, I ask questions about their solutions and require them to justify their design decisions. This forces them to engage in valuable metacognition and evaluate explicit or implicit tradeoffs they may have made.

Additionally, I believe it’s crucial to create regular opportunities for students to provide me with feedback. To assess whether my pedagogical methods are effectively helping my students achieve the learning goals, I seek formal feedback in the form of mid-semester surveys. However, I also use everyday interactions with students to gain insight into their current level of mastery, enabling me to adapt lessons as needed. Even when I use a traditional lecture format, I pepper my lessons with low-stakes assessments to gauge whether students have absorbed the material, followed by questions which require them to apply it in new contexts. My summative assessments likewise reflect the value that I place on design and application. Given time constraints, I rarely ask students to write code on exams; instead, I will often ask them to design data structures or sketch pseudocode. This allows them to demonstrate their higher-order problem-solving skills without getting mired in more rote skills such as syntax and debugging.
In conclusion, I believe that excellent teaching is characterized by its ability to spark “excellent” learning: learning that propels students to higher levels of mastery and enables them to achieve their potential. While this learning ultimately occurs in the form of cognitive connections, the techniques I use to incite it are uniformly strengthened by interpersonal connections. Connecting with students on a personal level — understanding what drives and inspires them — better equips me to motivate, support, and engage them. Further, I believe that the value of such interpersonal connections extends far beyond the classroom, so I strive to build long-term mentorship relationships. My own mentors have changed the course of my life for the better; I want to likewise help my students discover paths that are meaningful and personally-fulfilling. I encourage students to meet with me one-on-one, and countless have taken me up on this offer. Many return year after year to update me on their lives and to seek advice on everything from course selection and internship applications to career paths and grad school. Recently, one such student joked that she’ll probably call me up in 40 years asking for advice on retirement, and frankly, I’d be thrilled if she did.