My high school did not have any programming courses. Outside of school, I had no role model encouraging me to learn to program, so I did not write my first program until college. I fell into computer science by chance, as a supporting sequence for my math major. My advisor convinced me to take the introductory computer science course for majors. She said that it would be challenging for a non-major, and it was. Everyone around me already knew what they were doing. During the first project, the professor told us that we weren't allowed to use arrays. The entire class let out a collective groan, except for me. I had no idea what an array was.

At that point, I lacked confidence. I hadn't been a good high school student to begin with, and being surrounded by freshmen who already knew the material was frightening. I worked hard because I wanted to do well. I learned how I process information and how to study to optimize this. My grades began to improve, but I still did not believe in myself.

The summer after my freshman year changed me. I spent two weeks at the Google Computer Science Summer Institute, a program which aimed to increase the pipeline of women and minorities into computer science. There, we built a web application as a team. For the first time, nobody doubted me. I internalized the confidence others had in me. I thought of the application idea and worked with a teammate on the design. We came together to help the other teammates with coding. At the end, we presented the application to a team of Google employees.

The confidence carried on when I returned to school. I chose computer science as a second major. I began to excel in my classes. The experience formed the backbone of my desire to instill confidence in other women, minorities, and students who were late to the field. By believing in others, I can help them believe in themselves and succeed.

This fall, I am beginning my graduate studies at the University of Washington. My goal is to become a professor. As a professor, I will be a researcher, a teacher, and a mentor. I will further knowledge in the field I've grown to love. I will instill confidence in others and convince them that they are capable. Then I will continue the cycle by leading others to believe in themselves, to further their own studies, and to contribute their own ideas.

Research – Intellectual Merit – In my third year of college, I did research with Kasso Okoudjou in mathematics on the properties of tight frames. A tight frame is a generalization of a basis of a vector space with equal bounds. Once per week, I met with the professor to discuss the problem. On my own, I read background papers. I wrote MATLAB functions to observe patterns in the properties of tight frames, and then worked to formalize those patterns. The professor and I had different, harmonious ways of thinking. He approached problems visually, while I approached the same problems analytically. We were able to fill in holes in each other's thinking. The work that we did together inspired a publication several years later [1].

My senior year, I did cryptology research with Larry Washington. Using elliptic curves, I designed a protocol that allows two groups to communicate only if enough people are present. This is known as a threshold scheme. The scheme allowed for a different threshold of group members for each group. I read papers on my own, thought through the problem on my own, and met with the professor weekly to discuss the problem. As far as we were aware, this was the first elliptic curve threshold scheme, giving it an efficiency advantage over its predecessors. It used a bilinear pairing to establish a secret key that could be used by the two groups to communicate. Furthermore, it allowed for individual shares and group secret keys to be reused in subsequent sessions. The research culminated in my honors thesis [2].

From the start of my studies, I gravitated toward open problems. I spent my free time after class trying to solve famous problems like P vs. NP and the complexity of the graph

isomorphism problem. As an undergraduate researcher, I extended this love beyond famous problems. I learned that I enjoy every end of the research process, from background reading to clarifying the question to writing a paper. I developed my skills in tackling large, open problems.

Teaching – Broader Impacts – My sophomore year, I worked as a tutor for student-athletes. I met individually with students to help them with mathematics and computer science courses. My students were non-majors who found the courses challenging. I helped them improve their grades and even begin to enjoy the work. I will never forget the look of awe from one of my students when she saw her first website working for the first time. I reached students who otherwise weren't drawn to STEM fields. I began to recognize my love of teaching.

Senior year, I was a teaching assistant for Computer and Network Security. I graded projects, held office hours, and answered questions. I used my experience as a non-traditional learner to empathize with the way different students learn. When my students did not understand something, I framed it in a different way. When I found the optimal way to frame a topic for these students, they understood, and their excitement was rewarding. Students from both sections began to come to my office hours. I helped many students succeed in the course.

Becoming a TA is the best thing I have ever done for myself. It fostered not only my love of teaching, but also my confidence. To be confident enough to teach others, I had to trust myself. The lessons I learned carried over outside of teaching. Whenever a friend, classmate, or coworker does not understand something, I adjust my explanation and watch him glow. I will tap into these skills as I continue to develop as a teacher during graduate school.

Industry – Intellectual Merit & Broader Impacts – During college, I interned with two software companies. When I graduated from college, I returned to Amazon. I worked with a team of 5-10 engineers. We broke down large, vague tasks into individual tasks. I interacted with many other teams and made large contributions both to customer-facing code and to internal frameworks. My work is reached hundreds of thousands of times per day, but has low operational load. A recent change I made to the navigation bar impacted thirty-six different teams, but caused no major operational issues.

At Amazon, I gave a talk on FindBugs, a tool that uses program analysis to identify bugs in programs. It is widely used within Amazon. The talk inspired a coworker to compile the data of analysis tools run on our code into an internal website where we could view the results. This led to an increase in code quality. The experience helped prepare me for the presentations I will give as a graduate student.

The next year, I applied data flow analysis to solve a problem we faced with one of our internal frameworks. In parallel, a coworker developed annotations. Together, our changes allowed for contract-style programming. The analysis checked that the pre-conditions and post-conditions were logically sound at compile-time. This increased code readability, decreased run-time errors, and increased programmer productivity. It was my first experience with a self-directed project. It prepared me to drive my own research.

My experience in industry has given me insight into the real-world applications of my work. It has motivated me to solve problems at their roots, that way others can apply my solutions to societal problems. In college, I was interested primarily in theoretical computer science. Working in industry drew me toward programming languages research, which is backed by interesting theoretical concepts, but at the same time has immediate effects on real-world problems. My experience prepared me to focus on programming languages in graduate school.

Mentorship – Broader Impacts – During my three years as a software engineer at Amazon, I mentored several engineers individually. On my team, I served as a technical mentor for an intern. I discussed project work and progress with him. I helped him understand Amazon tools and technologies. Instead of solving problems for him, I gave him suggestions that he could apply himself. His code improved and he grew as an engineer over the course of the internship. Consequently, he received a return offer.

Outside of my team, I worked as a career mentor for three female engineering interns. The return offer rate for female software engineering interns at Amazon has historically been lower than the return offer rate for male software engineering interns. Being a female software engineer comes with its own unique set of challenges. Other engineers assumed I was the recruiter during interview debriefs, or that I was not an engineer in technical meetings. They spoke to me non-technically and ignored my ideas. I had to work hard not to internalize the notion that I was only there because I was a woman.

All three female interns whom I mentored received return offers. I saw an increase in their confidence and was able to prevent some of the issues I had experienced. I offered simple ways to cope with the difficulties that come with being a female software engineer. I gave career advice that applies to all engineers, regardless of gender. I encouraged my interns to believe in themselves. An incoming sophomore told me that she felt inadequate because everyone else on her team was a graduate student. I told her, “on the contrary, the fact that you are here means that you are extraordinary.” I encouraged her to trust herself. I made a difference not only in the return offer rate, but also in the lives of these young engineers.

I continue to mentor outside of Amazon. This fall, I am starting to work as a mentor for the TUNE House, a living-and-learning community for undergraduate women in computer science. Throughout the school year, I will meet with an undergraduate student to help her with her education and career. I plan to use my experience to help this young woman prosper.

Future Goals – I have always been drawn to solving open problems, and I am excited to continue to develop this skill. The University of Washington provides the ideal environment for me to succeed. It has a collaborative culture which encourages discussion between different research areas. Rather than discard my theoretical background to work on programming languages, I will use it to drive new results. I will work on the concepts I'm drawn to while having an immediate impact on the real-world problems I have experienced in industry.

At UW, I will continue to grow as a researcher, as a teacher, and as a mentor. I will continue to become more independent and drive not only the solutions to my research, but also the questions. I will publish my work and present it, using the presentation skills I've gained in industry. I will build upon my teaching skills and I will continue to reach out as a mentor.

When I am done with school, I plan to be a professor. I will continue to take advantage of my theoretical interests to explore problems that lie at the intersection between theory and programming languages. At the same time, I will be a positive role model. I will use my background to help minority students and non-traditional learners. I will arm my students with confidence early. I will encourage them to grow, learn, give, and continue the cycle.

---

1. Lemvig, J., Miller, C., and Okoudjou, K. Prime Tight Frames. *Advances in Computational Mathematics*, 40(2):315 – 334.

2. Ringer, T. An Elliptic Curve Threshold Key Establishment Scheme. Retrieved September 10, 2015 from University of Maryland: http://honors.cs.umd.edu/reports/ringer.pdf.