There are two steps to writing useful software: learn to bake cookies, then take up sailing. At least that's how I do it.

…

Cookies are great, and I say that as someone who generally can't eat them. I was born allergic to milk, egg, and nuts, so there aren't many cookies out there that I can have. Luckily, my mom has spent the last 19-some years refining her Eric-safe recipes to make up for it. Now they're all excellent, so I came into UW eager to impress everyone with my vegan (no milk, no egg) baking skills. I like baking so much because it's probably the most efficient way to do something worthwhile: just by following a recipe. In truth, baking is quite formulaic; if you combine the right ingredients in the right proportions and mix them well, the oven will do the rest, yielding perfect cookies every time. That's how I learned it's good to follow best practices when making something.

I say that cookie-baking is useful in the sense that I can easily churn out a product that people will like (myself included). It's also fun to be a perfectionist when making them. Since arriving on campus, I've tweaked some recipes to use AirBake pans, and learned to add more cinnamon to my snickerdoodles. Ensuring a product is perfect gives consumers the best experience, after all.

Beyond iterating through cookie recipes, I've also picked up sailing on campus. At first, I just wanted to figure out how to power a boat with nothing but a large triangle or two of fabric. Now that I have the basics down, I'm hoping to get good enough to take my friends and girlfriend out in a boat that's a bit larger. It would be super cool if others could enjoy my (newfound) skill too.

I liked the club's sailing lessons so much that I attended a meeting back on land. I thought maybe I could get involved with this cool club by learning to repair torn sails or something. Instead, when I introduced myself to one of the club's commodores, our term for presidents, he asked me what I wanted to study. Once I told him my intended major and experience, he offered me the position of computer consultant. Even though I'd promised myself that I'd do fun extracurriculars in college, when he explained that the role involved maintaining an aging monstrosity of a database frontend written in a combination of PHP and Perl, I jumped at the chance. I'm really glad I did, because I ended up with a really cool project on my hands: rewriting the ancient database app. Another club member named Thomas, who is a software dev for a local healthcare company, is helping provide ideas and guidance. So far, I have a working prototype off the ground, and I’m planning to finish it by the end of the year. Having past experience, especially working with others, is always good for software developers; just ask any employer.

…

If you ask anyone other than me, it would seem like I have a hodgepodge of weird interests. For example, I’ve taken up volunteering as a mentor at the Comotion makerspace, where I teach others to sew (a skill I learned in high school), and they teach me to use laser cutters, 3D printers, and CNC mills. Actually though, I just like to make useful things. Similarly to how I want to 3D print cup racks to better use my limited dorm shelf space, I want to study computer science because it lets me make life easier for everyone. For example, a better webapp for the sailing club means that the club won’t have to waste time dealing with database problems and can focus on sailing.

Like cookies, software should make life better. Like sailing, people should enjoy it. I’m hoping for the chance to make that a reality. As a freshman with lots of AP credit, I’m itching to start because I’ve almost run out of non-major classes to take. I want to major in CS because I’m sure it will help me build useful tools for myself, the clubs and organizations I work with, and people in general.