Dear Portland Code School,

I’ve been working in tech for almost my entire career, starting in high school when I was lucky enough to get a job testing computer software after school. Even when I wasn’t working in tech, it found me. I wrote plenty of college papers on technology, I worked for the Electronic Frontier Foundation, and I helped bring libraries into this century.

Now that I once again work in a “real” tech job, I have been taking note of days when I walk away from my desk exhausted and completely happy with the work I have done. Increasingly, those days are ones I have spent working on custom code for a website, wrangling html emails, or generally pushing bits around. In short, I like building things.

The problem is I don't always understand why what I’m building works. I can copy and paste as good as any other Googler, but if it doesn't work right I have a hard time fixing it. I spend way too much time changing random variables in the hopes that it will fix the problem, and then I try and learn from my successes. But that's not proving to be the best model of education for me. Part of what I would like to learn at Portland Code School is the "why" of code, in addition to the "how."

And, of course, getting a new job after completing the program wouldn't be half bad. Then I could spend more days coding and fewer days cursing the latest changes to the Facebook advertising regulations.

I could offer some general phrases that describe how I work with teams: I like to lead, but I don’t have to. I like to give input, but I my idea isn’t always the best. I like to work alone on discrete tasks, but creative planning is best tackled by a group. I think this is pretty general stuff. So let me tell you about the creative strategy meeting.

I love my department’s weekly, hour-long creative strategy meetings. I think it has a lot to do with the way I like working with a team. We spend a set amount of time throwing ideas around - any ideas, crazy ideas - for a project. And then the leader of that project steps back, organizes things, and returns to the team with a gameplan.

So what’s so great about these meetings? They give everybody a voice, even on projects where they might not be part of the team. They give the project leader the ability to gather ideas, even if she might not use them later. And they always lead to questions that one person alone wouldn’t have thought to ask, let alone answer.

In summation, I like to build things, I want to learn how software works, and I want to do both with like-minded people. And that’s why I want to attend Portland Code School.

Thanks,
Sara LaFassett