# Personal Reflections

## How has your knowledge of the tech field changed since class began?

The truth is I had basically no knowledge about the tech field itself coming into the class. (I knew a little about programming, but not about what it is like to work in the tech field.) The first week, even though I thought some of the information could’ve been condensed so we had more time to actually learn coding, I thought was helpful to talk about Agile, the SDLC, and others things to help give context what working at a software company is like. This definitely helped later on, when we learned about CircleCI, and other aspects (although I figured those could be condensed as well..) to give context why certain processes and tools are used on a software team. My initial thought is, “if I’m working on a project alone, this is just unnecessary steps”, but understanding the SDLC and the way teams are typically made up, helps understand why a bunch of this stuff is important.

## What was the most difficult part of Foundations for you? And how did you overcome that?

I'd say the most difficult part was the week of the computer science. Part because it was very high concept, and was hard to put into context of how to use most of it to become a better programmer. Also, part because it was toward the end of the course, and it felt like a jarring change from coding every day to very little of it.

## Did you learn anything about the way you learn? What was it?

I was pretty aware that I like to ask a lot of questions, that address concepts to help me think about coding rather than memorization, before I joined. However, I was probably less aware of how much I not only like explaining and talking about my code, but how much walking through my code (or even someone else's if I'm helping them), and having to figure out how to explain it in terms that make sense to another person, out loud, actually helps me understand it even better than before, on a foundational level.

## How will you apply what you know about your learning style during Specializations?

Hmmm I don't really know. I mean I'm kind of anticipating not being able to ask questions and get an immediate response (if its all through video) having an impact on how quickly I pick something up. I'll just have to adjust, and probably use the SMEs more often than I used the queue in foundations.

## What was your experience with pair programming like?

I mean I really like working with everyone in the group. It was great to be able to get to know them better. I'm not really sure using pair programming during our daily labs was a great example of what pair programming actually is, because we all were working on separate assignments, but we were able to use each other as a knowledge source.

If we had some sort of group project, possibly led by a mentor or instructor so we had some guidance and a defined leader who could give us an objective, we could have had the opportunity to actually practice one style of Agile, practice daily standups, practice working on a Github project with more than one person where we have to deal with conflicting code, and then possibly find a way to work pair programming in there, where more than one person can work on the exact same piece of code with the exact same objective.

## What parts of Foundations did you enjoy the most? (This could be anything from a specific lab or lecture to the general setup of class or a funny moment.)

Definitely having a group environment where we try to simulate a classroom with cameras and consistent verbal and visually communication makes it a more enjoyable experience for me. I definitely don't think I would have enjoyed it or even learned as much if everyone had their cameras off and we never spoke up. Then it might as well be a video playing in my opinion.

It also helped everyone get to know each other better, I think. This was also reflected in our final four weeks of morning squads, at least with Stu the last four weeks, where we all could figure out what structure would benefit everyone the most that morning. Somedays we used the time to do Codewars, and would talk through our code after, somedays we walked through a specific problem one person was having with a project as a group, one day we just got extra time to work on our projects if that's what helped everyone the most. That flexibility was helpful, and the fact that we all got to know each other, in my opinion, made it easier to do that effectively, and not have a cut/paste structure to every morning that is unchangeable, which would've been less helpful, I think.

The first four or five weeks or so were also probably the most enjoyable in my opinion, since (even though I had a small amount of code knowledge) we were able to dive deep into some of the fundamentals, and it helped clarify in my mind the flow of an app, the differences between front-end, back-end, and databases. That part where it is all knew and we learned so much in just a few weeks, is a really enjoyable part.