1. How well-motivated do you find Rust to be? How well have its designers communicated a purpose, justified the importance of that purpose, and followed through on living up to potential?

I'll admit the at first I was a little skeptical about this course because Rust was a completely new language for me. I was a little bit of hard time at first but I think I do find myself pretty motivated to code in Rust. I think the designers have clearly communicated its purpose that is to provide memory safety and concurrency without sacrificing performance.

2. How does your experience of Rust tooling (such as the package manager, compiler, and VSCode plugin) compare to your experience of other tools? High-effort/low-effort? High-error/low-error?

I'm not gonna lie, compared to other languages, Rust's tools feels more beginner friendly despite it's complexity. The compiler is super helpful and the package manager is smooth and intuitive.

3. How is your experience of the social infrastructure of Rust? E.g. Documentation, tutorials, StackOverflow questions and answers.

Rust's social infrastructure is very supportive and beginner friendly. The official documentation is very clear and helpful and there plenty of high quality tutorials on Youtube. The forums like the Stackoverflow and Reddit page of rust user can be very helpful as well.

4. In the second half of the term there will be lighter focus on implementation and more emphasis on design, human-centered computing, and application. What are you most interested in learning about in the second half of the term, and is there anything you're interested in learning about that isn't listed in the syllabus?

I am interested and looking to the lectures on gender and disability. To understand these topics through a lens of a programming language would be really really interesting.