In my time in Creative Coding, I have learned multiple new skills to further displaying my computer science skills. Initially, the syntax of both languages were pretty straightforward as it was nothing that I haven't seen before. Because of this, I barely spent time focusing on syntax taught in class and would just give my "best" shot at completing the exercises. Moving into the midway point of the semester, I had completely turned the habits on its head. I spent the majority of my time learning all the syntax I neglected early in the semester and I little less time just sitting in front of my software wondering how to do what. Because of this I feel I set myself off better for the later of the class and ultimately made my final project easier.

I feel as though I have learned a great deal from this class, like every language I have been taught in this school, complications and errors happen all the time. Thankfully, both processing and p5.js have an insane amount of online resources to help learn the language so I would say frustration was lower in general compared to other coding classes I have taken. For example, coming into my final project that had to do a lot with vector graphics was alarming in the sense that I did not even have a strong grasp of how they worked. I would say my project idea forced my hand in learning as much as I could, or else my code would be very mediocre.

Although initially any coder would say procedural programming is preferred, it is not the best practice. Often time, code is re-used in a single execution. Multiple instances of the same thing are required. With the inclusion of OOP it simplifies things, and does remedy the problem of re-using/re-writing code. Essentially, procedural programming looks easier to understand, OOP adds in the efficiency.

Looking back at my final project, I don't remember a step in the construction that didn't have a bug. Again, p5 and processing does so much for its users in terms of debugging that syntax and functionality is explained to you at every hiccup. Compile time errors were probably the only issue that wasn't easily solvable but usually it is because something is missing. In this instance debugging for me just involved scanning over my code and noticing what was out of place. Lastly moving forward, I see computer science as a big part of my life now. It is my major and the field in which I want to continue in the future. As a computer science major, I trust that I will continue being put in situations where I have to learn new languages and practices, So I will be doing this for a while. The rest of my classes in this school are all CS and I am glad to have experienced this side of the coding world and will make sure to keep these skills close.