In 3+ paragraphs, highlight your lessons learned and learning experience from working on this project. What have you learned? What did you struggle with? What would you do differently on your next project?

During the course of this second project, the three things that I learned most are exception handling, algorithm implementation, and the general functions of stacks and queues. After this project and last one, I have a far better grasp on making custom exception classes to be thrown in the case of an error. I also learned how to implement intermediate algorithms. The algorithms were given in the project explanation document, so after reading and understanding what they did I generally tried to implement them myself. This helped me with learning to understand pseudocode and being able to implement it accordingly. I also learned what stacks and queues are and how to use them for this specific situation, but also got me thinking on how these data structures can be used elsewhere. Stacks and queues are restricted data structures that are first in last out and first in first out respectively.

The two things that I struggled most with was implementing the algorithm of the infixToPostfix notation method and implementing my own generic Queue. The infixToPostfix notation algorithm was rather difficult to understand and I’m not sure I quite get it even still, as it combined both stacks and queues to convert infix to postfix. Implementing my own generic queue was difficult because it was the first out of the two restricted data structures that I tried to make, but after I made the queue making the stack was rather easy and quick.

On my next project, what I would do differently is try to really understand all of the algorithms, as when I was implementing the infixToPostfix algorithm I was almost rushing because I wasn’t sure I would be able to both understand it and implement it in time, only one of them. I would spend more time earlier to understand what needs to be done, and work accordingly.