During the creation of this Project, I had to re-read the sections In the OSP2 manual multiple times in order to fully understand the purpose of the assignment. The algorithms provided in the assignment were crucial to the implementation of the Project, and without them I truly would have been lost in attempting to implement these methods. I believe I was ultimately able to get the overall project working, as I have executed the program and processed the simulation multiple times without error.

My overall development process included adding the algorithmic design process from the assignment as well as the OSP2 manual’s more detailed information as comments into each method, and then during implementation of the method, turn these extensive comments into the code and ‘normal’ comment structure of the program. This method worked well for me, although I did miss, at first, the instructions for preempting a thread and had to refer to this after discovering errors. During the further development of this project, I decided to use the recommended data structure: GenericList, which I labeled ReadyQ. The use of this data structure was easy to understand, although at one point I did have to use a typecast to ensure that the object returned was of the type ThreadCB. The algorithmic process described in the assignment was truly key to the overall development of this project.

The actual implementation of many methods utilized ‘this’ to refer to the thread being utilized, which was sometimes confusing but ultimately, I could grasp this technique. My first attempt at using placeholder newThreads in all methods was abandoned when I realized the amount of errors I was obtaining upon execution. During development, I had multiple times where I was given an ‘illegal attempt to create a new thread’ error, that was mostly resolved using \*this\* instead of creating a new thread copy each time. Also during development, I had multiple instances of NullPointerExceptions which I had to surround in try{}catch{} blocks to ensure proper execution. In the future, hopefully I can find a better way than to have try{}catch{} blocks in every method.

During initial development, I only utilized the OSP2 manual as well as the assignment itself to produce all the code. Upon attempting to resolve errors, especially NullPointerExceptions, I found particular use of a GitHub (<https://github.com/ankitrkumar/CIS657-Principles-of-Operating-Systems-OSP2>). I only utilized this code in order to clean up my existing code when it wasn’t functioning properly. Specifically, I changed the structure of my code in the do\_resume method, as my initial structure of two if statements were not catching some cases, and I was obtaining errors. I also utilized this code in order to assist in finding, when the error log was not particularly useful, areas of code to surround in a try catch block.

Attempting to debug this code is not the easiest, as the use of the log function is not always specific enough to find the particular error in question. However, after sadly some extensive practice, I do feel more comfortable with the debugging process as it relates to OSP2 projects.