The purpose of this project was to create a computer emulator that helps display the instruction cycle where the processors fetches instructions from memory and executes them. The processor can access data from memory through instructions and has registers to hold data as well. In this project, we had to code a processor that can understand a certain set of assembly language instructions and execute them. We had to read in text files that contained these instructions and match the output. We also had to create our own text files to show that our code works. This required us to recall and understand assembly language as well to get the created text file to work properly.

The project was implemented using two Java classes. One was called “CPU.java” and it had the code for the processor. The other was called “Memory.java” and it had the code for the memory process. I implemented the project similar to the Java examples that were provided. Two of the classes provided were “Proc.java” and “Proc2.java” which made the child processes ”Hello.java” and “Hello You.java.” My “CPU.java” class was similar to “Proc2.java” and “Proc.java” in that it made the child process “Memory.java.” In my memory class, I initialized the memory array by loading reading in the text file using a scanner and storing the instructions in an integer array of 2000 integers. If my memory class got requests to read or write I had to determine which one it was and perform the read or write. After initializing the array, the memory waits for a read or write from the processor. My CPU class had the processor and it used java Runtime exec to spawn the memory process to have two processes going on. I had to connect the input and output streams of memory as well to communicate between the processes. I used switch case to distinguish between all the instructions and to determine how to handle all of them. I also had a timer and interrupt handler in my CPU class.

For my personal experience, I had to do a lot of debugging. I learned that the errors you get don't tell the whole story. The errors I got suggested that there was a software issue, so I tried deleting and redownloading the Java JDKs and JREs but to no avail. It turned out that some of the errors were actually issues with my code that I figured out by debugging. Other errors were because I had a newer JDK and JRE on my personal computer. The memory class wasn’t too difficult. I did have to make changes after I started making the CPU class since I realized it wasn’t as simple as I made it out to be and that there was more to it. The CPU class was by far the most difficult class to implement. I had to figure out how java Runtime works so the java examples helped a lot. I also had to do a lot of debugging for the communication between the processes which was the hardest part. I got many Input Mismatch errors because I was reading in strings instead of integers on many occasions and I got a lot of No Such Element errors because I actually wasn’t sending anything to the CPU process in some cases. Debugging was harder than normal because the output of the memory process went to the CPU process. Usually, I use a print statement and print out to the console to figure out what’s going wrong. However, since the output of the memory process went to the CPU process, I couldn’t do that and had to figure out ways to work around that. Therefore, I felt like this project took me more time to debug than all my past projects combined. Overall, this project was not easy but I felt like it was beneficial and helped me learn a lot. I think this is complex enough to put on my resume and since I learned a lot, I can talk about it with recruiters.