The goal of this project is to use semaphores to create a simulation of a bank scenario with bank tellers, a loan officer, and many customers. I decided to use C++ to program this just because that is what I am most comfortable with. When I initially began, the I created my design for the program because the professor suggested we do that before coding. I began writing vague customer, loan officer, and bank teller functions to have each of their own threads to start in. I at first thought I needed to create my own semaphore functions because I didn’t read the word document carefully enough, but after asking the professor about it I quickly was able to get back on track and start using the POSIX semaphore and threads like we are supposed to. I learned how to initialize pthreads join pthreads, and how to use correct syntax for pthread parameters. The first problem I encountered was printing using “cout”. When I asked the professor about why my processes in my program were printing in between each other. He explained that the >> operator is multiple steps so one thread could be running and printing at the same time as another thread. He recommended I used printf instead because it was just one step. I started using printf and realized that I needed to be printing out the integers to show which customer and bank teller is communicating with each other. Processes communicating is the first big wall I ran into this project. The reason I think I needed this is because my customer thread needs to know which bank teller thread is helping it so it can print out the correct integer. I opened the book again to find out how to implement processes communicating with each other as well as how to implement queues to create first-in-first-out code for my semaphores. I found the book was hard to understand so I went to ask the professor if he could explain it to me again. The professor suggested I create a queue and an array to keep track of customer balances and bank teller numbers. I used this method and pretty much completed the rest of my program. Except for one problem which is that each customer does the exact same thing as the previous customer. I emailed the professor about this issue and was reminded that I need to run my program on cs1.utdallas.edu. After testing it there, my program seemed to work the intended way.