Project 2 - Write Up

1. **How did you approach the project?**
    a. First, I read through the requirements for the project, include what Teller and Customer supposed to do. Then follow the suggestions, I understand that this project is an interaction between the two, so I need to start working on both, because one cannot work on it own. I took a look at the ThreadDemo file to understand it better and how should I plan synchronization mechanisms using semaphores.
2. **How did you organize the project? Why?**
    a. I organized the project into four main classes
        i. **Bank**: This is like a driver class, it manages shared resources through semaphores and handle customer-teller assignments
        ii. **Customer**: I implemented customer thread behavior, transaction request, and handle interaction with tellers
        iii. **Teller**: I implemented teller thread behavior, managing resource access like safe and manager, and also handle customer service flow
        iv. **TransactionType**: I made a simple enum for transaction type, this is just for me to make sure the consistent transaction type handle.
3. **What problems did you encounter?**
    a. Customer's action announcements were appear in random order
    b. Multiple customers would sometimes get assigned to the same teller
    c. Customers sometime skip the line or get lost in the queue
    d. Teller ran into deadlock when trying to access both the safe and manager
    e. After finished serving 50 customers, teller didn't properly terminate
    f. Customers enter before teller ready
    g. Customers and tellers sometimes miss signals
4. **How did you fix them?**

   a. I implemented printOrder semaphore to ensure ordered output from 0 to 49
   b. I implemented proper synchronization with volatile boolean (flags) and semaphores
   c. I added queueLock semaphore for synchronized access to customer assignment
   d. I implemented proper resource order (i.e, manager before safe)
   e. I combined volatile boolean (flags) with proper semaphore signaling
   f. I implemented bankOpen semaphore and teller ready signals
   g. I implemented multiple semaphores for different stages of interaction

5. **What did you learn doing this project?**
   a. I learned how to use semaphore on a low level synchronization, understand more about critical section. Also, I learned how to properly handling share resources, prevent deadlock and race condition, and the importance of proper thread termination. There was a problem with the output were not in order, I learned how do incorporate semaphore in multi-thread environment to the output to make the messages printed in sequence, balancing between concurrency and readable output.