Task 3: Explain how the code works

1. At a high level, the system is represented by accounts, a bank, transactions and a collection of transactions. These are the main building blocks of any banking system, but according to my understanding, the most important part of this implementation is the bank and how transactions are handled. The bank runs on one thread, gets some list of transactions and handles all of these in separate threads, where each thread then performs the transaction and updates the immutable accounts registered at the bank up to some threshold for times a transaction can go wrong. All transactions that are completed or have exceeded the allowed amount of failures are then stowed away, while the remaining transactions are prepared for the next round of processing which then happens shortly after.
2. The easiest features to implement were the ones that did not involve concurrency, because reasoning about what to do in those situations were pretty straight forward. Also, the instructions on what to do were pretty clear due to how the tasks were formulated and the comments that stated what had to be implemented for each function.
3. The most challenging parts of the system to implement in descending order were how a single transaction was to be processed and the collection of transactions. The single transaction implementation was challenging because reasoning about what happens when concurrency is in the picture on top of the implementation being the most extensive was a little bit much all at once. But actively trying to consider which parts would be affected by concurrency in that part of the code helped. As for the collection of transactions, I initially implemented it as I would for any normal non-concurrent implementation because I did not read the entire task description. After finishing the ‘first draft’ I actually read the entire task description, and then built the parts that took concurrency into consideration over that.
4. All the tests passed after running the tests the first time except for test 11. This was due to my updating of the accountRegistry map not considering concurrency as stated in task 3.3 above. This was fixed by putting the update inside a this.synchronized clause, forcing the potential other threads to update their registries with the correct state of each account. Except for that blunder, following the task descriptions and comments were enough to get the system up and running and the tests to pass.