Project 2 Summary

By Josh Cline

Test Output:

```
ERR: There were 1000 accounts but only 992 balance queries
ERR: Final balance of all 1000 accounts is 985919018, expect 995860210
The 299 transactions took 7090.78s total, or an average of 23.7150s each
The 992 balance checks took 18249.84s total, or an average of 18.3970s each
```

Summary

Looking at the output it looks like 8 checks were not done when the test asked for them. I looked through the code however I did see where these eight check could have been dropped. The way I did the end puts end commands that the end list one for each worker. When a worker gets an end command it then exits its thread. Once all threads are exited the main program exits. Since all end commands are at the end of the command list they should be the last commands to run with the last one run by the last thread still operating therefore it seems odd that 8 check statements got dropped. Overall I think this project went fairly well however it took a lot of time to try to get everything to work as it should and in the way I wanted.

Part II Coarse vs Fine grained

Coarse

Test 1	real	1m10.651s
Test 2	real	1m10.659s
Test 3	real	1m10.650s
Test 4	real	1m10.645s
Test 5	real	1m10.645s
	Average	1m10.65s

Fine

Test 1	real	1m18.074s
Test 2	real	1m17.684s
Test 3	real	1m18.057s
Test 4	real	1m18.049s
Test 5	real	1m18.050s
	Average	1m17.983s

1) According to running the testing script five times each for coarse and fine and then averaging the times coarse grain was faster.

2) I think this technique may have been faster as it did not have to wait for individual accounts to unlock and therefore used less time especially in the transaction operations which could wait a long time for accounts to unlock tying up threads.

3) I think in a case where the accounts were not so intertwined so less instance of resources being locked that fine grain would be faster especially for larger systems.

4) I think it would slow it down. I believe this because if you locked every 10 accounts of 1000 if a transactions used acccounts 1, 11, 21, 31, 41, 51, 61, 71, 81, and 91 then it would need 10 different locks however each lock covers 10 accounts that other threads may need and would wait on.

5) I think that for a case where the number of accounts is not a lot larger than the number of workers than coarse would work the best. If it was an extremely large bank with few worker threads than I think fine would work better. I think medium would not really work for either case unless the transactions were very sequential in accounts for example TRANS 1 100 2 100 3 100 4 100.