Anthony Nguyen (40210667)

Hadi Hawi (40096690)

Programming Assignment 1

Task 1

Basically, the deposit() and withdraw() methods perform simultaneously, since there are distinct depositor and withdrawer threads for each account that act concurrently and neither method is synchronized. When different threads try to access and modify the same data concurrently, it can cause a race condition. For example, let's say the initial balance for an account is 0, and the depositor thread increments the balance by 1 at the same time as the withdrawer that decrements the balance by 1. They would both take the initial balance of 0 and increment/decrement respectively. If the depositor finishes first, then the balance would be 1, momentarily, until the withdrawer finishes, which would update the balance to -1, instead of the expected 0. If this happens a million times, you can see why the balance would be so inconsistent. The code is easily fixed by making the deposit() and withdraw() methods synchronized.

Task 2

The order of the threads starts with the depositor and withdrawer from account 0, all the way to account 9. Changing the starting order will not affect the consistency of the accounts. No matter how you order the threads to run, it will not change the fact that the methods are not synchronized, and the same problems will occur.

Task 3

```
balance = balance + amount;
```

This segment of code found in the deposit() and withdraw() methods is where the depositor and withdrawer threads try to access and modify the same attribute concurrently.

Task 6

Essentially, a synchronized block is more flexible than a synchronized method. With a synchronized block, you can make it such that any block of code within your method is synchronized. You can put only the critical section of a method within the block for example. For a synchronized method, the entire method is synchronized. You can just put the entire code of a method within a synchronized block, and it would achieve the same thing as if you made the method synchronized.