ETEC3702 – Concurrency
Lab 2 – Concurrent Synchronization with Locks

Due date: 30 January 2020 by the end of class.

For this lab you are going to modify your concurrent program from Lab 1 to allow the simulated bank account program to function while avoiding orderings that produced incorrect results.

The bank account class was as follows:

```
class BankAccount(object):
    def __init__(self,initialBalance):
        self.balance=initialBalance
        self.transactionLog=[]
        self.transactionLog.append("initial balance:" +
                                   str(initialBalance))
    def getBalance(self):
        time.sleep(random.uniform(0,0.00001))
        temp=self.balance
        time.sleep(random.uniform(0,0.00001))
        self.transactionLog.append("getBalance:"+str(temp))
        return temp
    def setBalance(self,amount):
        time.sleep(random.uniform(0,0.00001))
        self.balance=amount
        time.sleep(random.uniform(0,0.00001))
        self.transactionLog.append("setBalance:"+str(amount))
    def withdraw(self,amount):
        # This method withdraws funds by:
        #      getting the balance using self.getBalance.
        #      subtract the specified amount.
        #      restore the amount using self.setBalance.
        #      log the transaction as: "widthdraw("+str(amount)+")"
    def deposit(self,amount):
        # This method deposits funds by:
        #      getting the balance using self.getBalance.
        #      adding the specified amount.
        #      restore the amount using self.setBalance.
        #      log the transaction as: "deposit("+str(amount)+")"
```

Part 1: Fixing the Concurrent Update Problem

- Before modifying the program, execute it several times with concurrent deposit(500), withdraw(50), and withdraw(10) operations. Note the incorrect values that are produced.

- Add a lock to the program above to protect against concurrent bank account accesses.

- Test this program by executing the program 100 times.

  Was the output correct each time?

  Is the execution still non-deterministic?

  Is the program still concurrent?