# CST8130: Data Structures Lab #5- BlockChains with Linked Lists

***DUE: demonstrate in lab during week October 29 – November 1. Note – I will publish the solutions on November 2nd so no demos will be possible after that date.***

In this lab, we will use linked list processing to implement a single BlockChain to secure a classes’ grades. (ie one BlockChain object per course).

For a quick primer on BlockChain concepts:

<http://www.iftf.org/future-now/article-detail/understand-the-blockchain-in-two-minutes/>

<https://www.youtube.com/watch?v=5BGCKkgW8CU>

I have written much of the code for you…you are going to write the methods in class BlockChain only.

The main method is a menu which allows you to add a new “good” block to the chain, add a “bad” block to the chain (to simulate someone trying to hack into the chain) , display the chain data, and verify if the chain is good. I have written this already – I create a BlockChain object (myBlockChain) and initialize it to “my CST8130 course”.

In our implementation, a Block consists of block chain data (student number, grade and a MyDate date which is the date of the entry, the previous block’s hash value, the hash value of the block). The block is also a node in our Linked List so it also contains the next node reference. All the methods that you will need in the Block class have already been written for you. YOU DO NOT CHANGE Block class.

So, you need to write the methods in BlockChain, The data for this class consists of a head reference to the linked list, a tail reference to the linked list and the name of the course whose data the list contains. A description of each of the methods that you need to write:

**public** **void** printBlockChain() {

// write code here to display the contents of the block chain

// (from the head – using the next reference)

}

**public** **boolean** verifyChain() {

// write code here to verify the chain – (from the head – using

// the next reference, checking that each block in the chain’s

// previous hash is equal to the current hash in the previous

// block’s

// Note: if they all are all equal, you return true; if any one

// is not equal, they are false

**return** **true**;

}

**public** **void** addBlock(Scanner keyboard) {

// write code here to add a block to the BlockChain (at the tail)

}

**SAMPLE OUTPUT:**

**Enter 1 to display the chain:**

**2 to add a good block:**

**3 to add a bad block:**

**4 to verify chain:**

**5 to quit: 1**

Chain for CST8130

0 100 2018/11/11 current: 126274.0 previous: 0.0

**Enter 1 to display the chain:**

**2 to add a good block:**

**3 to add a bad block:**

**4 to verify chain:**

**5 to quit: 2**

Enter date: Enter month - between 1 and 12: 10

Enter day - between 1 and 31: 10

Enter year: 2018

Enter student number: 1111

Enter grade: 99

**Enter 1 to display the chain:**

**2 to add a good block:**

**3 to add a bad block:**

**4 to verify chain:**

**5 to quit: 1**

Chain for CST8130

0 100 2018/11/11 current: 126274.0 previous: 0.0

1111 99 2018/10/10 current: 114809.0 previous: 126274.0

**Enter 1 to display the chain:**

**2 to add a good block:**

**3 to add a bad block:**

**4 to verify chain:**

**5 to quit: 4**

Chain is valid

**Enter 1 to display the chain:**

**2 to add a good block:**

**3 to add a bad block:**

**4 to verify chain:**

**5 to quit: 2**

Enter date: Enter month - between 1 and 12: 10

Enter day - between 1 and 31: 11

Enter year: 2018

Enter student number: 2222

Enter grade: 88

**Enter 1 to display the chain:**

**2 to add a good block:**

**3 to add a bad block:**

**4 to verify chain:**

**5 to quit: 3**

Enter date: Enter month - between 1 and 12: 10

Enter day - between 1 and 31: 12

Enter year: 2018

Enter student number: 3333

Enter grade: 77

**Enter 1 to display the chain:**

**2 to add a good block:**

**3 to add a bad block:**

**4 to verify chain:**

**5 to quit: 1**

Chain for CST8130

0 100 2018/11/11 current: 126274.0 previous: 0.0

1111 99 2018/10/10 current: 114809.0 previous: 126274.0

2222 88 2018/10/11 current: 114935.0 previous: 114809.0

3333 77 2018/10/12 current: 115061.0 previous: 0.23065048

**Enter 1 to display the chain:**

**2 to add a good block:**

**3 to add a bad block:**

**4 to verify chain:**

**5 to quit: 4**

Chain is broken

**Enter 1 to display the chain:**

**2 to add a good block:**

**3 to add a bad block:**

**4 to verify chain:**

**5 to quit: 2**

Enter date: Enter month - between 1 and 12: 10

Enter day - between 1 and 31: 13

Enter year: 2018

Enter student number: 5555

Enter grade:

66

**Enter 1 to display the chain:**

**2 to add a good block:**

**3 to add a bad block:**

**4 to verify chain:**

**5 to quit: 1**

Chain for CST8130

0 100 2018/11/11 current: 126274.0 previous: 0.0

1111 99 2018/10/10 current: 114809.0 previous: 126274.0

2222 88 2018/10/11 current: 114935.0 previous: 114809.0

3333 77 2018/10/12 current: 115061.0 previous: 0.23065048

5555 66 2018/10/13 current: 115200.0 previous: 115061.0

**Enter 1 to display the chain:**

**2 to add a good block:**

**3 to add a bad block:**

**4 to verify chain:**

**5 to quit: 4**

Chain is broken

**Enter 1 to display the chain:**

**2 to add a good block:**

**3 to add a bad block:**

**4 to verify chain:**

**5 to quit: 5**

Goodbye