**Lab 05**

**Name:** Will Townsend

**Class:** COSC 320 – Section 751

**Date:** 10/8/2020

**Lab Report:** I found this lab to be difficult, but not for the reason that one would assume. The file, d\_stiter.h, failed to compile because of the implementation of the friend keyword. I pinpointed it to this error exclusively, but unfortunately could not get a program to compile at all. I understood what the lab’s output should be and went off of that. So below is the lab\_05.cpp file based that was unable to run and therefore untested. My only chance is that I understood what the lab asked of me to do. Even though I could not run it myself, I am pretty confident in the objective of this lab. I do not expect to receive full or even substantial credit for Lab 05. This lab was done 100% independently without any outside help. I am way too confused on what was happening so this is the best I could do.

**Lab:**

**int.h**

#ifndef Integer\_H

#define Integer\_H

class integer

{

public:

// constructor. initialize intValue and set count = 1

integer(int n):intValue(n), count(1){}

// return intValue

int getInt(){return intValue;}

// return count

int getCount(){return count;}

// increment count

void incCount(){count++;}

// compare integer objects by intValue

friend bool operator< (const integer& lhs, const integer& rhs){

if(lhs.intValue<rhs.intValue)

return true;

return false;

}

friend bool operator== (const integer& lhs, const integer& rhs){

if(lhs.intValue==rhs.intValue)

return true;

return false;

}

// output object in format intValue (count)

friend ostream& operator<< (ostream& ostr, const integer& obj){

ostr<<obj.intValue<<"("<<obj.count<<")";

return ostr;

}

private:

// the integer and its count

int intValue;

int count;

};

#endif

**lab\_05.cpp:**

#include<stdio.h>

#include<iostream>

#include<utility>

#include"int.h"

#include"d\_except.h"

#include"d\_stree.h"

#include"d\_random.h"

#include"d\_stiter.h"

int main(){

stree<integer> tree;

randomNumber rand;

const int NUMCOUNT = 10000;

puts("Lab 05:");

for(int i=0;i<NUMCOUNT;i++){

integer num = integer(rand.random(7));

if(tree.size()==0||!(tree.find(num).nodePtr->nodeValue==num))

tree.insert(num);

else

tree.find(num).nodePtr->nodeValue.incCount();

}

tree.displayTree(2);

puts("");

return 0;

}

**NO OUTPUT TO RECORD DUE TO COMPILING ERRORS**