

# CS 2110 Timed Lab 6

## Policy

### **RESTRICTIONS**

You must submit by the end of your lab period. Bear in mind that the clock on your computer may be a few minutes slow. We will not accept late submissions, be they 1 second or 1 hour late.

You may not leave the classroom until we have verified that you have submitted the lab. If you leave the classroom without submitting you will receive a zero.

The timed lab has been configured to accept one submission. If you accidentally submit or submit the wrong version, call one of the TA's and we will reopen submission for you.

### **AUTHORIZED MATERIALS**

- The assignment files
- Your previous Homework and Lab submissions
- Your mind
- Blank paper for scratch work

### **UNAUTHORIZED MATERIALS**

- The Internet (except the T-Square Assignment page to submit)
- Any resource on T-Square that is not given in the assignment
- Dropbox
- Notes on paper or saved on your computer
- Email
- Contact in any form with any other person besides TA's
- If you have any questions on what you may not use then assume you can't use it and ask a TA

### **QUESTIONS**

If you are unsure of what questions mean, the TA's will clarify them to the best of their ability. In the end you are solely responsible for what you submit. We will not be able to answer any questions about how to reach a solution to the lab questions.

### **VIOLATIONS**

Failure to follow these rules will be in violation of the Georgia Tech Honor Code and you will receive a zero. and you will be reported to the professor and the Office of Student Integrity. We take cheating and using of unauthorized resources very seriously and you will be in serious trouble if you are caught.

# The Assignment

## OBJECTIVE

In today's timed lab, you will be making an integer linked list in C. It must conform to the following requirements:

- The list is singly linked, i.e. each node only points to the next node in the list.
- The list has a head pointer stored in the LIST struct.
- The last node in the list has its next pointer set to NULL.
- The size field in the LIST struct is set to the number of nodes currently in the list.
- The list only contains integer values as data. This means you will store the integer in the node without calling malloc/free to allocate space for the integer data.
- The code must implement `create_list`, `create_node`, `push_front`, `pop_back`, `destroy` and `print_list`.

## RESTRICTIONS

- Your code must not crash, run infinitely or produce any memory leaks.
- Your code must compile with the Makefile that we have provided! If it does not compile with our Makefile you will receive a 0.

## RUNNING YOUR CODE

We have provided a makefile for this assignment that will build your project. Your code must compile with our makefile. Here are the commands you should be using with this makefile:

- To run the tests in test.c: **make run-test**
- To run the tests in test.c in debug mode: **make run-debug** (be sure to run this command first: **make clean**)
- To debug your code using gdb: **make run-gdb** (run the clean target first)
- To run your code with valgrind: **make run-valgrind** (run the clean target first)

## HINTS

- Note that in this timed lab you will NOT be using any function pointers. You don't need to worry about that.
- Make sure that you test your list thoroughly in your test.c file. Our test cases will be exhaustive.

## DELIVERABLES

Please turn in your list.c, list.h, test.c and Makefile.