

CS 121  
Bruce Bolden

Lab Assignment #9  
Date: April 2, 2015

## 1 Exam Programming

The programming section of the first exam was a difficult section for quite a few people. One way to fix these concepts firmly in mind is to actually code and then run that code—a compiler is often much less lenient than a grader!

### 1.1 Functions

The problems as defined on the exam:

1. Define an *interface* for a stack class. Your code should define all standard methods. No external objects may be used in your definition (no list object).

**Note:** Write a class method to insert an item onto a stack (use the interface defined in the previous item). No external class objects may be used.

2. Write a function to *zip* (join by interleaving) two singly-linked lists together, such that the second list is *reversed* before joining. For example, given the lists  $L1 = \{a, b, c\}$  and  $L2 = \{x, y, z\}$  the resulting list would be:  $Lzip = \{a, z, b, y, c, x\}$ . You may assume that you have a Stack class. No global variables may be used. There is no I/O (input/output) in the function.

3. **Additional:** Write a function (not a program) to remove (delete) all of the negative values from a singly-linked list of numbers.

Describe any additional assumptions you make when writing this function. *Objects* may **NOT** be used!

### 1.2 Testing

The process of testing is often slighted during development. It often reveals flaws (minor and major) that are unforeseen in the design process.

For example, to demonstrate that your last function behaves properly, you should have tests for the following::

- The *empty* list.
- A list with one item in it.
- A list with two items in it.
- A list with three items in it.
- A list with numerous items in it.

You may find it useful to write several functions to generate test cases to exercise your functions (think loops, starting with zero items). You may also find it useful to have the test generating functions generate lists (using an `AddToEnd()` function that are organized both inorder (ascending and descending) and randomly).

### 1.3 Deliverables

Annotate a `script` session to demonstrate:

1. Results of file compilation.
2. Results of using functions in a program(s). It should be clear to the reader of the output what test is being exercised.