



UNIVERSITY *of* WASHINGTON | BOTHELL

SCHOOL OF STEM

Computing & Software Systems

CSS342 Data Structures, Algorithms, and Discrete Mathematics I

Spring 2020

Assignment 5: Linked Lists

Due date: Friday 12 June

Purpose

Practice the use of pointers and pointer operations to implement various linked-list functionalities. In this assignment you will design and implement your own single-linked-list class (not at template) from scratch in C++.

List Functionality

Design your own linked list class to hold a series of objects.

Basic Operations

The class should have member functions to implement all basic functionality of a list (i.e., appending, inserting and deleting nodes) as well as constructors and destructors.

You should implement this list such that it can hold any kind of object in the node. You must implement this as a pointer to an object.

Here is what the abstract node would look like

```
Node
{
    Object *data;
    Node *next;
}
```

For the purpose of this assignment, the Object can simply be a pointer to an integer, so that you can easily display and verify your results, but in theory, you could plug in **any** object.

In addition, you should implement the following member functions:

Add a print member function. This should display all the values in the linked list, test the class by starting with an empty list, add some elements and print the resulting list.

Add a copy constructor. Test by making a list, making a copy and displaying the values in the copy.

Reverse

Add a member function that rearranges the nodes so that their order is reversed. Test by printing the list before and after it is reversed.

You are not allowed to use any additional list to reverse the original list. You should do this by rearranging the pointers.

Part 1: implement an iterative version of the reverse function, place your solution in a function named ReverseIterative

Part 2: implement a recursive solution of the Reverse function, name it ReverseRecursive

Hint: implement the recursive solution first, then use that as a basis for the iterative solution.

Note: if you use *any* external sources or references, you must include the reference in your document, otherwise it will be considered plagiarism.

Driver

Allocate a list of 10-20 random integers.

Demo all the functionalities of your list with a driver program.

Clearly print out the results of each operation (e.g., list before and after an insert).

Deliverables

1. Source code
 - a. List Class
 - b. Driver
 - i. Must demonstrate every an each functionality described above
2. Documentation
 - a. Discuss your design and approach for solving the problem
 - b. You are encouraged to use pictures to illustrate what you're doing
 - c. Discuss the differences and similarities between the two approaches, i.e., iterative vs. recursive

Grading Guide

Out of 20 pts

1. Documentation (6 pt)
 - Design and approach discussion: 4pt
 - Differences and similarities: 2pt
2. Correctness (12 pt)

As usual, zero points if program doesn't compile

 - [1p] Correct implementation of basic operations
 - [1p] Copy Constructor
 - [1p] Print list
 - [4p] Reverse Iterative
 - [4p] Reverse Recursive
 - [1p] Driver
3. Program organization (2 pt)

Comments:

Good (1pt) Bad or none (0 pt)

Coding style

Good (1pt) Bad or mediocre (0 pt)