# Homework 5:
# A Link Between Worlds

Due date: April 10 at the beginning of lecture

## Overview

In this homework, you will program a singly-linked list of integers similar to the linked stacks from lecture. You will use modern C++ memory management via the `unique_ptr` class found in the `<memory>` library. You can use the `ModernLinkedStack` as inspiration, but will need to add a few more methods to it.

## Assignment

Copy the files `ModernLinkedList.h` and `ModernLinkedList.cpp` from the GitHub repository, then make a new project with those files. Review the files so you're clear on what instance variables and methods are expected on this type, then begin to implement the methods from the .h file as described. Some notes:

1. The procedure for adding a new element can be different depending on whether the list is currently empty. You can use the current `mSize` to determine if the list is empty.
2. The existing `PrintList` and `PopBack` functions shows an example of "walking" through the nodes of the list. You will need to do something similar for `PushBack`, `Insert` and `RemoveAt`, and `operator[]`. (`operator[]` is equivalent to the "get" function from Java's linked lists – it returns the data element at the given index.)
3. You should write a main to test your list. Add and remove some elements, and use `PrintList` to periodically check the correctness of the list.
4. If you do this correctly, you will **never** have to use `new` or `delete` – that's the **entire purpose** of the `unique_ptr` type.

## Deliverables

Hand in:

1. ModernLinkedList.cpp and your main.cpp. Submit to Dropbox.