EE208 Laboratory Session 6

Bryan Hennelly

5 March 2018

1 Lab objectives

In this lab you will continue to work with classes in C++ with some additional exercises. You should read over the lectures on classes. The purpose of this lab is to practice the fundmentals of using classes, including access specifiers (public, private, etc.) as well as contructors and destructors. You will need to read over your lecture slides to understand these concepts in order to complete the exercised below. The due date is 5pm Monday 12th March.

Learning Outcomes
From a C++ perspective you will be able to implement simple C++ classes and use instances of those classes (objects) within your programs. In this lab you will create and use classes, you will employ contructors and destructors and you will overload the + operator. You will store your code in different cpp nd h files as described below.

2 Question:

For the problem given below write a C++ program that provides a solution. Your solution should be made up of three files:

- **You should have a header file called myarray.h in which you define your class. However, only function declarations should exisit in this class definition.**
- **You should have a cpp called myarray.cpp in which you define your class functions using the scope :: operator.**
- **You should have another cpp file in which you have a main function that creates objects of type myarray and tests the functionality described below.**

Exercise 6.1: Create a new class called **myarray**. This class will behave just like an array with some extra functionality. The class should have just two private data members: a pointer to an integer, which we will use for dynamic memory allocation, and an integer called N.

You should create a constructor that takes in an integer, sets N equal to this value, and uses dynamic memory allocation to create memory for N integers. You should also create a default contructor that sets the pointer = NULL and N = 0. Similarly you should add destructors to you class to free up any allocated memory.

Your class should have functions called void setvalue(int index) and int getvalue(int index) that allow you to set and get the value in the array at a given index.

You should overload the operator+ such that you can perform the following operation in the main function A = B + C; where A, B, and C are all objects of type **myarray** and the values stored in A are given by the values stored in B follwed by the values stored in C. Therefore A has length equal to the sum of the length of B and the length of C.

Finally, add a function to your class called void push(int newvalue); This function should take in a new value and add it to the end of your array. This is a little tricier than you might think as you will have to grow the size of your array somehow to N+1.