CS 121 Week 6 - Vector Worksheet

**Introduction:**

Before today, the arrays that we've been working with are called *static arrays*. This pretty much means that the size they take up in RAM is a constant/fixed amount. Now we will be working with vectors, a dynamically-sized type of array, which come with both major pros and cons.

If you need to recall certain parts about vectors, consult your notes, the power point notes, the book, or online (specifically, cplusplus.com). Search "vectors cplusplus" and it should be the first result.

Focus on the following vector functions:

- (constructor)\*, operator=, size(), empty(), clear(), operator[x], at(x), front(), back(), push\_back(x), pop\_back()

\*(constructor) includes all the functions from "Defining Vectors", section 8-3, page 3 of the Ch8.1 PPT.

It's a good idea to try the following out on your computer, writing down notes to help you remember the purpose and name meaning of each function.

**Syntax Practice:**

Use the following variables for the below questions:

const int STUDENT\_SIZE = 4;

double grades[STUDENT\_SIZE] = {99.3, 75.4, 88.2, 44.3};

vector<double> dyn\_scores;

vector<double> more\_scores;

1. Add five values (ranging from 0 to 100) to dyn\_scores (use push\_back for this, for loop or manually)
2. Print out the contents of dyn\_scores to console output.
3. Remove all values from dyn\_scores (write two ways: one with pop\_back and a for loop, and the other which consists of a single line)
4. Copy the contents of grades to more\_scores (use either at, [], or push\_back)
5. Copy the contents of more\_scores to dyn\_scores (one line)
6. Print out the mean/average of dyn\_scores (declare and initialize a double for getting the mean)
7. Set each element of dyn\_scores, more\_scores, and grades to 0 (show the shortest way possible for each).
8. Create an if-statement to see if dyn\_scores and more\_scores are **both** empty (just need the conditional)
9. Write four ways to print the first and last positions of more\_scores.

**General Concept:**

1. When working with vectors, do we need to define a constant integer for the size of a vector and an integer counter to count the amount of elements inside a vector (like static arrays)? Why or why not? Provide examples for equivalents if they exist.

2. Is a vector a primitive data type? Explain why or why not.

3. List all the pros and cons static arrays and vectors have that we have discussed in the notes. Try to do this by what you remember, then refer to your notes.