**Exercises for Week 3**

Note 1: Throughout this exercise pay attention to formatting and readability of your code, correct indentation, header comments on the class plus all methods, plus internal comments on anything interesting/difficult/noteworthy in the implementations.

Note 2: Use Git to manage your work. Create a new Git repository for this project. Each time you get something working, commit the changes. Experiment with creating a branch, working in the branch and the trunk independently, and then merging the changes back together.

Note 3: I strongly suggest that you work on Exercises 1 & 2 simultaneously. In other words, create a minimal version of class ArrayOfNumbers and a minimal version of the driver program, just enough so that you can compile and run. When that's actually working correctly, then commit it and proceed by adding one new feature at a time to the class, and corresponding code in the driver.

1. Last week's program involved a class of objects that stored exactly two integers internally. Make a new class called ArrayOfNumbers that stores an array of integers instead. Remember to declare this as private or protected. It should not be accessible except through the public interface of the class.

This class must have two constructors.

- One constructor should take an integer parameter that tells how big the array should be.

  This should set all the values to zero.
- The other constructor should take an array of integers as an input parameter, and set the internal storage equal to a copy of that array. (Be careful here: do not just assign

  something like storage = input. You will need to create a new array of the right size for

  storage, and then loop through, copying each array element across.)

The class will need an accessor method item() that takes an integer parameter saying which item to look at.

You will need a setter method called setItem() that takes two integer parameters. The first one is the index of the item you are setting, and the second is the value you are putting in there.

Create a method max() that returns the largest item stored. You will need a loop for the implementation.

Create a method equal() that takes two integer parameters saying which two elements to check for equality.

Create a similar method called gcd() that returns the greatest common divisor of two elements.

Add new methods called count(), sum() and average() that do what it says on the tin.

Write two new mutator methods:

- scalarMultiply() takes a single integer parameter, and multiplies each element of the array by that number.
- addConstant() takes a single integer parameter and adds that to each element of the array.

2. Write a test program for this class that exercises all its functionality. A simple one would just be hard coded. A better solution would have a menu interface that prompts the user for commands. Do what you can.

3. If you get all of that working (and only if), you could think about how you might implement a version of addObject() for this new class. What conditions would need to be satisfied by the two ArrayOfNumbers objects for this to work? Try it and see.