COEN 44 Applied Programming **Fall 2016**

Lab 7

We've already seen examples of using arrays to store tabular data. Arrays can also be used to represent objects. For example, a vector in 2-space can be represented by an array

```
        float vector2d [2]
```

where vector2d [0] is the x coordinate of the tip of the vector, and vector2d [1] is the y coordinate of the tip (assuming the tail is at the origin). Similarly, a vector in 3-space would have three elements, for x, y and z coordinates.

We're going to use these representations to do some vector calculations

Objectives:
1. Gain experience with using reference parameters and arrays as parameters

Part 1: Design and implement a program to do simple vector arithmetic

Two 2-D vectors represented as described above can be added by adding the x coordinates of both vectors to get the x coordinate of the sum, and adding the y coordinates to get the y coordinate of the sum. Similarly for subtracting vectors – just subtract the second x coordinate from the first, and the same for y coordinates. 3-D vectors also have to deal with a z coordinate.

Your physics professor has asked you to write a program to read in x, y and z coordinates for the tips of two vectors in 3-space, and store each in a 1-dimensional array. Your program should then call two programmer-defined functions, vectorsum and vectordiff, which will add and subtract the vectors. Each of these functions has three parameters: an array representing the first vector, an array representing the second vector, and a third array which will hold the vector resulting from the vector arithmetic operation.

Because your professor hopes to win a Nobel prize, he is also developing a theory that "extreme coordinate" vectors (those with a very large or very small coordinate) hold the key to understanding dark matter. He's asked you to write a third programmer-defined function, vectorminmax, which will determine the largest and smallest coordinates of the two vectors. Vectorminmax has four parameters, two arrays representing the vectors, and two variables to hold the minimum and maximum coordinate.

Your main function should read in two vectors; call vectorsum, vectordiff, and vectorminmax; then print out components of the vectors read, the components of the sum and difference vectors, and the minimum and maximum coordinate values.

Part 2: Program execution

Lab Steps:

1. Turn in your written program outline to the TA
1. Open Visual Studio and create a new project called Lab7
2. Create your program
3. Debug and test the program
4. When working properly, demonstrate for the TA
5. Turn in a program listing to Camino

Part 3: Extra Credit

The functions written to solve Part 2 of this lab only work for vectors in 3-space. Make these functions more general by modifying them to work for vectors of any positive-dimensional space by adding a parameter that holds the number of dimensions the vectors have.

Lab Steps:

1. Demonstrate your working program from Part 2 first
2. Modify your program as described above
3. Debug and test the program
4. When working properly, demonstrate to the TA that you can handle 2-dimensional and 4-dimensional arrays
5. Turn in a program listing to Camino