

Monday 10 November

Programming Assignment 08: due Wednesday 12 November

## Class Activity 22:

### *Objectives:*

Learn about simple array useage.

### *Activity 1:*

Retrieve the program `ca22a.cpp` from `get12`. The program calculates the average of a series of lengths. After reviewing the code, run the program a couple of times.

### *Activity 2:*

Modify the program to take up to 20 lengths. Save the modified program as `ca22b.cpp`. Run the program.

### *Activity 3:*

Modify the program to ask the user for the lengths rather than calculating them internally. Save the program as `ca22c.cpp`. Run the program.

### *Activity 4:*

Design a function that will find the minimum value in a type `double` array. Use the function prototype

```
double min_d_array( double array[], int numused );
```

Hint: you must search the array from the first element (0) to the last element in use (`numused-1`) looking for the minimum. Since you don't know the minimum, assume that the first element is the minimum and then compare succeeding elements with the minimum, always updating the minimum if the new element is less than the minimum.

Add your function to `ca22c.cpp` and use it to find the minimum length input by the user. Modify the display function to show the minimum value in addition to the average. Save the program as `ca22d.cpp`. Run your program.

### *Activity 5:*

Modify your function from activity 4 to also find the index of the array element that holds the minimum value. Use the new function prototype

```
void min_d_array( double array[], int numused,  
                 double *p_minvalue, int *p_loc );
```

Why did the prototype change from type `double` to type `void`?

Replace your function from activity 4 with the new function. Modify the display function to also show the index value for the minimum. Save the program as `ca22e.cpp`. Run your program.

**Activity 6:**

Modify `ca22a.cpp` to take up to 1000 lengths. Save the program as `ca22f.cpp`. Run the program.

**Activity 7:**

Modify `ca22a.cpp` to take up to 100,000 lengths. Save the program as `ca22g.cpp`. Run the program.

**Activity 8:**

Modify `ca22a.cpp` to take up to 1,000,000 lengths. Save the program as `ca22h.cpp`. Run the program.

**Turn in:** A copy of your program from activity 5 (`ca22e.cpp`) and an example screen display from running the function with 5 input lengths.