We'll use a couple of UNIX commands -- create a directory (folder), then change to that directory. Edit a small program to find the maximum of an array... And then compile the program using the tool "make."
pmadden@ubuntu: ~/cs120/lab

pmadden@ubuntu:~/cs120$ mkdir lab
pmadden@ubuntu:~/cs120$ cd lab
pmadden@ubuntu:~/cs120/lab$ ls
pmadden@ubuntu:~/cs120/lab$ emacs lab2.c
pmadden@ubuntu:~/cs120/lab$ ls
lab2.c
pmadden@ubuntu:~/cs120/lab$ make lab2
cc  lab2.c  -o lab2
pmadden@ubuntu:~/cs120/lab$ ./lab2
The maximum is 9
pmadden@ubuntu:~/cs120/lab$
What's Happening in the Terminal

• We create a new folder with `mkdir`
• We change to the folder with `cd`
• `ls` is the command to list the files in the current directory. If we've just created the directory, it's empty.
• I've edited "lab2.c" with emacs -- but if you want to use vi, gedit, or something else, go for it.
• After saving the file, `ls` shows that the file exits.
• Unix systems have a tool called `make`, which can be used to compile source code into an executable program. Make is very complex, very powerful, and will be your friend for years and years and years.
#include <stdio.h>

int maximum(int *array, int length)
{
    int m = array[0];

    for (int i = 1; i < length; i = i + 1)
    {
        if (array[i] > m)
            m = array[i];
    }

    return m;
}

int main()
{
    int a[] = {3, 8, 2, 9, 4};
    printf("The maximum is %d\n", maximum(a, 5));
}
What's happening with the code

• We're creating a function (subroutine) called maximum. It accepts an array of integers, and an integer that specifies the length. In object oriented languages, an array is often a more complex data structure, and it can tell you the length. In C, you don't have many fancy features, and have to keep the length stored separately. When we get down to the hardware on a microprocessor, this will make a lot more sense. For right now, just go with the flow....

• We scan through the array, looking for the largest element; we put that into a variable called m, and then return it.

• In the main program, we create an array called a, and load five numbers into it. We then print the maximum.
Coding assignment

• Modify the code to add functions called **minimum**, **total**, and **average**.

• Print the maximum, minimum, total, and average values (use integers for everything; don't worry about floating point numbers).
This shouldn't be hard either

No expectation that this should be a tough coding challenge. The goal is to get everyone familiar with the C compiler, and the syntax of the C language. If you know other programming languages, it shouldn't be too much of a stretch to get C working.

Make sure you're doing the coding on your own. If you need help from someone else, or you can only code using cut-and-paste, we've got trouble, and it's good to know that before the drop deadline.