When you have completed your lab exercises, call for a TA to check your work. The marking will be done on a 2/1/0 basis for each of the four problems (2 = great/1 = credible attempt/0 = little or no effort). The TA will record your mark (maximum of 8 for this lab) and you're done! If you can't complete your lab before the end of the lab period and you want to continue working, you may continue to work on the lab on your own until the beginning of your next lab session. You must then have your lab marked at the *beginning* of that session. Do not ask to have your lab marked later than that -- the TAs will be too busy to accommodate you at that time.

We will use g++ as the default compiler, using the CPSC department's Linux machines (i.e., lulu.ugrad.cs.ubc.ca). You can easily log on to lulu from home, using ssh. For example: "ssh lulu.ugrad.cs.ubc.ca" or "ssh lulu.ugrad.cs.ubc.ca -l r2d2" will connect you, where "-l" reads "minus el", and where "r2d2" should be replaced by your own Unix userid. You may find helpful information at www.cs.ubc.ca/support/connecting-cs-undergrads

Exercises for Lab 1:

This lab is due by the *start* of your next lab.

The following is an introduction to C++ through a variety of simple activities. It is also recommended that you read the C++ Primer in your textbook, and ensure that you *practice* as much as possible.

You may complete the following using the command line or an IDE of your choice (note that Eclipse is *not* recommended for C++ development; Xcode or Visual Studio are fine). Ask the TAs for assistance if you have trouble getting your program up and running.

1) Declare an array with 10 elements. Write a function to fill the elements with the integers 1-10 called `fill_array`. Modify the function so that it accepts two integers. The first integer represents the first element, the second integer represents the increment between the each element. (E.g. `fill_array(4,2)` will fill the array with the numbers 4, 6, 8, 10, 12, 14, 16, 18, 20, 22). Print the contents of the array to the screen once it has been filled.

2) C++ arrays do not have a length member. Using what you know from the good programming practices taught to you in your previous classes, how should array length be handled in a C++ program?

3) Playing with IO: Create a plain text file called "`first_file`" and enter a few lines of text. Compile and run the following two programs that demonstrate input and output in C++.

Program 1(file I/O):

```
#include <iostream>
#include <string>
#include <fstream>
using namespace std;
int main(void) {
    ifstream in("first_file"); // Open for reading
    ofstream out("copy_of_first"); // Open for writing
    string s;
    while(getline(in, s)) // Discards newline char
        out << s << "\n"; // ... add it to the new file
    cout << "End of program\n";
return 0; }
```

Program 2 (terminal I/O):

```
#include <iostream>
using namespace std;
float circle_area(float radius); // declare function before main()
int main(void) {
    float circle_radius;
    cout << "Enter radius:" << endl;
    cin >> circle_radius;
    cout << circle_area(circle_radius) << endl;
    return 0;
}
float circle_area(float radius) {
    return 3.14159 * radius * radius; // = pi * r^2
}
```

4) Write a program that simulates a guessing game. It should randomly generate a number (look up `rand` and `srand` using Google for help on random numbers) and ask the user to input a guess. The game should keep running until the user gets the number correct, or otherwise indicates that they wish to end the game.

5) Be sure to show your work to your TA before you leave, or at the **start** of the next lab, or somewhere in between. Otherwise, you will not get credit for your work.