Exercises for Lab 2:

Pre-Lab: Pointer introduction: http://www.functionx.com/cpp/Lesson13.htm

These must be completed and shown to your lab TA either by the end of this lab, or by the start of the following lab):

1. Download the code found in the Lab 02 code folder on UBC Connect.

2. Consider the program below. On a piece of paper, draw a box for each variable. Inside each box put the contents of that variable; if the box contains a memory address, draw an arrow pointing to that address instead. As you walk through the program and prepare your diagram, fill in the blanks below.

```
int x = 5, y = 15;
int * p1, * p2;

p1 = &x;        // x contains _____; y contains _____

p2 = &y;        // x contains _____; y contains _____

*p1 = 5;        // x contains _____; y contains _____

*p1 = *p2;      // x contains _____; y contains _____

p2 = p1;        // x contains _____; y contains _____

*p1 = *p2+10;   // x contains _____; y contains _____
```

3. Examine the following code (a copy is also in the file you downloaded), and refer to the questions on the next page:

```
#include <iostream>
using namespace std;

int a = 7;
int b = 6;
int* c = &b;
void test( int& x, int y, int*& z ) {
    x++;
    y++;
    z= &a; }
int main() {
    test(a,b,c);
    cout << a << " " << b << " " << *c << endl;
    return 0;
}
```

a. Draw a memory diagram showing how memory has been allocated and the contents of each memory location. Note that `int&` is an integer passed by *reference* (not passed by copy).

b. What happens if you make `b` a pointer? Explain and update your diagram from (a), if necessary.

c. What happens if you make `y` a pointer in the original program? Explain and update your diagram from (a), if necessary.

d. What happens when you modify the test arguments of the original program? In particular, try changing the various arguments from pass-by-reference to pass-by-value and vice versa. What happens to the output?

e. Compare these two uses of ampersand (&):

```
int* c = &b;
void test( int& x, int y, int*& z ) {
```

What is `&` doing in each case?

4. Be sure to show your written work to your TA before you leave!

5. **In lab or take home:** A brief look at classes in C++. Complete and debug the `CDate` class that you downloaded at the beginning. Note that while our emphasis will not be OOP here, but you should be familiar with how classes are handled in C++.