The website for these SI sessions is `https://github.com/benblazak/2014-fall-si-cpsc120`.

Many of these examples are from `https://github.com/benblazak/2014-spring-si-cpsc120`, which is full of stuff I wrote for a lab last semester. If you're looking for extra practice, this is one of many places you might start.

Along with your book, `http://www.cplusplus.com/doc/tutorial/` is a great resource for tutorials, and `http://www.cplusplus.com/reference/clibrary/` is a great reference.

Booleans

simple-booleans.cpp

```
#include <iostream>

/**
 * Notes:
 * - Sometimes (especially in documentation) we say "lvalue" instead of "the
 *   value on the left hand side" and "rvalue" instead of "the value on the
 *   right hand side".
 *
 * References:
 * - http://en.wikipedia.org/wiki/Operators_in_C_and_C%2B%2B
 */

int main() {
    bool a = true;
    bool b = false;
    bool c = 1;
    bool d = 0;
    bool e = 5;  // remember automatic type conversion?

    std::cout << "a = " << a << std::endl;
    std::cout << "b = " << b << std::endl;
    std::cout << "c = " << b << std::endl;
    std::cout << "d = " << b << std::endl;
    std::cout << "e = " << b << std::endl;

    std::cout << std::endl;

    std::cout << "(5 < 2) : " << (5 < 2) << std::endl;
    std::cout << "(2 < 5) : " << (2 < 5) << std::endl;
    std::cout << "(2 <= 5) : " << (2 <= 5) << std::endl;
    std::cout << "(5 >= 5) : " << (5 >= 5) << std::endl;

    std::cout << std::endl;

    // does this make sense? what is it doing? :)
    std::cout << "(\"hello\" < \"world\") : "
              << ("hello" < "world") << std::endl;

    std::cout << std::endl;

    // in C++, ‘=‘ is for assignment (so ‘a = 5‘ means "set a to 5"), while
    // ‘==‘ is for comparison (so ‘a == 5‘ means "is a equal to 5?")
    std::cout << "(2 == 5) : " << (2 == 5) << std::endl;
    std::cout << "(5 == 5) : " << (5 == 5) << std::endl;
    std::cout << "(’c’ == ’c’) : " << (’c’ == ’c’) << std::endl;
    std::cout << "(4.5 == 4.5) : " << (4.5 == 4.5) << std::endl;
    std::cout << "(true == false) : " << (true == false) << std::endl;
    std::cout << "(true == true) : " << (true == true) << std::endl;
    std::cout << "(\"this\" == \"that\") : "
              << ("this" == "that") << std::endl;
    std::cout << "(\"this\" == \"this\") : "
              << ("this" == "this") << std::endl;

    std::cout << std::endl;

    // we often read ‘!‘ as "not"
    std::cout << "(!true) : " << (!true) << std::endl;    // not true
    std::cout << "(!false) : " << (!false) << std::endl;  // not false
    std::cout << "(5 != 2) : " << (5 != 2) << std::endl;  // does 5 not equal 2?

    std::cout << std::endl;

    // ‘&&‘ means "and": true only if the left value *and* the right value are
    // true
    std::cout << "(true && true) : " << (true && true) << std::endl;
    std::cout << "(true && false) : " << (true && false) << std::endl;
    std::cout << "(false && true) : " << (false && true) << std::endl;
    std::cout << "(false && false) : " << (false && false) << std::endl;

    std::cout << std::endl;
```

```
    // ‘||‘ means "or": true if either the left value *or* the right value is
    // true
    std::cout << "(true || true) : " << (true || true) << std::endl;
    std::cout << "(true || false) : " << (true || false) << std::endl;
    std::cout << "(false || true) : " << (false || true) << std::endl;
    std::cout << "(false || false) : " << (false || false) << std::endl;

    return 0;
}
```

```
a = 1
b = 0
c = 0
d = 0
e = 0

(5 < 2) : 0
(2 < 5) : 1
(2 <= 5) : 1
(5 >= 5) : 1

("hello" < "world") : 1

(2 == 5) : 0
(5 == 5) : 1
('c' == 'c') : 1
(4.5 == 4.5) : 1
(true == false) : 0
(true == true) : 1
("this" == "that") : 0
("this" == "this") : 1

(!true) : 0
(!false) : 1
(5 != 2) : 1

(true && true) : 1
(true && false) : 0
(false && true) : 0
(false && false) : 0

(true || true) : 1
```

```
(true || false) : 1
(false || true) : 1
(false || false) : 0
```

string-comparisons.cpp

```cpp
#include <iostream>
#include <string>    // needed for ‘std::string‘
#include <cstring>   // needed for ‘strcmp()‘

/**
 * Notes:
 * - The code ‘std::string("this")‘ casts the string literal (which is really a
 *   cstring) to a ‘string‘ object.  If we were using ‘using namespace std;‘ it
 *   would only be ‘string("this")‘, and would look a little more normal.
 *
 * References:
 * - string library reference (see the "compare" function, near the bottom)
 *   http://www.cplusplus.com/reference/string/string/
 * - ‘strcmp‘ function reference, from the cstring library
 *   http://www.cplusplus.com/reference/cstring/strcmp/
 */

int main() {

    // can’t really rewrite this one.  what would it mean to say one string was
    // less than another anyway?

//     std::cout << "(\"hello\" < \"world\") : "
//               << ("hello" < "world") << std::endl;

    // the second two can be rewritten though

    std::cout << "std::string(\"this\").compare(\"that\") : "
              << std::string("this").compare("that")
              << std::endl;

    std::cout << "std::string(\"this\").compare(\"this\") : "
              << std::string("this").compare("this")
              << std::endl;
    // note that we could also avoid the string library altogether, using the
    // cstring library functions instead.  it would probably be better not to
    // do this though, at least during this class, as it's really better form
    // in C++ to do things the C++ way; unless for performance reasons you find
    // you can't.

    std::cout << "strcmp(\"this\", \"that\") : "
              << strcmp("this", "that")
              << std::endl;

    std::cout << "strcmp(\"this\", \"this\") : "
              << strcmp("this", "this")
              << std::endl;

    return 0;
}
```

```
std::string("this").compare("that") : 8
std::string("this").compare("this") : 0
strcmp("this", "that") : 8
strcmp("this", "this") : 0
```

If-Else

if-else.cpp

```
#include <iostream>
using std::cin;
using std::cout;
using std::endl;

int main() {
    int x;  // to store the user's integer

    cout << "Please enter an integer: ";
//     cin >> x;  // we'll just set it manually in the code, for now
    cout << endl;
    x = 15;

    if (x < 0) {
        cout << "Your integer is negative\n";

        if (x % 2 == 0)
            cout << "Your integer is even\n";
        else
            cout << "Your integer is odd\n";

    } else if (x < 1000) {
        cout << "Your integer is fairly small\n";

        if (x % 2 == 0)
            cout << "Your integer is even\n";
        else
            cout << "Your integer is odd\n";

    } else {  // x >= 1000
        cout << "Your integer is fairly large\n";

        if (x % 2 == 0)
            cout << "Your integer is even\n";
        else
            cout << "Your integer is odd\n";
    }

    return 0;  // success
}
```

```
Please enter an integer:
Your integer is fairly small
Your integer is odd
```