

# Week Six

Max Olivier

March 7, 2009

## Readings

For this week please read Chapters 10 and 11 in Overland on Classes and Objects. A good deal of Chapter 10 is probably review, so feel free to skim through it keeping an eye out for the distinctive features of C++ objects and classes.

## Main Ideas

- As you are well familiar with from previous CS courses, capacity for object-oriented programming is often a desirable feature of a programming language, and the big advantage of C++ over C is just this, so I couldn't create a C++ course without talking about classes and objects, which we will do for the next 3 weeks. At the end of this first week, though, you should be familiar with
- The general principles behind object-oriented programming layed out in Chapter 10 (though as I mentioned much of this should be review.
- The syntax and structure of class and object declaration in C++.
- The difference between C structs and C++ classes, and what struct means in C++.
- What an inline function is, and how it can be used to make code less cumbersome.
- How to include header files to be able to use classes across programs.
- The difference between public and private variable declarations, and when to use which.

## Exercise 1 (Exercises from the Reading)

All the exercises from the reading, which are only in Chapter 11.

## Exercise 2 (Cryptology Program)

First download and look at the header file `Translator.h` (for a reminder on what header files are, see pages 282-283 of Overland). Study it, and read the comments to find out what it does because it will be very helpful in writing this weeks programs and for the rest of the term.

Now look at the program `Add_Cipher.cpp`. As you can see this makes and tests a class with all the necessary functions and variables for an additive cipher, and much of the code comes from the cipher.function programs you have been writing over the past several weeks. Also note the `#include "Translator.h"` command at the beginning of the program (see page 283 of Overland for a refresher on what this does).

For this week, write a similar program called `Mult_Cipher.cpp` with the corresponding encode functions for the multiplicative cipher. **Note: do not write the decode function for the multiplicative cipher.** As you may remember from the cryptology reading a few weeks ago, decoding a multiplicative cipher requires finding the key's multiplicative inverse, which we do

not yet have a general algorithm for (though we will by next week). Also note that you will not want to encode or decode with keys that have a value  $\leq 0$  in a multiplicative cipher so adjust your encode function accordingly (you can't do anything about the encode\_rand function however).