CS106L
Spring 2009

Handout #00
April 2, 2009

Course Information

Instructor Keith Schwarz

Email htiek@cs.stanford.edu. Don't hesitate to send me emails! I enjoy helping out and answering questions. Even if you have a C++ question that pertains to something we didn't cover in class, send it to me and I'll be more than happy to answer.

Office Hours While I don't have "office hours" in the conventional sense, I will stay around after lectures to answer questions. If this doesn't work out, send me an email and I can try to meet with you at another, more convenient time.

Lectures Tuesdays and Thursdays, 4:15 – 5:05 in 540-108.

Units 1 unit satisfactory/no credit. However, that's one unit jam-packed with useful C++ information and you will have a high ratio of content to units.

Prerequisites This class is designed to be taken concurrently with CS106B or CS106X, but anyone with equivalent experience should be able to benefit greatly from the material.

Readings At the beginning of each lecture I will have several handouts that should serve as the main readings for this class. However, C++ is a large and nuanced language and in the span of one quarter we simply do not have the time to explore it in its entirety. Thus, in addition to the handouts, I recommend reading some of the following books, each of which is outstanding in its own right:

Accelerated C++: Practical Programming by Example by Andrew Koenig and Barbara Moo. This introductory C++ textbook is an invaluable resource for aspiring C++ programmers and is perhaps the best introductory C++ text available. While not an exhaustive reference, *Accelerated C++* will quickly get you up to speed on professional-level C++ topics.

Effective C++, *More Effective C++*, and *Effective STL* by Scott Meyers. These books are incredibly useful and will change the way you think and program in C++. The books in the *Effective C++* series are collections of useful pieces of advice, so you can quickly navigate to relevant sections. While Meyers' books are targeted at audiences with a solid understanding of core C++ language features, they should still be quite useful in the second half of the course.

The Design and Evolution of C++ by Bjarne Stroustrup. This most excellent book by the creator of C++ provides insight into the design decisions and overarching philosophy of C++. If you're interested in how the language came to be, this is the definitive resource.

In addition to the above books, I strongly recommend picking up a C++ language and library reference book for use in this class. Stroustrup's *The C++ Programming Language* is an excellent choice.

Website

The class website is cs106l.stanford.edu and it's loaded with announcements and handouts. I will upload the lecture code and handouts after each section, so don't worry about frantically copying down the code I'm writing.

Grading

There will not be any exams or midterms in this class. Instead, there will be five short problem sets and two small programming projects. Ideally, the problem sets should take no more than an hour apiece and will be a great way to practice the material we've covered in class. The programming projects are somewhere between a quarter and half the size of a CS106 project and hopefully won't take particularly long to complete.

To receive a passing grade, you need to complete **both** of the programming projects and **four** of the five problem sets. This means that you can entirely skip an problem set without affecting your grade. Since you only need to complete four of the five problem sets, if you are unable to complete one of the problem sets on time, you can just skip it and wait for the next one to roll around. If you are concerned about missing a deadline and want to discuss extensions with me, please send me an email.

Unlike CS106B/X, we will not hold interactive grading sessions to go over your assignments. However, feel free to email me with questions or comments and if you'd like we can meet to go over the assignments in more detail.

Honor Code

This one should be pretty simple. Don't copy someone else's programs, or post the solutions to any of the assignments online where everyone else can see them. I have absolutely no problem with you collaborating and working together – in fact, I encourage you to do so – but please make a note of it on your assignment before you turn it in. If you are ever concerned about whether something is permissible, please email me and ask.

Syllabus

C++ is chock-full of features and unfortunately we will not have time to explore the language in its entirety. However, there are several key aspects of the language that I believe will best prepare you to work with professional-level C++. Below is a tentative class syllabus, which is subject to change based on how quickly we're able to move through the material:

Week 0: Intro to C++, IOStream library.
Week 1: Pointers and References
Week 2: C Strings, the Preprocessor.
Week 3: STL Containers, STL Iterators.
Week 4: STL Algorithms
Week 5: Exception Handling
Week 6: `const`, `static`, Initializer Lists, Copy Constructors
Week 7: Assignment Operators, Operator Overloading
Week 8: Inheritance
Week 9: Advanced Topics, C++0x