

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 try to 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 room 260-113.
Units	1 unit satisfactory/no credit. However, that's one unit jam-packed with useful C++ 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	CS106L has a course reader that's available in the bookstore. I highly suggest that you pick up a copy – the reader explores all sorts of interesting techniques and also includes worked examples showcasing exactly how much mileage you can get out of C++. However, the course reader is by no means a complete reference on C++. Thus, in addition to the reader, 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.

Because this is the first quarter using a course reader, there are undoubtedly a few errors that have yet to be spotted. A list of errata to the course reader can be found at cs106l.stanford.edu/errata/

Website

The course website is cs106l.stanford.edu and it's loaded with announcements and handouts. I will upload all lecture code and handouts after each section, so don't worry about frantically copying down the code we'll be writing. The website also contains errata to the course reader.

Grading

There will not be any exams or midterms in this class. Instead, there will be two problem sets and two small programming projects. Ideally, the problem sets should take no more than an hour or two apiece and will be a great way to practice the material we've covered in class. The programming projects are roughly a quarter the size of a CS106B/X project and shouldn't take particularly long to complete. To receive credit in the class, you need to complete both problem sets and both programming assignments.

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.

Tentative 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 build large software systems in C++. Below is a tentative (and ambitious) class syllabus, which is subject to change based on how quickly we're able to move through the material:

Tuesday, September 22	No Class
Thursday, September 24	Introduction to C++, Streams Recommended Reading: Chapters 0 – 3
Tuesday, September 29	Streams Recommended Reading: Chapter 3
Thursday, October 01	The Preprocessor Recommended Reading: Chapter 13
Tuesday, October 06	STL Sequence Containers Recommended Reading: Chapters 4 and 5
Thursday, October 08	Example: Solitaire Recommended Reading: Chapters 4 and 5, Chapter 13
Tuesday, October 13	STL Iterators, STL Associative Containers Recommended Reading: Chapters 6 – 8 Problem Set 0 Out
Thursday, October 15	Example: CityFinder Recommended Reading: Chapters 6 – 8

Tuesday, October 20	STL Algorithms Recommended Reading: Chapters 9 and 10
Thursday, October 22	Example: Hearts Recommended Reading: Chapters 9 and 10 Problem Set 0 Due Programming Assignment 0 Out
Tuesday, October 27	C Strings Recommended Reading: Chapters 11 and 12
Thursday, October 29	Templates Recommended Reading: Chapter 14
Tuesday, November 03	<code>const</code> , Member Initializer Lists Recommended Reading: Chapters 15 – 17
Thursday, November 05	Conversion Constructors, Copy Constructors, Assignment Operators Recommended Reading: Chapters 19 – 21 Programming Assignment 0 Due Problem Set 1 Out
Tuesday, November 10	Operator Overloading Recommended Reading: Chapters 22 – 25
Thursday, November 12	Functors Recommended Reading: Chapters 26 – 28 Problem Set 1 Due Programming Assignment 1 Out
Tuesday, November 17	Inheritance Recommended Reading: Chapter 29

Thursday, November 19	Example: A Polymorphic Function Wrapper Recommended Reading: Chapter 30
Tuesday, November 24	Thanksgiving Break, no class
Thursday, November 26	Thanksgiving Break, no class
Tuesday, December 01	Advanced Techniques: Policy Classes
Thursday, December 03	C++0x Recommended Reading: Chapters 31 and 32 Programming Assignment 1 Due