

## Course Information

---

*Thanks to Jerry Cain for the layout idea*

<b>Instructor</b>	Keith Schwarz
<b>Email</b>	<a href="mailto:htiek@stanford.edu">htiek@stanford.edu</a> . 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 try to offer some advice.
<b>Office Hours</b>	While I don't have "office hours" in the conventional sense, I will stay around after lecture 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.
<b>Lectures</b>	Wednesday from 3:15 to 4:30 in 370-370 (the CS106X lecture room, right after CS106X lecture)
<b>Units</b>	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.
<b>Prerequisites</b>	Because this is the first quarter we've taught CS106L, enrollment is restricted to students in CS106X. If you are not enrolled in CS106X but are still interested in the material, you are still welcome (and encouraged to) attend lectures and read the lecture notes. The class assumes familiarity only with the C++ concepts taught in CS106X and should be appropriate for students of almost any programming background.
<b>Readings</b>	<p>I will distribute handouts at each lecture that should serve as a good starting point for the material. However, we cannot possibly cover all of the C++ language or libraries in this class, so you should consider getting a reference book or reading online articles. If you're interested in supplemental readings, you can consider:</p> <p><i>Accelerated C++: Practical Programming by Example</i> by Andrew Koenig and Barbara Moo. This is a great book that prioritizes the material quite well and gives a good treatment of some of the most important material.</p> <p><i>Effective C++</i>, <i>More Effective C++</i>, and <i>Effective STL</i> by Scott Meyers. These books are incredibly useful and can completely change the way you program in C++. They're a bit advanced and require a pretty solid understanding of the material, but they're well-worth the read.</p> <p><i>C++: The Complete Reference</i> by Herbert Schildt. This is my personal "reference of choice," and has boatloads of information. It is a reference, not a textbook, but is a great source for complete information on various topics.</p>
<b>Website</b>	The class website is <a href="http://www.stanford.edu/class/cs106x/cs106l/">http://www.stanford.edu/class/cs106x/cs106l/</a> 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 three small programming exercises. To receive a passing grade, you need to complete **two** out of the three with passing grades. This means that you can entirely skip an assignment without affecting your grade. Please try to complete assignments on time – while I don't have a formal late policy, I most likely will not accept late assignments. Since you only need to complete two of the three assignments, if you are unable to complete one of the assignments on time, you can just skip it and wait for the next assignment to roll around. If you are concerned about missing a deadline and want to discuss extensions with me, please send me an email.

Unlike CS106X, we will not hold interactive grading sessions to go over your assignments. However, if you are interested in discussing your grade with me, we can meet after class or by appointment.

**Honor Code** This one should be pretty simple. Don't copy someone else's answers, 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.

**Errata** This is the first time we have taught this class and I am bound to make mistakes in lectures and in the handouts. I will post corrections on the class website, so make sure to check from time to time to see what's been fixed.

**Syllabus** The syllabus is subject to change based on how the class progresses. Here is our tentative goal:

**Week 0:** Course overview, IOSTream pt. I.

**Week 1:** IOSTream pt. II.

**Week 2:** STL Containers pt. I: **vector**, **stack**, **queue**, **deque**

**Week 3:** STL Containers pt. II: STL Iterators, **map**, **set**, **list**

**Week 4:** STL Algorithms

**Week 5:** C++ Classes pt. I: **const**, **static**, and initializer lists.

**Week 6:** C++ Classes pt. II: Copy constructors and assignment operators.

**Week 7:** Operator overloading, more STL Algorithms

**Week 8:** Inheritance topics