

CS 106L Course Information

Overview

C++ is one of the most popular, complex, and expressive programming languages. Over the course of 9 weeks, we will explore the modern C++ language, looking at powerful language features such as streams, the Standard Template Library (STL), templates and generic programming, smart pointers, and C++ support for object-oriented programming. We will look at features in C++11 and beyond, exploring the problems they aim to solve, the new software design patterns they bring, and the direction they are leading C++ toward.

In CS 106B/X, you learned enough C++ to be able to understand important programming concepts such as abstraction, recursion and backtracking, memory management, class design, and data structures. The Stanford libraries abstracted away the details of the C++ language not important to CS 106B/X. In CS 106L, we will understand how the Stanford libraries were designed and implemented, in the hopes that you will be able to implement similarly powerful libraries. By the end of the quarter, you will be equipped to use this language and its surrounding community as a tool to solve any meaningful problems you decide to tackle in the future!

Lecturers

Avery Wang (averywang@stanford.edu)
Office Hours: TBD

Anna Zeng (aszeng@stanford.edu)
Office Hours: TBD

Website

The course website is <http://cs106l.stanford.edu> and you should check on this regularly for updates. It will contain all sorts of useful material including lecture slides, handouts, starter code, and other interesting stuff.

Lecture

Lectures are Tuesday and Thursdays from 3:30 - 4:20 PM in 380-380C. Screencasts may occasionally be made, but we do not guarantee that all of them will be released. You are strongly encouraged to attend lecture and ask questions.

Prerequisites

The prerequisite is CS 106A or equivalent, and the corequisite is CS 106B or equivalent. We will not go into the basics of programming in C++. Instead, we will jump straight into the interesting features of C++ not explored in CS 106B/X. The course schedule is aligned with CS 106B's schedule, so you are encouraged to take this class alongside CS 106B/X.

Units	This is a one unit class, taken Satisfactory/No Credit (S/NC). Auditors are welcome, although we will not be able to grade your assignments.
Grading	You will receive credit for the class by satisfactorily completing two out of the three assignments.
Assignments	There will be three assignments over the course of the quarter of which you are required to do at least two. These assignments should be done individually. The assignments serve to provide a framework for you to practice the concepts from class and as such, are not meant to be conceptually strenuous. Our focus will be on writing elegant, functional programs that adhere to the design philosophy of C++. You will be assigned a section leader after the add/drop deadline, who will be grading your assignments and giving you feedback. Assignments are submitted via Paperless.
Late Policy	You are given two late days to use throughout the quarter.
Honor Code	All of the Honor Code policies from CS 106B apply to CS 106L. Please make sure that you are familiar with the Stanford Honor Code and how it applies to introductory computer science courses before starting the assignments. In short, you are not allowed to look at code that is not your own, or to show your code to other students (such as on a public online repository). You must indicate all sources of assistance you received. When in doubt, please ask!
Exercises	After most lectures, we will release short coding exercises to review the lecture's material. These exercises are <u>completely optional</u> , but you will have a much better grasp of the material by doing these exercises.
Readings	We will be using the CS 106L course reader, written by Keith Schwartz. We will also post numerous handouts and links to other resources which we will reference throughout the quarter. The reader focuses on pre-C++11 language features, so we will post additional handouts for concepts in C++11 and beyond. There is only so much detail we can cover in lecture so these resources will provide an opportunity to more thoroughly reinforce the concepts learned in class
Piazza + Email	We will be using Piazza for this class. Please post any questions you have on Piazza, and we will respond. Just like in CS 106B, do not post assignment code on Piazza publicly. If you have private comments, questions, or concerns, need special accommodations, or just want to talk privately, you can make private Piazza posts instead, which only Avery and Anna will see. If you have more urgent issues, please email both Avery and Anna.

CS 106L Syllabus

This is a tentative schedule that is subject to change.

Part 1: Introduction to C++

- Lecture 1: Introduction [Avery + Anna]** **9/24 (T)**
Why learn C++? What are the goals of CS 106L?
Topics: overview of the class, history and philosophy of C++, first C++ program
- Lecture 2: Streams [Avery]** **9/26 (Th)**
How do programs interact with external devices?
Topics: stringstream, buffering, state bits, i/o streams
- Lecture 3: Types and Advanced Streams [Avery]** **10/1 (T)**
Handling stream errors to implement simpio.
Topics: type inference, structures, initialization, getline, file streams, manipulators
- Lecture 4: Sequence Containers [Anna]** **10/3 (Th)**
The more powerful STL vector and its friends.
Topics: vector, list, deque, container adaptors, iterators
Assignment 1 (Graph-Viz) Released *due 10/17 (Th) 11:59 PM*

Part 2: Standard Template Library

- Lecture 5: Associative Containers and Iterators [Anna]** **10/8 (T)**
Another perspective on the map and set through iterators.
Topics: map and sets, map iterators, range-based for loops
- Lecture 6: Advanced Containers [Anna]** **10/10 (Th)**
More advanced maps, sets, and iterators.
Topics: multimap, multiset, iterator types
- Lecture 7: Templates [Avery]** **10/15 (T)**
A quest for the ultimate generic min function.
Topics: template functions, variadic templates, concept lifting, implicit interfaces
- Lecture 8: Functions and Algorithms [Avery]** **10/17 (Th)**
Manipulating containers without a single for loop.
Topics: function parameters, lambdas, algorithms, iterator adaptors, ranges
- Lecture 9: Modern STL Usage [Anna]** **10/22 (T)**
Putting it together.
Topics: summary, STL example, and discussion of good STL practices
Assignment 2 (Wiki Racer) Released *due 11/7 (Th) 11:59 PM*

Part 3: Object-Oriented Programming in C++

Lecture 10: Class Design with Iterators [Anna] **10/24 (Th)**

Designing classes that we will be using for the rest of the quarter.

Topics: C++ file organization, namespaces, type aliases, iterators, const correctness

Lecture 11: Operators [Avery] **10/29 (T)**

*Defining new meanings to +, *=, [], (), <, and <=>.*

Topics: friendship, arithmetic, subscript, functors, relational, and conversion operators.

Lecture 12: Inheritance and Polymorphism [Avery] **10/31 (Th)**

The cornerstone of object-oriented programming.

Topics: basics of inheritance, virtual functions, dynamic binding, polymorphism.

Lecture 13: Template Classes and Concepts [Avery] **11/5 (T)**

Combining generic programming with object-oriented programming.

Topics: template classes, concepts, type/value templates, templates and inheritance

Lecture 14: Special Member Functions [Anna] **11/7 (Th)**

All about constructors, destructors, and copying.

Topics: copy and assignment construction, destruction, default and deleted operations

Assignment 3 (Kd-Trees) Released

due 12/5 (Th) 11:59 PM

Part 4: Modern C++ Programming

Lecture 15: RAII, Resource Safety, and Smart Pointers [Anna] **11/12 (T)**

Smartly managing memory, and the most important and worst named C++ idiom.

Topics: exceptions, RAII, unique pointer, shared pointer

Lecture 16: Move Semantics [Avery] **11/14 (Th)**

Why copy when you can just move?

Topics: copy-elision, rvalue references, move and swap, universal references, perfect forwarding

Lecture 17: Multithreading [Anna] **11/19 (T)**

A very light introduction to C++'s parallel programming capabilities.

Topics: threads, race conditions, mutexes, atomic operations

Lecture 18: Final Lecture [Avery + Anna] **11/21 (Th)**

More cool stuff, wrap up the course, and discuss where to go next!

Topics: C++ outside of Qt Creator

No classes during Thanksgiving break, week 10, or finals week.