

Syllabus for UCLA Computer Science 35L

Software Construction Laboratory

[\[35L home\]](#)

No textbook has been selected; online material will be referred to extensively.

Topics, labs and homeworks are tentative and may be changed as the schedule progresses.

Lectures

1. Introduction, files and editing

- Multiuser and multiprocess operating systems
- GUI basics (e.g., [X](#), [Wayland](#), [GNOME](#), [KDE](#))
- CLI basics (e.g., [Bash](#), [xterm](#))
- Unix file system layout
- Everything is a file — device files
- Unix permissions
- Basic commands: [ls](#); [pwd](#), [cd](#), [mkdir](#), [rmdir](#); [echo](#), [cat](#); [cp](#), [mv](#), [ln](#), [rm](#); [chmod](#), [kill](#), [ps](#)
- Documentation and [man pages](#)
- [Emacs](#) basics: [introduction](#), online tutorial (C-h t), [help](#) (C-h ?), [basic editing](#), [directory editing](#), [running shell commands](#), [building programs](#), [Emacs Lisp](#).

2. Commands and basic scripting

- Unix wildcards, basic regular expressions
- More advanced commands (e.g., [grep](#), [find](#))
- Pipelines and redirection
- Simple shell scripting
- Idea of interpreted languages

3. More scripting, VMs, and construction tools

- Basics of [Python](#)
- [Java](#) as a compromise between interpreted and compiled languages
- Building from source
 - [make](#)
 - [automake](#) and [autoconf](#)

4. Change management

- [diff](#) and [patch](#)
- Basics of Makefiles
- Version control systems, e.g., [Git](#), [Subversion](#)

- retrieving a tree to build and install
- committing a change
- dealing with merge conflicts

5. Low-level construction and debugging

- The C compilation and linking process
- Introduction to C
- Debuggers and debugging tools, e.g., [GDB](#), [Valgrind](#), [strace](#).

6. Systems programming

- C and system programming
- Library calls vs. system calls

7. Faults, failures, errors, and holes

- Ways in which a program can go wrong
- Buffer overruns, and techniques for avoiding them
- Ken Thompson, [Reflections on Trusting Trust](#) (1984). In 2003 Jon Hall was reported to have said that the paper is not a theoretical speculation.

8. Security basics

- Threats, including eavesdropping, tampering, forgery, and denial of service
- Authentication, authorization, and accounting
- Chains of trust
- Firewalls, kernels, and sandboxes
- Intrusion detection
- Backups
- Security policies

9. Parallelism

- [SIMD versus vector processing versus MIMD](#)
- Processes versus threads
- Synchronization
- [POSIX threads](#)
- [OpenCL](#)
- [OpenMP](#)
- Clusters, massive parallelism, grids, and clouds

10. The crystal ball

- Trends in computing research and development
 - Things to increase your chances at grad school: research, 199's, etc
 - Other general tips for excelling in upper division classes
-
-

