CS 18: Software Design Practicum — Spring 2026

Course Syllabus

Information At-A-Glance

Instructor

Name:	**Prof. Ordentlich**
E-mail:	eordentl@caltech.edu
Office:	ANB 122
Office Hours:	By private meeting

Course Email

cs018@caltech.edu

All course-related email goes here.

Course Website

https://cs18.sof.tware.design

Visit early. Visit often.

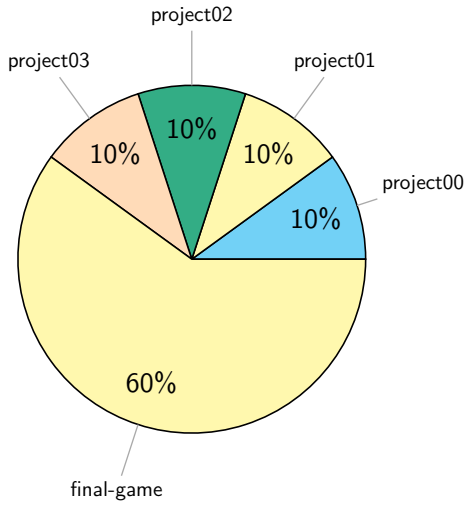
Course Overview

Prerequisite: CS 3x or instructor permission

CS 18 is a practical introduction to designing large programs in a low-level language. Heavy emphasis is placed on group work, documentation, testing, and software architecture. Students will work in groups throughout the term to create their own physics engine and open-ended game using the C programming language. Software engineering topics covered include code reviews, testing and testability, code readability, API design, refactoring, and documentation. Prior experience with the C programming language at the level of CS 3x is required and assumed.

Assessments

The course is divided into two phases: incrementally developing a physics engine, and developing a game using that physics engine. At the beginning of the term (in the OM), you will be placed into groups of two or three, which will stay together for the entire term.



Physics Engine Projects

Each physics engine project consists of three components: some "engine" features, some "game" library features, and a "demo" applying them. For each project, you will then combine each member's knowledge to design and implement an additional "creative component".

Some parts of these projects will be autograded via a set of automated tests. Other parts will be graded on various criteria that include correctness, code quality, and an in-person code review with all group members.

Game Project

After completing the physics engine and game library projects, you will spend the bulk of the term building on these for an open-ended game project. We will give more details on this closer to the release date.

Collaboration & Academic Integrity

See our "collaboration table" on the website. We reserve the right to modify or clarify this policy as needed. Notably, you may not, under any circumstances, look at another group's code or write pseudocode with another group.

Late Policy

We've designed a late policy that is very intentional in providing flexibility while also making sure groups do not fall so far behind that they might not be able to pass. Please see below for details. Extensions are taken as a group.

`https://caltechcs.fillout.com/extensions`

Please do not e-mail us asking for an extension. You must use the website form. There are limits and restrictions on extensions which the form outlines. Please read these before you need an extension! This flowchart outlines common scenarios in which a student may need an extension: