

## Course Syllabus

### Information At-A-Glance

Instructor	
Name:	<b>Adam Blank</b>
E-mail:	blank@caltech.edu
Office:	ANB 115
Office Hours:	By <a href="#">private meeting</a> .

Course Website
<a href="https://sof.tware.design">https://sof.tware.design</a> Visit early. Visit often.

Lecture
ANB 105 on MW 01:00 PM – 01:55 PM

### Course Overview

This course covers large program design in the C programming language. Prerequisite: CS 2. CS 3 is a practical introduction to designing large programs in a low level language. Heavy emphasis is placed on documentation, testing, and software architecture. Students will work in teams on two 5-week long projects. In the first half of the course, teams will focus on testing and extensibility. In the second half of the course, teams will use POSIX APIs, as well as their own code from the first five weeks, to develop a large software deliverable. Software engineering topics covered include code reviews, testing and testability, code readability, API design, refactoring, and documentation.

### Course Components

In this course, there are several types of assessments: diagnostics, two group projects (physics engine for the first half and a game for the second half), and a solo project throughout the term.

#### Diagnostics

Throughout lectures, we ask poll questions which will (together) serve as a single “diagnostic”.

While we will “grade” each poll question asked in lecture on a 0-1 basis (each being worth a fraction of the total 1 point for a diagnostic), you will be allowed a second attempt (for full credit) for each diagnostic. Note that you will be required to explain all of your answers for each poll question on the second attempt (even if you got some answers correct on the first attempt). For the second attempt, you can collaborate with students *but you may not ask for help from course staff in office hours*. The second attempts from the previous week will be due at *11:30 PM on Mondays*.

#### Group Projects

At the beginning of the term, you will form a group of three (who must all be in the same section on REGIS) which will stay together for the entire term. If you need help finding group members, please contact one of the Head TAs or Prof. Blank.

#### Physics Engine Group Project

Each part of this project in the course will consist of three components: the physics engine, a demo, and a potential feature of the game. Each student in a team will be responsible for **one** of the components every week. The component that each student is assigned to will rotate weekly. The projects are the heart and soul of this course. We prefer the term *project* to *set* because all the individual parts of the assignment will come together to create a single finished product that we hope you will be proud of. The projects will be exclusively programming assignments. Some parts will be auto-graded via sets of unit and integration tests we have designed to catch most of the common mistakes. Your final grade on each project will be determined after

an in-person “team meeting” with one of the TAs. Additionally, each student will be graded individually on the completion and functionality of their assigned component that week. All code for these projects will be due at *11:30 PM on Fridays*.

After the three components of each project are completed, the team is expected to come together to submit answers to several questions via an embedded survey on the website. These answers will be due at *11:30 PM on Mondays*.

## Game Group Project

For the last several weeks of the term, we will transition from the physics engine project to an open-ended game project. We will give more details on this when it’s closer to the release date.

## Solo Project

Throughout the term, you will iterate on a code base for a web server. These parts will be completely solo with absolutely no collaboration of any kind. In the first week, you will write some code to orient yourself to C; in the second week onwards, you will add features weekly to a codebase we’ve started for you.

## Team Meetings

We will assign your group a ten minute slot within their REGIS section during which *all of your group members* must meet with a TA for a discussion of the previous week’s project. There will be no team meetings during the first week of the course.

## Late Policy

For the group projects (engine, game, and demo components), the late policy does **not** apply. If your group faces extenuating circumstances and requires an extension of any kind, you must talk to Professor Blank, who will grant an extension on a case-by-case basis. Otherwise, the following late policy (which is the same as in CS 2) applies to the solo project:

Throughout the term, we will keep track of a “count” for you. The “count” represents how much grace you’ll have for turning in projects late. Diagnostics must always be turned in on time or not at all. Your grace count starts at zero, but turning in assessments early or late can change it in the following ways:

- For any *solo project* that is submitted early (with all tests passing), your count increases by  $\text{Math.floor}(\frac{\text{hours early}}{24})$ .
- For any *solo project* that is turned in on time, your count remains unchanged.
- For any *solo project* that is submitted after the due date, your count decreases by  $\text{Math.ceil}(\frac{\text{hours late}}{24})$ .

Note, your last submission is the one used to calculate your grace count. Moreover, your grace count is subject to the following additional rules:

- Under **normal circumstances**, the count cannot go below 0. This means that if turning in an assignment would make your count go negative, you cannot turn it in.
- If you have **extenuating circumstances** or CASS accommodations that might entitle you to go below 0 and still turn in an assignment, you must request the extension **at least 24 hours before the normal due date** by e-mailing Prof. Blank. TAs (including the Head TAs) are not allowed to grant extensions. Note that extensions **do** affect your count, as you are turning in an assignment late.
- Under **absolutely no circumstances** can the count go lower than -5 or larger than 7 during the course. This includes all circumstances, even illness and family emergency.

Note that in extreme cases such as deaths in the family, you will still be limited to the restrictions above if you choose to complete the course normally. However, we find in such circumstances, it’s often the case

that I grades are appropriate instead, which we are happy to work with you on. To potentially receive an I grade, you should contact the Deans's Office!

Note that, regardless of your grace count, office hours help will **only** be available for the **current** projects.

## Grading Scheme

### Diagnostics are a “threshold” on your letter grade

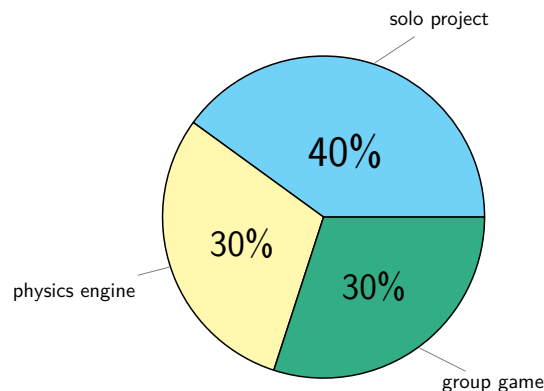
Since CS 3 is not P/F, we expect you to do all of them. The percentage of available diagnostics will limit your final letter grade using the following thresholds:

A+	≥ 95%
A	≥ 88%
A-	≥ 80%
B+	≥ 75%
B	≥ 70%
B-	≥ 65%
C+	≥ 60%
C	≥ 50%
C-	≥ 45%
D	≥ 40%
F	≥ 0%

If you **do not get the percentage of diagnostic points** corresponding to a given letter grade, you **cannot get that letter grade**, regardless of your grade based on the other components of the class. For example, if your final diagnostic grade is 85%, then an A- is the highest grade you can receive.

### Grade Percentages

After we've applied the threshold, other parts of the course will have the following weights:



## 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 student's/group's code or write pseudocode with another group.