



# About

Welcome to CS 61BL

Course Format

Assignments

Grading

Resources

Collaboration and Cheating

A Parting Thought

## Welcome to CS 61BL

The CS 61 series is an introduction to computer science, with particular emphasis on software and machines from a programmer's point of view. In CS 61A, students are exposed to a diverse set of mental models for computational problems and solutions through programming paradigms like functional programming, object-oriented programming, and declarative programming. In CS 61BL, we refine those mental representations by focusing particularly on the efficiency of writing programs (design) and running programs (runtime).

## Hours and Workload

This is a summer course, which means it is run at twice the speed of a normal semester. In addition, CS 61B naturally has more involved programming than CS 61A, and CS 61BL has even more, so expect there to be a lot of work. **We strongly recommend not taking any other courses alongside CS 61BL.**

While we can't predict how many hours you will spend on the course, here is a brief breakdown of what you will have to do in this course.

- 12 hours per week in lab.
- One two-hour lecture per week.

- Four exams in the 8 weeks of the course.
- Three projects in the 8 weeks of the course.

The first couple weeks will be lighter in comparison to the latter half of the course.

## Prerequisites

CS 61A is an important prerequisite. We expect to build heavily on data-oriented and object-oriented design approaches introduced in this course and on algorithms for recursive list and tree manipulation. Engineering 7 students may find the beginning of the course to be a bit scarier, particularly when it comes to object-oriented programming and recursion. We assume you are coming in with zero Java experience, but we will move through basic Java syntax very quickly.

We recommend all students to complete the optional introductory assignment to get comfortable with Java and practice some of the programming skills expected by this class.

## Alternatives

This is a course about data structures and programming methods. For those who may have already had a data structures course and simply want to learn Java, self-study may be a better option. The self-paced center offers CS 9G: Java for Programmers, that will teach you more of what you want to know in less time. Students with “sufficient partial credit” in CS 61B should also consider taking CS 47B to complete the CS 61B requirement without taking the full course.

## Course Format

### Labs

Lab sections meet every day Monday through Friday, except Wednesday. You are expected to attend lab whenever your lab section meets. All lab work must be done with a partner.

If you would like to attend a lab section that you are not officially enrolled in, you must receive permission from the lab TA of the section you would like to attend. Historically, the afternoon labs (including 2-5 pm and 3-6 pm labs) have the highest attendance, so it's not likely that you will find a seat there. Switching sections will not be permitted following the first week of class except under exceptional circumstances.

### Lectures

There is one 110-minute lecture 3-5 pm every Wednesday except for July 4. Our midterm exams will be held during lecture in the same lecture room. We expect most of the learning will be done in lab, so attendance at lecture is not required except for when there is an exam.

## Instructor Office Hours

Instructor office hours are primarily for short questions and administrative problems, but we're happy to make appointments by email for longer periods of time. The instructors are teachers too: we like teaching, and we'd rather see you as soon as you don't understand something than right before the exam. If you don't find us in our scheduled office hours room, it'll usually be because we're helping students in lab next door.

In a lab-based course like CS 61BL, we expect that most of the course-related questions will occur in lab or our online forum, Piazza. For administrative inquiries like enrollment and DSP accommodations, email the course's administrative address, [cs61bl@berkeley.edu](mailto:cs61bl@berkeley.edu).

## Assignments

### Lab Exercises

Monday and Tuesday lab exercises are due 11:59 pm Wednesday, while Thursday and Friday lab exercises are due 11:59 pm Saturday. Labs exercises and self-assessments will be graded via online submission to Gradescope, and you may submit as many times as you like before the deadline. The last lab of each week will additionally require you to fill out a self-reflection form about the labs in the past week.

### Quizzes

Each lab section will start with a short written quiz intended to help students and staff identify areas for improvement. Quizzes are normally completed individually with feedback given in lab.

You can only take the quiz in your own lab section. Because of how quiz points are capped, you can miss several quizzes without detriment to your overall quiz grade. There are no makeup quizzes.

### Projects

There are three projects in this course. Projects are larger and more challenging assignments than you would typically find in a lab. All projects are to be done with your partner.

than you would typically find in a lab. All projects are to be done with your partner.

Project 1 will be relatively easier than projects 2 and 3, taking less time and with greater levels of scaffolding. Project 2 will be a very difficult project (on par with what you might expect from Hilfinger's harder projects). Project 3 will be challenging, but not as time consuming as project 2.

Each project has a specific theme:

- Project 1: Introduction to Java
- Project 2: Large Scale Design and Testing
- Project 3: Working with an Existing Code Base

For all projects, we will ultimately release all tests that determine your grade. Passing all tests on Gradescope will allow you to earn full points for the autograded portion of the project.

## Exams

This course has three midterm exams and a fourth, final exam. The three midterm exams will be held in lecture 3-5 pm Wednesday, June 27, July 11, and July 25. The final exam will be 6-8 pm Thursday, August 9.

Inspired by the great Paul Hilfinger tradition, exams may cover any material whatsoever. For fear of our lives, exams will almost exclusively test material covered in the course.

**There will be no alternate exams.** If you miss an exam, your score will be reweighted with your performance on the final exam according to the exam supersession policy. Students with DSP accommodations that require alternate exam timing will be honored, so long as you can make a time that overlaps the official time. If you have a disability that prevents your ability to make such a time, we will discuss alternate arrangements with you directly. If you are traveling on official UC business, and have a proctor available, we will allow remote exams to be taken at the same time as the official exam.

If you have an exact time conflict with two or more midterm exams, or if you are unable to make the final exam, we suggest you seek accommodations with the conflicting party, or take CS 61B next semester. If, for whatever reason, you still need to take this course this semester, email [cs61bl@berkeley.edu](mailto:cs61bl@berkeley.edu) by the first week of the course explaining your needs. We're happy to accommodate, but expect that most students will need to take an incomplete grade and complete the exams the next semester as there will be no alternate exams.

### Exam Supersession

For those of you who miss an exam, have a bad night, or make major improvements over the

semester, the exam supersession policy gives you a chance to replace one of your midterm exams. Specifically, if it helps your score, we will replace one of your midterm scores by its “final statistical equivalent” (FSE). We compute the FSE of an exam as follows:

Let  $F$  be the number of standard deviations above the mean that you score on the final. For example, if you are 0.3 standard deviations below the mean,  $F = -0.3$ . Let  $\overline{M}$  be the class-wide mean (not including zeros) on a midterm,  $M$ . Let  $\sigma_M$  be the class-wide standard deviations (not including zeros) on the same midterm  $M$ . Your FSE for that exam is  $\sigma_M F + \overline{M}$ .

If the FSE of midterm  $M$  is better than your original midterm  $M$  score, we will replace the original score with the FSE instead. If the FSE of all three midterm exams is better than each original score, then we will replace the one exam that gives you the greatest overall benefit. If all of the FSE are worse, nothing happens, so doing badly on the final won't hurt your earlier exam scores.

## Grading

Your course grade is computed using a point system with a total of 300 points.

Category	Percentage	Points
Labs	20%	60
Quizzes	5%	15
Effort, Participation, Altruism	5%	15
Project 1	5%	15
Project 2	12%	36
Project 3	8%	24
Midterm Exams (3)	25%	75
Final Exam	20%	60
Total	100%	300

Category	Percentage	Points
Grade breakdown		

Your final score will be rounded to the nearest integer before being converted to a letter grade. Each letter grade for the course corresponds to a range of scores:

A+	A	A-	B+	B	B-	C+	C	C-	D+	D	D-	F
295	280	265	245	225	205	195	185	175	170	165	160	0

Grade bins

**There is no curve.** Your grade will depend only on how well you do, and not on how well everyone else does. These bins were designed based on past semester student performance with the intention of complying with [departmental grading guidelines](#).

Incomplete grades will be granted only for dire medical or personal emergencies that cause you to miss the final, and only if your work up to that point has been satisfactory. You must complete all coursework before the drop deadline to be considered for an incomplete grade.

There are a total of 24 graded lab exercises, each worth 3 points. Thus, while the total point value for labs is 72 points, lab grades are capped at 60 points. This means that you can miss up to 12 points on labs without penalty. In other words, getting more than 60 points on lab is equivalent to getting 60 points for your lab score. A similar idea will be applied to quizzes, which are each worth 1 point.

## Late Policy

For each project, you'll also receive 24 hours of slip time, during which there is no penalty. Unused slip hours do not earn you points, nor can they be applied to later projects. This is not quite the same thing as extending the deadline by 24 hours, because you'll also be required to include a file called "explanation.txt" in your project directory that explains why you're late. The goal with this is to collect information about what makes students submit projects late, even if it's as simple as getting started late.

For each hour a project is late after slip hours, you'll incur a penalty of 5/12 percent (0.417%), rounded off in some unspecified fashion. This means that you will lose roughly 10% of the points for a project each day late (though tracking is by hours late, not days). The maximum lateness penalty for the autograded portion of the project is 50%. This means that

you can submit the project any time between about 5 days after slip hours and the last day of instruction for up to 50% credit.

## Extensions

**If you're hurting from academic or personal stress, or need more time on any assignment, please contact a lab staff member immediately!** The intention of these assignments is for you to learn, so if you need extra time we will be willing to grant an extension.

Students who normally receive additional time under their DSP accommodations should email [cs61bl@berkeley.edu](mailto:cs61bl@berkeley.edu) within the first week.

## EPA

A small portion of your grade in this course will be based on **effort, participation, and altruism**, which can be earned by consistent engagement in lab activities, collaborating with lab partners in equal parts, and helping others in lab or on Piazza, as well as creating software, art, or tutorials that help others learn, and many more. Extraordinary contributions to the course community may be rewarded with extra credit. EPA scores will be kept internal, but our intent is for full EPA credit to be easily attainable for hard-working, collaborative students.

## Resources

### Online Forum

Our discussion forum this semester will be [Piazza](#). For most questions about the course, Piazza is the right place to ask them. The course staff read it regularly, so you will get a quick answer. Furthermore, by posting online as opposed to emailing us directly, other students benefit by seeing the question and the answer.

### Staff Email

The email address [cs61bl@berkeley.edu](mailto:cs61bl@berkeley.edu) will send a message to the instructors. You can use it for correspondence that you don't want to send to Piazza. If you send a question that is of general interest, we may post the response on Piazza. (We will keep personal information out of it, of course.) **Please do not email the instructors directly for exceptions.**

## Reading

You are expected to read each lab's contents. Additional optional reading assignments will be posted on the course calendar. The first text we use is Josh Hug's [free, online course notes](#). If you find these notes insufficient, you might consider consulting Paul Hilfinger's (free) [Java Reference](#) or Head First Java, 2nd Edition by Sierra and Bates (O'Reilly, 2005). The optional textbook for the latter half of the course is Algorithms, 4th Edition by Wayne and Sedgewick.

## Collaboration and Cheating

Plagiarism on any lab or project will result in a negative score on that assignment. A second instance of plagiarism on a homework, lab, or project will result in an F in the course. All incidents of plagiarism will be referred to the Office of Student Conduct, including carelessly leaving code up on GitHub.

Deadlines can be stressful, and we know that under extreme pressure, it becomes tempting to start rationalizing actions that you would otherwise would consider inappropriate. If you find yourself in this situation, please talk to a staff member immediately! See the Extension policy above.

During the Spring 2017 semester, we compiled a series of [incident reports written by students who were caught plagiarizing](#). If you find yourself tempted to cheat, you might turn to the words of others who have made the wrong choice for guidance.

For labs and projects, if you commit some act that is unreasonable but bring it to the attention of the course staff within 72 hours, we may impose local sanctions such as assignment deductions, but the course will not refer the matter for further disciplinary action except in cases of repeated acts.

### Lab Collaboration

The entire point of labs is to learn, so we emphasize active learning and pair programming. For labs, feel free to collaborate with others however you choose, though keep in mind that greater independence between pairs is likely to give you a better learning experience, as long as you aren't totally stuck. Even though we will allow close inter-pair collaboration on labs, **the code you share with your partner should still be your own work!**

### Project Collaboration

By contrast, the projects were designed not just for learning (particularly how to be self-reliant in the context of large unfamiliar systems), but also for the dual purpose of evaluating your mastery of the course material. As such, they are intended to be completed primarily



independently of any other pair, particularly when it comes to writing actual code, although we encourage you to discuss high-level strategies, specific syntax issues, and solutions to bugs.

## Exam Cheating

**For exams, we will be absolutely unforgiving.** Any incident will result in a failing grade for the course, though Berkeley will let you retake CS 61B next semester. All incidents of exam cheating will be referred to the Office of Student Conduct.

## Academic Honesty

**The golden rule of academic dishonesty is that you should not claim to be responsible for work that is not yours.** To help (but not entirely define) the bounds of acceptable behavior, we have three important rules for projects:

- **By You (and Your Partner) Alone:** All project code that you submit (other than skeleton code) should be written by you and your partner alone, except for small snippets that solve tiny subproblems (examples in the Permitted section below).
- **Do Not Possess or Share Code:** Before you've submitted your final work for a project, you should never be in possession of solution code that you (or your partner) did not write. You will be equally culpable if you distribute such code to other students or future students of CS 61BL (within reason). **DO NOT GIVE ANYONE YOUR CODE! EVEN IF THEY ARE DESPERATELY ASKING. DO NOT POST SOLUTIONS TO PROJECTS ONLINE (on GitHub or anywhere else)!** If you're not sure what you're doing is OK, please ask.
- **Cite Your Sources:** When you receive significant assistance on a project from someone else, you should cite that assistance somewhere in your source code with the `@source` tag. We leave it to you to decide what constitutes "significant".

## Permitted

- Discussion of approaches for solving a problem. Giving away or receiving significant conceptual ideas towards a problem solution. Such help should be cited as comments in your code. For the sake of others' learning experience, we ask that you try not to give away anything juicy, and instead try to lead people to such solutions.
- Discussion of specific syntax issues and bugs in your code.
- Using small snippets of code that you find online for solving tiny problems such as code for finding the distance between two points. Such usages should be cited as comments in your code.

## Permitted with Extreme Caution

- Looking at someone else’s project code to assist with debugging. Typing or dictating code into someone else’s computer is a violation of the “By You Alone” rule.
- Working on a project alongside another person or group of people. Your code should not substantially resemble anyone else’s!

## **Absolutely Forbidden**

- Looking at someone else’s project code to understand a particular idea or part of a project. This is strongly discouraged due to the danger of plagiarism. We are very serious about the “By You Alone” rule!
- Possessing project solution code that you did not write yourself or another student’s project code in any form before a final deadline, be it electronic or on paper. This includes the situation where you’re trying to help someone debug. Distributing such code is equally forbidden.
- Posting solution code to any assignment in a public place (e.g. a public git repository, mediafire, etched into stones above the Mediterranean, etc). This applies even after the semester is over.
- Working in lock-step with other students. Your workflow should not involve a group of people identifying, tackling, and effectively identically solving a sequence of subproblems.

You can obey the letter of this entire policy while completely violating its spirit. However, this policy is not a game to be defeated, and such circumventions will be seen as plagiarism.

## **A Parting Thought**

Grades and penalties aren’t the purpose of this course; we really just want you to learn and be successful in the course.

The entire staff is very excited to be teaching CS 61BL this semester and we’re looking forward to meeting such a large and enthusiastic group of students. Welcome to CS 61BL!

## **Acknowledgements**

Some course information ideas derived from [Paul Hilfinger’s CS 61B handout](#), [Josh Hug’s CS 61B syllabus](#), [Brian Harvey’s CS 61A handout](#), and [David Malan’s CS50](#).