

CS 61B

Course Info

This document provides an overview of the policies for this course as taught at UC Berkeley.

General Background Information

These course policies are pretty long. With the exception of the collaboration policies we're not expecting you to read the entire thing in one sitting. However, you'll hopefully find the answers to any logistics questions you may have somewhere in this document.

Welcome to CS 61B

The CS 61 series is an introduction to Computer Science, with particular emphasis on software and machines from a programmer's point of view. CS 61A covered high-level approaches to problem-solving, providing you with a variety of ways to organize solutions to programming problems as compositions of functions, collections of objects, or sets of rules. In CS 61B, we move to a somewhat more detailed (and to some extent, more basic) level of programming by focusing particularly on the efficiency of writing programs (design) and running programs (runtime).

Prerequisites

This class assumes you have taken CS 61A, CS 88, or E7, or have equivalent background to a student who has taken one of these courses. The course is largely built upon the assumption that you have taken CS 61A. CS 88 and E7 students may find the beginning of the course to be a bit scarier, particularly when it comes to object oriented programming. We assume you are coming in with zero Java experience, but we will move through basic Java syntax very quickly.

If you already have Java experience, great! We hope that you'll help out your fellow students in discussion, lab, and on our class forum, particularly in the opening weeks when everyone is catching up on Java.

Alternatives

This is a course about data structures and programming methods. It happens to also teach Java, since it is hard to teach programming without a language. However, it is **not** intended as an

exhaustive course on Java, creating Android apps, user interfaces, graphics, or any of that fun stuff.

Some of you may have already taken a data structures course, and simply want to learn Java or C++. For you, self-study may be a better option.

INFO

Finally, the 1-unit self-paced course CS 47B is for students with “sufficient partial credit in 61B,” allowing them (with instructor’s permission) to complete the CS 61B course requirement without taking the full course. The 47B guide is at [this link](#).

Mental Health and Wellness

As a student you may experience a range of issues that can cause barriers to learning, such as strained relationships, increased anxiety, alcohol/drug problems, depression, difficulty concentrating and/or lack of motivation. These mental health concerns or stressful events may lead to diminished academic performance or reduce a student’s ability to participate in daily activities. UC offers services to assist you with addressing these and other concerns you may be experiencing. If you or someone you know are suffering from any of the aforementioned conditions, consider utilizing the confidential mental health services available on campus. We encourage you to reach out to the Counseling Center for support. An on campus counselor or after-hours clinician is available 24/7.

The National Suicide Prevention Lifeline is a 24-hour number any student or faculty/staff person can call to speak with someone about suicide: (800) 273-TALK (8255).

More resources can be found under our [Resources tab](#).

If you’re experiencing extenuating circumstances impacting your mental health and would like to meet with a member of course staff to work out accommodations within the course, or would like to be connected to more resources, you can set up a meeting with us [here](#)

Accommodation

UC Berkeley is committed to creating a learning environment that meets the needs of its diverse student body including students with disabilities. If you anticipate or experience any barriers to learning in this course, please feel welcome to discuss your concerns with the instructors.

If you have a disability, or think you may have a disability, you can work with the Disabled Students’ Program (DSP) to request an official accommodation. The Disabled Students’ Program (DSP) is the campus office responsible for authorizing disability-related academic

accommodations, in cooperation with the students themselves and their instructors. You can find more information about DSP, including contact information and the application process [here](#). If you have already been approved for accommodations through DSP or are working through the process, please schedule a meeting with course staff so we can develop an implementation plan.

Course Format

Lecture

Lecture is scheduled on Monday, Wednesday, and Friday from 1-2 PM in Dwinelle 155. Since the classroom is not large enough to allow all enrolled students to physically attend, lecture will be simultaneously webcast over Zoom.

We will post lecture recordings as well as Josh Hug's Spring 2023 web videos to the content schedule on the main page. Both will mostly cover the same content.

Discussion

Discussion sections are 1 hour long and feature a worksheet with problems reviewing the material learned in the past week. In discussion sections, TAs will help students review the material and solve the problems on the worksheet. Note that your TA may not cover all the problems on the worksheet, and that's intentional - you can review the remaining problems later with our posted solutions and walkthroughs.

Discussion sections will primarily be held in-person, though we may have limited remote options on Zoom.

Information about the staff running each section can be found on the staff page. We will not have section signups this term.

The schedule of all sections and lectures can be found at the bottom of our course website's [main page](#).

Discussion attendance is not mandatory, though you are given [pacing points](#) for attending.

We have three (3) different types of discussions this semester:

Bridge (2hr)

Do you want more of an emphasis on the concepts before diving into the discussion worksheet? If so, our bridge discussion may be the right place for you! Some of our amazing TAs will lead a two-

hour discussion section intentionally organized for students who are not as familiar with the material that week. It's a great way to get a little more comfortable with the topics and receive more individualized attention during the section. The material and difficulty covered during the bridge discussion section will be the same as the regular discussion section, and we'll have an extended mini-lecture to reinforce conceptual understanding.

Regular (1hr)

This section is best for students who are caught up with lectures and are ready to tackle some medium-difficulty questions. We recommend attending this section if you are unsure which to attend.

Exam-Level (1hr)

Exam-level sections are faster-paced sections that cover a different worksheet with more difficult problems. The goal of these sections is to get comfortable with exam-level problems and problem-solving strategies. In general, there will be less focus on review/mini-lecture and more time spent on the problems.

Lab

Lab sections are 2 hours long and feature coding assignments that students submit for credit. In these sections, TAs will introduce the assignment, review relevant material, and answer students' questions.

Labs will primarily be held in-person, though we may have limited remote options on Zoom.

Lab attendance is not mandatory, with the *exception* of one lab - Project 3 Checkoffs. This lab includes an assignment for which you will need to get checked-off by a TA to receive credit.

Information about the staff running each section can be found on the staff page.

We will not have section signups this term. For the mandatory attendance lab(s):

- Project 3 Checkoffs: There will be official sign ups for this checkoff, in which you and your partner will present the work you completed for Project 3. The signups will be available shortly before the last week of classes.

Office Hours

In Office Hours, you can get help from our staff with the different assignments, exam preparation, logistical matters, and any advice you may need. We will hold both in-person and online office hours. You may also use lab sections to ask questions unrelated to lab.

We will hold Office Hours over Zoom. To learn more about how CS 61B Office Hours work, please read our [Office Hours Guide](#). We will use the online [Office Hours Queue](#) to keep track of students in Office Hours. Staff will always skip tickets on the queue that do not adhere to our Office Hours policies.

Resources

Website

The [course home page](#) will provide one-stop shopping for course information. The course schedule as well as all handouts, homework, labs, FAQs, etc., will be posted there.

Discussion Forum

Our discussion forum this semester will be Ed Discussions. For most questions about the course, Ed is the right place to ask them. The course staff reads 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. Don't forget to check Ed before asking your question, just in case someone else has already posted it. If you have a question about something pertaining to your own code that shouldn't be shared with the class, or if you have a question about a personal matter, you can make a private post on Ed, which will only be seen by the instructors and staff.

Please read our [Ed Guide](#) and [Policies](#). We will only respond to questions that adhere to our policies of using Ed.

Staff Email

The e-mail address cs61b@berkeley.edu will send a message to the course staff (instructors and the head TAs). You can use it for correspondence that you don't want to seen on our class forum. The head TAs and instructors all read it, so you will usually get a reply within a few days. If you send a question that is of general interest, we may post the response on Ed (we will keep personal information out of it, of course). If you have any problems that require an exception to course policy (e.g. medical emergencies or sudden necessary travel that result in extended absences), please contact cs61b@berkeley.edu. **Please do not email the instructors or course staff for exceptions. Email cs61b@berkeley.edu.**

Beacon

Rather than using bCourses, we will be using our own custom learning management system called Beacon located at [beacon.datastructur.es](https://sp24.datastructur.es). You can use Beacon to keep track of your grades and

late assignments. You can also read our full guide on Beacon [here](#).

Reading

There is no required textbook for the class.

There is an online textbook written by myself and a large team of course staff. It can be found at <https://cs61b-2.gitbook.io/cs61b-textbook/>. 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 \(O'Reilly, 2005\)](#). These are not required for the course. The **optional** textbook for the weeks 5-14 of the course is Algorithms, 4th Edition by Wayne and Sedgewick.

The official description of the Java core language is available online in [The Java Language Specification](#) (Java SE 17 Edition) by James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley, Daniel Smith, and Gavin Bierman. It's extremely thorough and precise, at the expense of being quite dense and technical. You will find the official [Java 17 documentation](#) to be useful as well

Software

This official coding environment and text editor for the course is the Integrated Development Environment (IDE) called IntelliJ. While we will not enforce using IntelliJ, we strongly recommend it. We will not officially support any editing or programming environment other than IntelliJ.

This semester, we will use Java 17 or higher (and therefore require at least IntelliJ 2021.2.1, though we recommend IntelliJ 2022.3 or IntelliJ 2023.2).

You will be able to do any work you'd like on any Windows, Mac OS X, or Linux computer. You will set up your own computer in Lab 1.

We'll be using the version-control system [Git](#) this semester with private repositories on [GitHub](#). Version-control systems allow you to maintain a series of "snapshots" of your files at various points in their development. Used properly, this provides you some back-up protection, so that you can recover previous states of your work when something goes wrong. Also for team-oriented projects (as well as in the real world), version-control systems help manage collaborative work.

Coursework

There are five required aspects of the course for which you earn points:

1 Weekly Surveys

- 2 Lab Assignments
- 3 Homeworks
- 4 Projects
- 5 Exams

Weekly Surveys

While lecture and section attendance is not required, nor even expected, we do expect you to stay up to date with material. To help us keep track of your progress and sentiment about the course, there will be 12 weekly surveys due on Mondays at 11:59 PM.

Submitting 10 weekly surveys will receive full credit (i.e. you can skip 2). Submitting more will not yield extra credit.

Lab Assignments

During Phase I of the course (Weeks 1 through 5), labs will provide you with help getting your computer set up and teach you how to use essential Java programming tools, and will also include a peer code review after we're done with Project 1.

During Phases II and III of the course (Weeks 6 - 15), labs will usually involve implementation of some data structure or algorithm described in lecture. All labs should take no more than two hours to complete, though some may run slightly longer. You will turn in everything electronically using GitHub, and your results will be available on Gradescope.

Labs are graded according to autograder tests on Gradescope. There are no hidden tests for which your score is not displayed, though we may have tests which you do not have access to.

Out of the 14 labs, only 10 will entail an assignment you need to submit. Each of these 10 labs will be worth 5 points (for a total of 50 points).

You do not need to attend lab to receive credit for lab, with the exception of one specific assignment (see [above](#)).

This semester, we are **not** allowing lab partners for lab assignments. We do encourage you to discuss ideas (see the [Lab Collaboration policy](#)), but the code you submit should be your own.

No labs will be dropped, though we have a generous lateness policy.

Homeworks

There are 5 required homework assignments in the course.

We have three kinds of homework:

- Programming homeworks (HW0, HW2), which will be similar to labs.
- HW1 is a homework about being a good classmate in CS courses. It is intended to set the norms for the course, and remind us as to how we can respect and include all members of the Berkeley community in CS 61B.
- Conceptual assignments due shortly before midterm 2 (HW3) and the final (HW4) to help you study for the exams. They will be available on Gradescope and include a combination of multiple choice and fill-in-the-blanks questions. You can expect to have to work out these problems on paper before filling in your answer on Gradescope.

Homeworks 0, 1, 3, and 4 are worth 15 points. Homework 2 is more challenging, time-consuming assignment and will be worth 30 points.

No homeworks will be dropped, though we have a generous lateness policy.

Projects

In addition to the HWs and labs, there will be 4 programming projects. In these projects you will build an entire system. All projects except project 3 must be completed on your own.

Project 0 and 1 will be relatively easier than projects 2 and 3, taking less time and with greater levels of scaffolding. Project 2 will be a larger scale design project with very little skeleton code. Project 3 will be a capstone project in which you will design a project from scratch. Depending on how ambitious you are, it might end up being much more work than Project 2.

Each project has a specific theme:

- Project 0 (2048): Introduction to Java
- Project 1 (Deques): Basic Design, Testing, and Code Review
- Project 2 (Ngordnet): Design, Large Scale Implementation
- Project 3 (BYOW): Large Scale Design

Projects will have different grader release schedule and restrictions on the number of allowed submissions per a time interval. Full grader details will be provided on the release of each project.

Projects 0, 1, 2, and 3 will be worth 50, 50, 125, and 125 points respectively.

Project 3

As Project 3 is a large-scale design project, it will have some differences from the other three projects.

Project 3 will have a synchronous checkoff portion.

For Project 3, all students will be required to work with a partner. You will either be able to choose a partner to work with or request to be paired up with another student who has similar working habits and goals as you.

One of the main goals of CS 61B is to give you the tools to become a successful software engineer, one of which is the ability to work effectively with others. **Barring extreme circumstances, we will not approve for students to officially work alone on Project 3.** If you think you have a compelling reason for working alone on Project 3, we will release a form that you can fill out closer to the release of the project (keep an eye on Ed for this) - we will let you know if your request is approved shortly before the release of Project 3.

Full details on partnerships can be found [here](#).

Exams

INFO

There will be two midterms and a final exam:

- Midterm 1: Thursday, February 15, 7–9pm PT
- Midterm 2: Thursday, March 21, 7–9pm PT
- Final exam: Tuesday, May 7, 8–11am PT

Midterm 1 grades will be unavailable before the drop deadline. If you're a prospective CS major and you are worried about dropping the course in the time before the drop deadline, please reach out to course staff for advice.

Remote Exams

We are **not** offering any remote exams this semester. You will need to be in-person on the listed dates to take the exams.

If there are any extenuating circumstances that prevent you from taking an exam or its alternate such as conflicting exams or sickness, please fill out this [form](#).

If you are unable to take the final exam, we may be able to let you take the exam the following semester - but no guarantees.

Alternate Exams

If you are unable to take the exam at the scheduled time, we will be offering only one alternate exam time per exam, in-person only.

- Alternate Midterm 1: Thursday, February 15, 8–10pm PT
- Alternate Midterm 2: Thursday, March 21, 5–7pm PT
- Alternate final exam: Tuesday, May 7, 11am–2pm PT

There are no other alternate exam times.

We are only offering the alternate exam time if you are unable to take the exam at the normal time. For example, if you have another exam at the same time, you can take the alternate-time exam. However, wanting a break between non-conflicting exams would not be a valid reason to take the alternate-time exam, because in this case you are able to take the exam at the normal time.

We'll release a form closer to the exams for you to sign up for an alternate-time exam.

Exam Grading

Our exams are intended to have an average score of 65%. If the exam is too difficult, and the class average score is below 65%, we will curve everyone's scores so that the average score is 65%.

If the average score is at or above 65%, then we will not change scores (i.e. your score will not decrease).

The maximum score on an exam is 100%. If your score exceeds 100% after curving, it will be capped at 100% when calculating grades.

You don't need to worry about the specific calculations for grade curving, but we've described the calculations below for transparency:

INFO

To compute curved scores, we will multiply everyone's scores by $(0.65 / \text{average score})$. Another mathematically equivalent way of stating this calculation is: we will decrease the maximum score on the exam until the average score is 65%.

Example: The class average is 60%, and your score is 70%. After curving, your score will be $70 * (0.65 / 0.60) = 75.83\%$.

Final Exam Clobbering

For those of you who miss an exam, have a bad night, or make major improvements over the semester, the exam clobbering policy gives you a chance to use your final exam score to make up some of the points on your midterm scores.

The clobber policy will only be applied if it helps your score. For example, if you score the median on both midterms, but then have a bad day and do terribly on the final, we will not change your midterm scores.

The clobber policy can only be used to make up points on the midterms. For example, if you score well on the midterms, that cannot be used to make up points on the final exam.

You don't need to worry about the specific calculations for clobbering, but we've described the calculations below for transparency:

INFO

If it helps your grade, we will replace your midterm exam percentile scores with the average of your midterm and final percentile scores. For example, if your score for Midterm 1 is in the 50th percentile, Midterm 2 is in the 30th percentile, and Final is in the 80th percentile, your clobbered Midterm 1 score will be in the 65th percentile, your clobbered Midterm 2 score will be in the 55th percentile, and your final exam score will remain at the 80th percentile.

Extra Credit

Extra credit is very limited in this course, because we would rather you achieve your grade by demonstrating your knowledge on the assignments and exams. That said, here are a few:

- Pre-semester survey (2.5 points)
- Mid-semester survey (2.5 points)
- End-of-semester survey (2.5 points)
- Official university course evals (up to 2.5 points)

We may add additional extra credit, but do not expect us to do so.

Pacing Points

There are three ways to earn "pacing" points, which are given for keeping up with the class.

- Attending discussion synchronously (in person or online)
- Attending lab synchronously (in person or online)
- Attending live/Zoom lecture, watching the pre-recorded videos, or watching the recorded lectures on time (before the start of the subsequent lecture).

Each discussion or lab is worth 0.5 pacing points. Each lecture is worth 0.5 pacing points. You may earn up to 16 pacing points. We will calculate pacing points using responses to the weekly surveys and formative assessments. See each the weekly announcement for details!

Pacing points are added to your score, but cannot increase your score beyond 750 points. In other words, if you have 740 points and 15 pacing points, your final score will be 750 points.

Grades

Your letter grade will be determined by the total points out of the possible 1000. In other words, there is *no curving in this course*, other than the clobbering policy above. Your grade will depend on how well you do relative to the bins, and not on how well everyone else does. Though we do not usually do this, we reserve the right to lower the the grading bins for 61B if grades are too low. We will not raise them.

Category	Percentage	Points
Homeworks	9%	90
Labs	5%	50
Surveys	1%	10
Projects	35%	350
Midterm 1	10%	100
Midterm 2	15%	150
Final Exam	25%	250
Total	100%	1000

A+	A	A-	B+	B	B-
984	914	854	794	754	714

C+	C	C-	D+	D	D-	F
679	644	609	569	534	499	0

These bins were designed to comply with departmental guidelines that the average GPA for a lower-division required course be in the range 2.8 - 3.3 (including students who drop or take the class for a P/NP grade). The design process involved setting of specific standards we expect students to achieve for the A, B, and C bins, with numbers adjusted and other bins interpolated based on a model that we built of predicted student performance based on previous semesters.

Since the class is not curved, and we provide all the grading details above, occasionally students will only be "a few points away" from the next grading bin after they receive their final total points value. Please do not contact us in this case. We cannot "round up" your grade or give you credit for work you did not complete. If you ever find yourself unable to complete an assignment, please refer to our friendly lateness policy (details below), or contact us before the deadline.

We will grant grades of Incomplete 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. Do not try to get an incomplete simply as a way to have more time to study or do a project. That is contrary to University policy. Before requesting an Incomplete grade, please contact a college advisor or review your college's Incomplete grade policies to understand if this is a right option for you.

Collaboration and Academic Misconduct

Deadlines can be stressful, and we know that under extreme pressure, it becomes tempting to start rationalizing actions that you would otherwise consider inappropriate. Perhaps you'll find yourself facing a CS 61B project deadline, and under all this stress you'll convince yourself that you're just going to cheat for the moment so you can get the points, and that you'll come back later and really learn the thing you were supposed to have learned in order to restore your karmic balance (I've heard something along these lines a few times).

Please don't do this. We care that you learn the material, and that your grade reflects your learning, regardless of the timeline. If you feel that you need to engage in academic misconduct to meet a deadline, please reach out to course staff.

Academic misconduct on any homework, lab or project will result in requiring you to write an anonymous reflection to be posted on this course website and to resubmit the 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 shared code with others. Given our friendly lateness policy (see below), there will be no exceptions to this rule.

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.

Homework and Lab Collaboration

The entire point of homework and labs is to learn. For homework or labs, you should feel free to collaborate with others however you choose, though keep in mind that greater independence is likely to give you a better learning experience (as long as you aren't totally stuck). **Even though we will allow close collaborations on HW and labs, the solutions you submit should still be your own work!** Identical or near identical submissions will be treated as plagiarism.

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 on your own (or with your partner on project 3, which allow partners), particularly when it comes to writing the actual code.

Exam Misconduct

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. As above, all incidents of cheating will be referred to the Office of Student Conduct.

Academic Misconduct

What constitutes cheating? **The golden rule of academic dishonesty is that you should not claim to be responsible for work that is not yours.**

This is obviously open to some interpretation, and you'll be getting some help from instructors, the internet, other students, and more throughout the course. This is OK, and we hope that the class is an open, welcoming, collaborative environment where we can help each other build the highest possible understanding of the course material.

To help (but not entirely define) the bounds of acceptable behavior, we have three important rules for coding assignments:

- 1 **By You Alone:** All 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).
- 2 **Do Not Possess or Share Code:** Before you've submitted your final work, you should never be in possession of solution code that you (or your partner) did not write. Looking up solution code online is effectively possessing solution code.

You will be equally culpable if you distribute such code to other student or future students of 61B (within reason). **DO NOT GIVE ANYONE YOUR CODE – EVEN IF THEY ARE DESPERATELY ASKING. DO NOT POST SOLUTIONS ONLINE (on GitHub or anywhere else)!** If you're not sure what you're doing is OK, please ask.

- 3 **Cite Your Sources:** When you receive significant assistance on a coding assignment from someone else, you should cite that assistance somewhere in your source code with the `@source` tag in a comment. We leave it to you to decide what constitutes 'significant'. However, citing does not change the extent of collaboration or reference allowed.

WARNING

Examples:

```
// @source I worked with [person], the code below receives a list of objects
// and iterates through the list until it finds the number 3 and
// returns the index.
```

```
...
```

```
// @source From https://stackoverflow.com/questions/16242733/sum-all-the-elemer
// the following code iterates through the list and sums its values.
```

```
...
```

```
// @source This code was generated by ChatGPT. It reads and parses
// all integers from the file, which I then pass into my computeSum method.
```

```
...
```

For clarity, some examples of specific activities are listed below:

Permitted:

- High-level 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 other's 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 (e.g. googling "uppercase string java" may lead you to some sample code that you copy and paste into your solution). Such usages should be cited as comments in your HW, lab, and especially project code!

Permitted with **Extreme Caution**:

- Working on a project alongside another person or group of people. Your code should not substantially resemble anyone else's!
- Use of GitHub Copilot / GPT3 / etc. If you're just generating some amount of boilerplate code, that's ok. However, you should not use such tools to generate non-trivial methods. We are trying to build your fundamental skills, and leaning on an AI is going to cause you trouble in circumstances where you don't have an AI to help, such as exams. **Any AI generated code must be cited and explicitly commented as such.** Violation of this citation rule is a serious act of plagiarism. Note that AI generated code, especially from sources like ChatGPT, is often subtly buggy or completely incorrect.

Absolutely Forbidden:

- Typing or dictating code into someone else's computer is a violation of the "By You Alone" rule.
- Possessing another student's 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.
- Possessing solution code that you did not write yourself (from online (e.g. GitHub), staff solution code found somewhere on a server it should not have been, etc.) before a final deadline. Distributing such code is equally forbidden. **Looking at solution code online counts as possessing code.**
- Posting solution code to any assignment in a public place (e.g. a public git repository, Google Drive, Discord, 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.

We have advanced cheating detection software, and we will routinely run this code to detect cheating. Every semester, we catch and penalize a significant number of people (roughly 100 cases per semester). Do not be one of them. If you find yourself at such a point of total desperation that cheating begins to look attractive, contact one of the instructors. Likewise, if 61B is causing massive disruption to your personal life, please email `cs61b (at) berkeley.edu`.

In essence, our ultimate goal is for students to grasp the code they've crafted. It's not just about completing projects; it's about cultivating a deep comprehension of their code. We assign these projects to give you an opportunity to build a strong foundation in coding, ensuring that you truly understand the core principles behind the concepts we teach in this course.

If you admit guilt to an act of plagiarism before we catch you, you will receive the same penalty (redoing the assignment), but we will not refer your case to the Center of Student Conduct. However, we would strongly prefer that you ask us for additional time before opting to plagiarize. You do not need to admit guilt to ask for additional time.

Obviously, the expressive power of Java is a subset of the English language. And yes, you can obviously 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.

If you would like to review our academic misconduct resolution procedures, you can find them [here](#).

Lateness

The expected deadlines for each assignment are posted on the class website. Our assumption is that you will complete each assignment by its due date. Especially in the first few weeks, the course content builds on itself, and not completing an assignment will make it much more difficult to complete later assignments and do well on exams.

There is no partial credit for work that is submitted late. And thus, Gradescope will give you zero points if you submit after the deadline.

However, we understand that you may be busy, tired, or experiencing a crisis. We would like your grade to reflect what you demonstrate proficiency in and accomplish in the class, and have designed some flexibility to our lateness policy to achieve that as much as possible. Your health, safety, and well-being are the priority, and ideally your grades should reflect what you have learned, not when you learned it.

Extensions

By default, all assignments will follow this extension policy:

- Short extension: Extension requests of less than 24 hours will be instantly processed by an automatic system,
- Long extension: Extension requests between 24-72 hours will be manually processed by GSIs and will receive a reply within 1 business day.

Extensions requests can be made in Beacon ([guide](#)) and will require you to provide detailed reasons for your extension. Extensions can be requested in Beacon. If you have [extenuating circumstances](#), you may request a longer extension. Extensions are not guaranteed.

WARNING

Exceptions:

- 1 For **Project 2 only**, a short extension will be up to 72 hours and a long extension will be 72-120 hours. Like before, short extensions are automatically granted while long extensions require manual approval.
- 2 Surveys may **not** be extended.
- 3 Conceptual homeworks (HW3 and HW4) may **not** be extended, as we will release solutions as exam prep.
- 4 Project 3B/C may **not** be extended, as it is difficult to arrange late project presentations.

INFO

An important note on extensions: leading up to an assignment deadline, we usually provide increased support for that assignment (in the form of increased staffing). As a result, continuing to work on an assignment after the original deadline may lead to slower response times on Ed/longer wait times in OH, as staff will actively prioritize assignments whose deadlines have not passed.

Extenuating Circumstances

We define extenuating circumstances as circumstances outside of the student's control that directly inhibits the student's ability to complete assignments in a timely manner. Some examples include unforeseen physical/mental health crises, technical issues (broken laptop), family emergency, etc.

If you are experiencing such circumstances and require an extension past what can be requested in Beacon, please fill out [this form](#). There is a separate Google form and a more involved process. We will be restrictive about granting these extensions.

DANGER

We will **not** be considering retroactive submissions to the extenuating circumstances form. This means that you should proactively submit to the form **before** the assignment's deadline to be considered. Exceptions will only be made in extreme cases.

Additionally, if you'd like to speak to a student support TA, book an appointment with us [here](#).

Late Adding CS61B

If you are attempting to officially enroll in CS 61B late in the semester, you must turn in all outstanding course assignments within 3 days of joining the course. This means that you should be following along with the course schedule according to the schedule on the main page and announcements on Ed. Please email cs61b@berkeley.edu if you are not added to Ed or Gradescope.

Please use the [Late Add Intent form](#) if you are waiting to be enrolled and need access to course platforms. If you'd like to meet with course staff for advice on catching up with the course, please make an appointment [here](#).

Auditing CS61B

This is for students who are unofficially auditing this class. Maybe you are a non-Berkeley student who wants to brush up on your programming knowledge or even a high school student who wants to get a head start on your programming career. If this sounds interesting to you, we have set up a public autograder for the Spring 2021 version of the course.

To get everything set up, go to [Gradescope](#) and select the "Add a course" button. Enter course code **MB7ZPY** to be added. You will not be given a student repo, so you will have to create your own on GitHub in order to submit assignments. After you complete setup and lab1, you will also have access to all the code for the class.

That's all you have to do! All lectures, worksheets, and specs will be available on the [Spring 2021 Website](#). **Important:** you should be using the Spring 2021 skeleton, website, and gradescope. You

will not be using any resources from this current semester. Every term we change assignments and lectures up a little bit, so make sure to stick to the Spring 2021 version of things.

Acknowledgements

Some course handout material derived from previous terms, professors, and GSIs.