

CS 61B

[Resources](#) / [Guides](#) / Ed Policies and Guidelines

Ed Policies and Guidelines

Author: Ethan Ordentlich, Jedidiah Tsang

Introduction

We use Ed Discussions as our course forum. If you're new to Ed, please read the [Ed guide](#), which should get you familiar with this platform. This post includes a collection of guidelines for etiquette on Ed, that we ask you to read entirely so we can all have an enjoyable experience this semester!

Keeping Ed Inclusive

Any content on public forums like Ed that targets, excludes, disrespects or discriminates against a certain individual or community will be removed and the person responsible for the content will face repercussions. Please aim to keep the climate inclusive for everyone. If at any point you have felt targeted, excluded, disrespected, or discriminated against by a student or someone from our course staff, please fill out our [Incident Report form](#) to report it. If you would rather not speak to someone from staff, there are several campus offices you may report to instead (<https://diversity.berkeley.edu/report-incident>).

Overall Requests

We don't want to start this with a list of what not to do, but please **don't**:

Post unhelpful follow-ups

Follow-ups like "+1" and "me too" contribute nothing but cause annoying emails to be sent to everyone following the post. These may be deleted. You can use the heart icon near a post to express "+1", gratitude, etc.

Use annoying or misrepresentative titles

These are titles in all caps or with an exorbitant amount of punctuation just to capture the attention of users. Please keep titles (and content in general) descriptive and professional.

Post off-topic or speculative content or trolling

Content that is unrelated to the course and/or non-educational does not belong on Ed. Content that is speculative, drama mongering, or intentionally misleading prevents effective communication from course staff and other students.

Say “nevermind”, or “fixed” if you figure out the answer to your own question

The forum is a wonderful source of collective knowledge. Answering your own question is valid, and contributes to this collective knowledge. Everyone hates when somebody replies to their own question with something like “[Nevermind, I got it!](#)”, or [when they don’t answer at all](#). Other students can’t learn from that if they happen to have the same problem. Instead, add an answer or comment with your solution!

Why Use Forums?

[Recent research from Duke/NC State/UNC/UF](#) has shown that student participation on course forums, like Piazza and Ed, is most beneficial when students ask **constructive questions** that reflect their reasoning or attempts to solve a problem. By creating and asking these constructive questions, students are practicing their *metacognitive* skills, or how they *think about approaching problems and learning*.

Forums are also an effective way of seeking asynchronous help. Office hours and lab aren’t always happening, and it isn’t always possible to speak synchronously with a TA. Instead, you can ask a question on Ed, and continue working on other things while you wait for a response.

Additionally, forums and other asynchronous methods of asking for help are a widely used tool outside in many real-world contexts as well, from [StackOverflow](#) to internal company tickets.

CS 61B Forum Policies

Many of the benefits of forums only apply if we all use it appropriately and effectively, and put in the work to maintain its quality.

Categories

We have many students in CS 61B, and will almost certainly have many posts. To organize these posts, we use *Ed categories*. To make a post or ask a question on Ed, you **must** select an appropriate category.

An overview of the categories is as follows:

Logistics

General questions about what’s happening in the course.

Lectures

Questions about each lecture's content. This category has a template asking for the lecture numbers.

Projects, Labs, Homeworks

Questions and requests for debugging help on the assignments. These categories have subcategories for each assignment in. Each subcategory also has subsubcategories, for distinguishing between private debugging questions (Gitbugs) and various kinds of public questions. For example, Projects – Project 1 – Gitbugs, and Projects – Project 1 – Conceptual (Circular Buffer).

Discussions

Questions and conceptual clarifications on discussion worksheets. This category has a subcategory for each week.

Exams

Questions about exam logistics, past exams, and anything related to exams. As we get closer to the exam, we will add additional categories for asking questions about previous exams.

Miscellaneous

Any questions about the course that doesn't fit into the above categories can go in here.

Social

Looking to form a study group or hold a study session? Post here.

External Announcements

Any postings from external groups not affiliated with CS 61B will be posted in this category. Posts in this category must be approved by course staff.

How to Ask, and How to Not Ask Questions

Asking for help on Ed is unfortunately slower, as we can't spend all our time on the forum. Before you ask for help, you should try to answer your question yourself! We also want you to learn how to help yourself, which in some cases can be slower than asking staff, but will be more beneficial in the long run. Some resources that can help you do that are:

Previous questions on Ed.

You can search within a category with Ed's search, so that you don't have to look through everything.

Official course materials

The assignment spec is a great source of information. Additionally, each assignment will have an **FAQ** linked from the top of the page. The FAQ will be updated while the assignment runs, so check back often! Course staff will also create several supplemental materials for assignments while students work on them, as we find where students are struggling.

Google or a search engine of your choice.

While searching Google might not seem like “independent programming”, professional programmers often search the web for solutions to problems similar to the one they are trying to solve. Learning good search techniques now will pay off later in the semester and into your careers. **An exception to this is searching for solution code. Don’t do that.**

However, you shouldn’t spend hours on the same problem, and should ask for help, possibly on the forum. When you do ask for help on Ed, here are some things that you can do to make sure that the response you receive is more likely to resolve your question.

Explain what your problem is, and what you think will help you proceed.

Course staff are not mind-readers. We won’t magically know what’s wrong, or what you want answered.

If your question doesn’t contain enough information for us to be able to answer, or doesn’t show us what you’ve tried so far, we will ask. You’ll get a slower answer overall, but you’ll also learn how you can improve your future questions to get a quicker response.

Similarly, there are some things that you should **not** do to get a response:

Asking for “a hint” or “the answer”

We don’t know what kind of hint you want or what you are trying to accomplish. We don’t even know if you’ve worked on the problem, or if you saw the error and immediately asked.

We will not give the answer, and will instead ask a follow-up question to clarify what you are asking for.

Saying that “I don’t understand anything”

It’s very hard, often impossible, to give an answer to this. Instead, you should try to figure out a starting point for what you don’t understand. If you can’t get started, that’s still a valid question to ask – and is very different from “anything”.

Some material here drawn from Adam Blank’s guide on [“How to Ask For Help”](#).

Conceptual Posts

Often, you'll have a question about a conceptual idea from lab or a project, or a problem on a discussion, past exam, or outside resource. Here are some guidelines that you should follow to get a helpful answer more quickly:

Don't ask if your solution "is correct"

Instead, explain why you think your solution is correct, and point out any pieces that you're not sure about. Reconstructing your argument is a lot of work for staff to try to go through, so explaining your train of thought to us helps you get a better answer and address any misconceptions you may have.

Private Debugging Help, or Gitbugs

Sometimes, you may run into a problem where you need to show course staff your code. However, per our [collaboration policy](#), you may not post code publicly, including on public Ed posts. Instead, you can use the Gitbugs category for each assignment to make a private debugging post.

Here are some guidelines that apply to asking private debugging questions specifically:

Link your GitHub repo and push.

Sometimes (but not always), we will look at your code. This ensures that we can easily look at the most up-to-date version.

Provide a screenshot of the section of your code you have a question about.

We will require that you provide a screenshot, so that we know what section of code you're looking at.

When describing your bug, be specific.

Descriptions like "everything", "it isn't working", "it fails the autograder", "I don't know", etc. are not specific. We cannot help without knowing what's wrong. At a *minimum*, you should tell us what test is failing, or what error you're seeing, though this alone doesn't guarantee that the first response will be helpful.

If you can identify a smaller piece of code, such as a single method or a few lines, that are not behaving as you expect, that's wonderful – tell us what you've found, and what you're confused about!

We expect that you've tried to debug. Tell us what you've learned.

This goes hand-in-hand with the above guideline! If we can't tell that you've tried to debug, we will not debug for you, and instead ask what you've tried. This can be print debugging, or using the IntelliJ debugger. Screenshots can be very helpful here, as long as you explain what we're looking at!

Ed Tips and Tricks

Change your email settings to receive fewer emails.

In the upper right corner, click on your avatar, then "Settings", and then hit "Notifications" to adjust the number of emails you receive from Ed. Announcement emails will always reach you.

Follow posts you want to know more about.

If you see an ongoing question you'd like to keep track of, you can either:

- Click the "Watch" option at the top right of the question to receive email updates, or
- "Star" the post at the top right of the question to keep it pinned on your Ed.

Refer to other questions, answers, and comments

Using #N (for example, #1 or #1c) is a clean way to refer to other threads. You can use this to provide an answer to a repeated question, or to point to things that you've tried but haven't worked.

Make code (and stack traces) readable.

Use the runnable code button in the editor toolbar so that your code looks nice. Syntax-highlighted code is much easier to read than code as text.

If you have a question about a personal matter that you don't want other students to see, please make a private post. Private posts are only seen by the course staff. If you want only the head TAs and Josh to see your question, please email `cs61b [at] berkeley.edu`

TABLE OF CONTENTS