

Last edited: Jedi Tsang (December 19, 2025)

## Introduction

If you're reading this, that means you're likely interested in using EdStem's LLM integration, Bots++! Bots++ allows staff to leverage LLMs to automatically generate draft answers to questions.

## Setup & Help

To get Bots++ set up, email the Director of the Center for Teaching, Learning and Outreach, Jasmine Bryant: [jrbryant@caltech.edu](mailto:jrbryant@caltech.edu). For technical assistance related to Bots++, you can reach out to Ed support at [support@edstem.org](mailto:support@edstem.org).

## Configuring Bots++

### Choosing a LLM provider

Once you're added to the course, click  (found on the top right), and you'll see Bots++ on the left hand toolbar. From there, you can see a list of LLM providers.

At Caltech, we've set up a local LLM called "YertleBot" that you can use for your course. At this time, other LLM providers have not been authorized for use at Caltech.

### Uploading Course Resources

You can also set up additional context that EdStem vectorizes automatically for you without any pre-parsing done on your end. TIP: Naming your pdfs appropriately seems to really help the bot out.

You can also embed entire previous course iterations by clicking on the ellipsis (...) to the right of Upload File, Import Embeddings, and select the course you want. For a course with ~800 students enrolled in a semester-long class, this took approximately 1-2 hours.

The screenshot shows the Bots++ interface. On the left is a sidebar with 'Bots++', 'Discussion', and 'Bot' options. The main area is titled 'Bots' and shows a bot icon with 'Bot ENABLED' and 'Edit Disable' links. A '+ New Bot ...' button is in the top right. Below is a 'Resources' section with the text 'Upload course resources for the bot to use when responding. Try including lecture slides, course outlines, problem sets, etc.' and buttons for 'Refresh', 'Requeue', 'Delete', and 'Upload File ...'. A large dashed box contains the text 'Drag files here to upload'.

## Disclaimer text

You can provide custom disclaimer text to be added before the bot's output. This can be used to remind readers that LLMs can make mistakes.

## Help deciding between Ed Discussion and Piazza #4

Jasmine Bryant **STAFF**  
29 minutes ago in **General**

PIN STAR WATCH VIEWS 4

What should I look for when trying to decide between two similar educational technology tools?

Comment Edit Delete Endorse ...

Sort by Newest ▾

Add comment

Bot **BOT** 28m  
*This bot may give you incorrect information. Please always verify and evaluate the output.*

Before I can give you the most useful guidance, could you tell me a bit more about your situation? For example:

## Responses

Bots++ can be configured to automatically respond to:

NOBODY

This is useful when you only want to trigger Bots++ manually.

## [Recommended] STAFF ONLY COMMENT

This is useful where draft responses can be researched and prepared automatically ready for the staff to review, post, endorse. Especially if this is your first time using Bots++, we recommend using this feature

## STUDENTS

This is useful when you want to provide automatic responses to students.

## STUDENTS AND STAFF

This is useful when you want to provide automatic responses to students and staff.

Bots++ can also be configured to respond with:

## ANSWER

This is useful when you want to provide automatic answers to questions.

## COMMENT

This is useful when you want to provide comments on posts

## Model Context

You're also able to provide some written context to the LLM.

respond to direct replies to the bots comments

### About the course

Please write down what the course is about and any relevant learning objectives. e.g. "This course is about ...". This helps the bot better make judgements related to your course material. This text will be integrated into the overall prompt.

CS 61C at UC Berkeley is a lower-division college course on computer architecture. The subjects covered in this course include C and assembly language programming (RISC-V), translation of high-level programs into machine language, computer organization, caches, performance measurement, parallelism, CPU design, warehouse-scale computing, and related topics.

### Core prompt

This is the prompt used to generate the bot's responses, leave it blank to use our default prompt.

The default prompt is: "You are a helpful assistant to the instructors and students of the course. You only answer questions related to the course or course material, use your best judgement based on the available context and the course code and name. If the question is not clear, please ask the asker to elaborate. If you are not sure, you can always refer the students to the course staff for help. "

You are a helpful assistant to the instructors and students of the course. You only answer questions related to the course or course material, use your best judgement based on the available context and the course code and name. If the question is not clear, please ask the asker to elaborate. If you are even remotely not sure, you can always refer the students to the course staff for help.

Here is some sample text you could use for inspiration:

About the course: CS 61C at UC Berkeley is a lower-division college course on computer architecture. The subjects covered in this course include C and assembly language programming

(RISC-V), translation of high-level programs into machine language, computer organization, caches, performance measurement, parallelism, CPU design, warehouse-scale computing, and related topics.

Core prompt: You are a helpful assistant to the instructors and students in the course. Before you answer a question, ask a clarifying question in order to understand what the user is asking about. Try to explain concepts using analogies. When asked to solve a problem, do not solve numerical problems, but ask the user what strategies they have tried already.

## Using the Bot

After asking a question on EdStem, click on the ellipses next to Endorse.

### Testing with the fa21 mt1 embedding #4



Jedidiah Tsang **STAFF**  
Last week in **General**



PIN



STAR



WATCHING

35

VIEWS



FA21-Midterm-Q2.1: Why is \*str1 located in static memory? Can you explain why the other choices are incorrect?

Comment Edit Delete Endorse ...

Sort by Relevant ▾

You should be able to see “Trigger LLM Bot”. After some delay, the response by the bot should show up! If you’re interested to see the context that was provided to the model, you can click “View LLM Context.”

## Testing with the fa21 mt1 embedding #4



Jedidiah Tsang **STAFF**  
Last week in **General**



PIN



STAR



WATCHING

35

VIEWS



FA21-Midterm-Q2.1: Why is `*str1` located in static memory? Can you explain why the other choices are incorrect?

Comment Edit Delete Endorse ...

Sort by Relevant ▾



Bot **BOT** 1w Visible to staff only

The reason `*str1` is located in static memory stems from the nature of how C handles string literals. When you assign a pointer to a hard-coded string like "Hello World", the string itself is stored in a segment of memory reserved for constant values which don't change at runtime. This segment is typically referred to as static (or sometimes text) segment. The pointer `str1` points to this location, making `*str1` located in static memory. Here's why the other choices are incorrect:

- **Code:** The code segment is primarily reserved for the actual machine instructions of the program, not for data storage.
- **Heap:** The heap is used for dynamic memory allocation. Variables only get stored on the heap if they are explicitly allocated memory at runtime, typically using functions like `malloc`. Since `*str1` is not dynamically allocated, it doesn't reside on the heap.
- **Stack:** The stack is used for static-duration storage that is allocated and deallocated as functions are called and return. Since string literals exist for the duration of the program and are not local to a specific function scope, they are not placed on the stack.

In summary, `*str1` is in static memory because it is a compile-time constant whose lifetime spans the entire execution of the program, making it unsuitable for the stack, heap, or code segments.

2 Reply Edit Delete ...

You can regenerate a response to the question by clicking on the ellipsis next to the Delete button on the LLM output and clicking "Regenerate."