

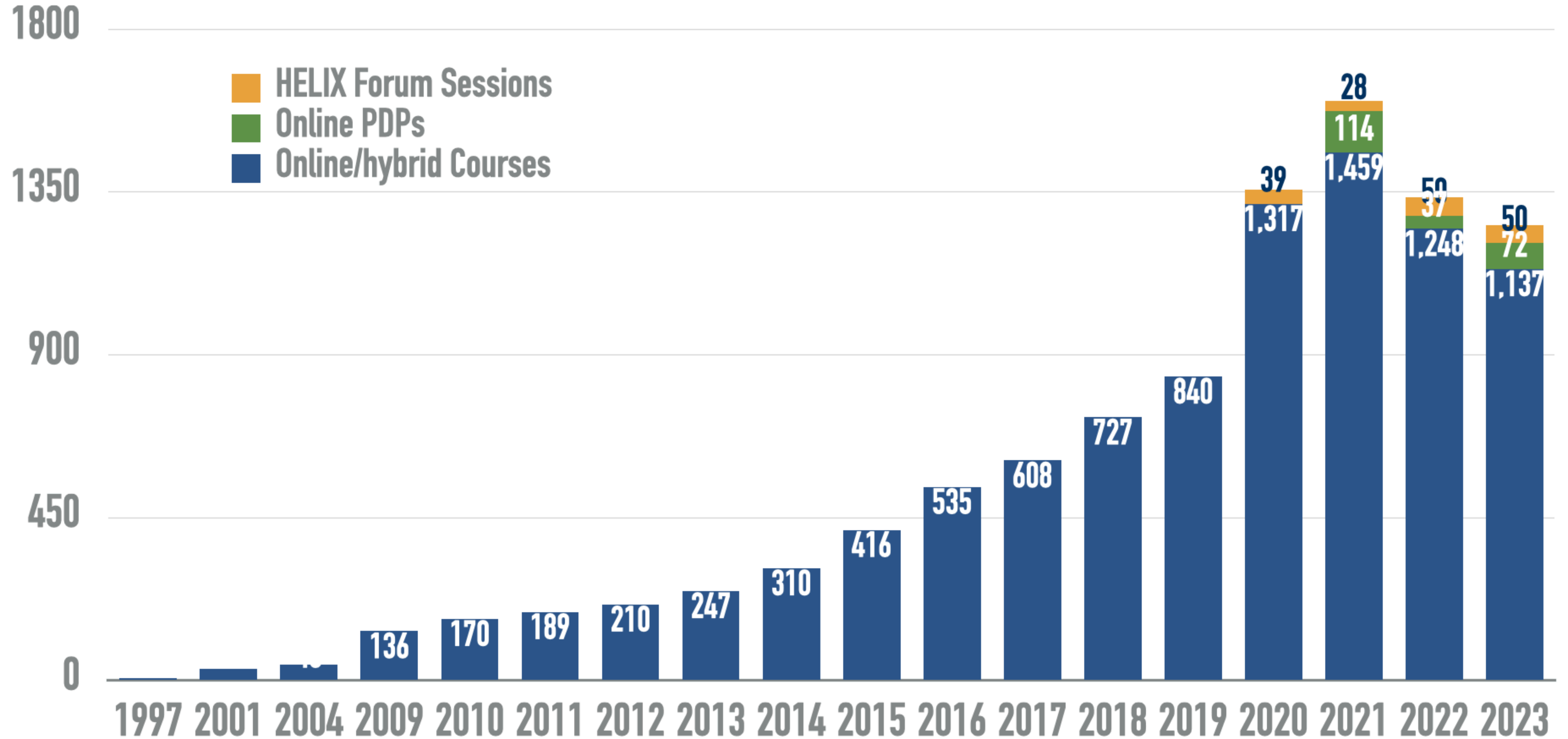


Harvard  
Extension School  
HARVARD DIVISION OF  
CONTINUING EDUCATION

# Instructor Development Night



# DCE ONLINE OFFERINGS PER ACADEMIC YEAR



# Types of Students

Local

Geographically distant

High-schoolers

Retired individuals

Undergrad students

Grad students

Part-time workers

Full-time employees

Non-native English  
speakers

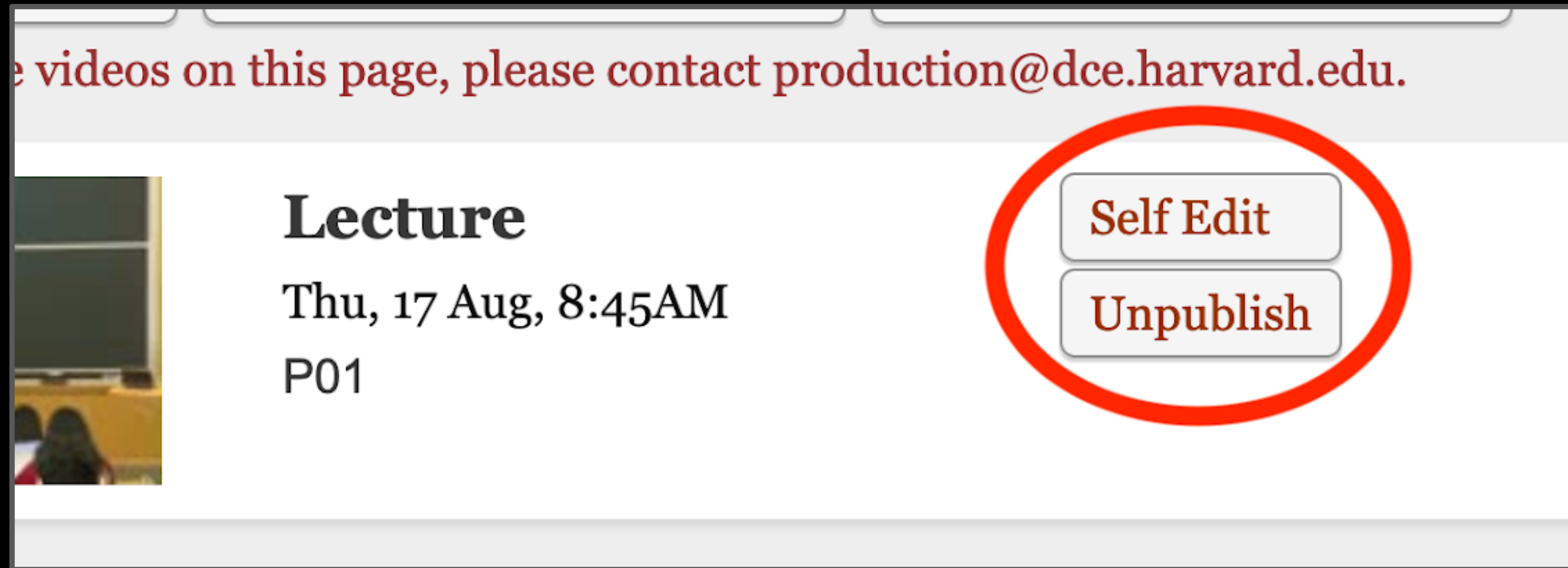
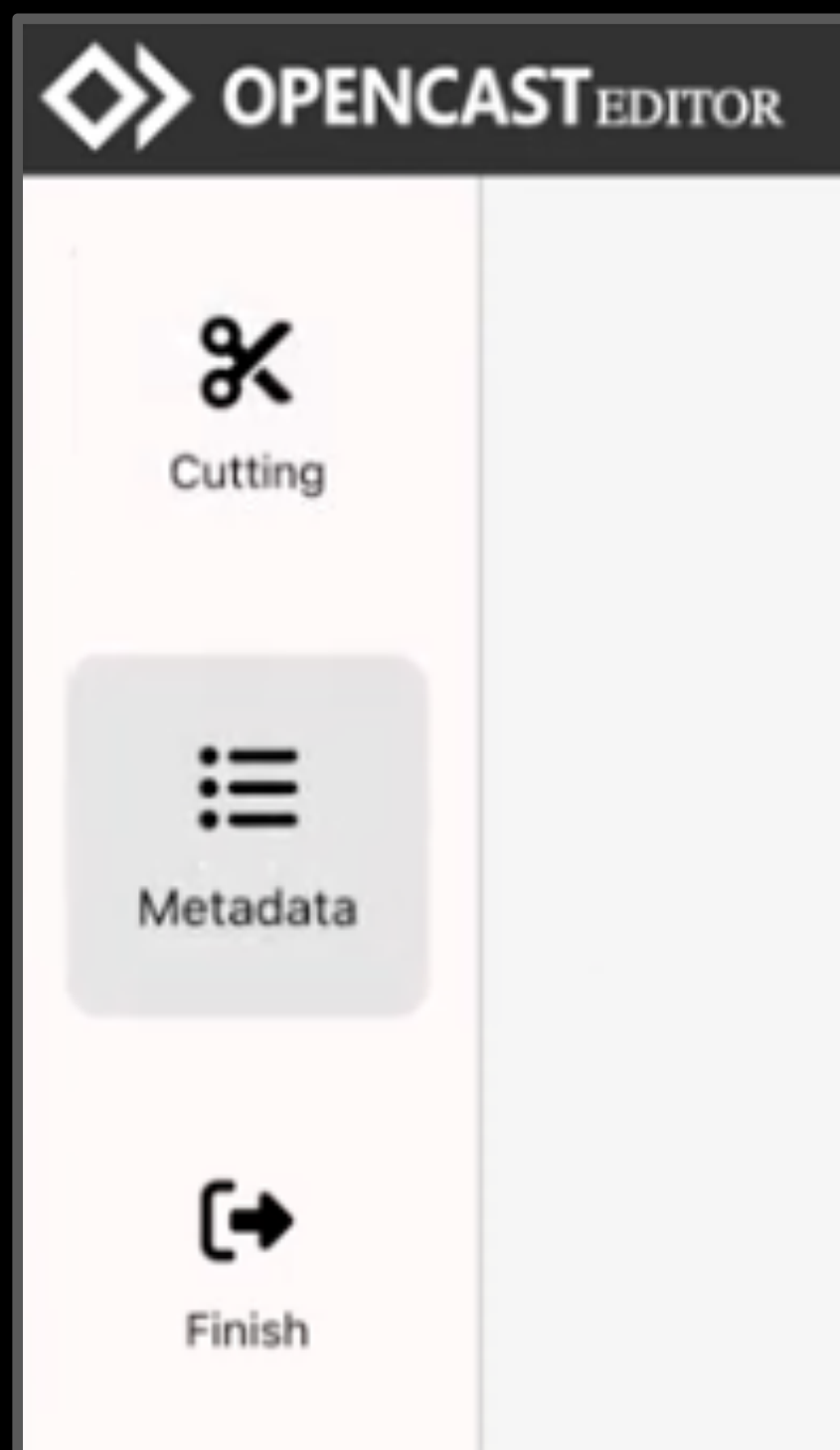


# Technological/Pedagogical Innovations

- Unpublish and self-edit lecture videos
- Gather *Study Lounge*
- Portable *HELIX*
- Immersive Classroom
- AI in teaching computer science
- Augmented and virtual reality

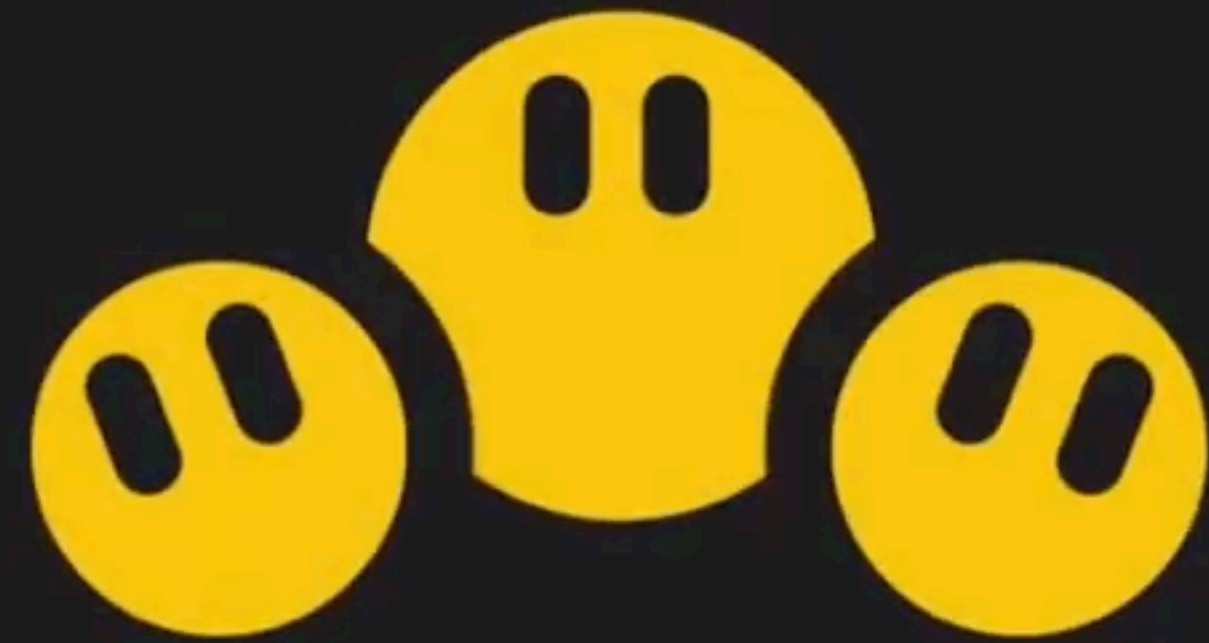
# OpenCast Self-Edit and Publish

- Unpublish videos to take them down immediately
- Self Edit to trim out start, end, and breaks



- Self edit **Metadata** to change the title and description on the listing
- OpenCast users: watch your email for more information
- Demo video: <https://tinyurl.com/OCselfedit>

# Gather Study Lounge



# Gather

# Gather Study Lounge

A space where students get together either informally to socialize or network, or to collaborate on coursework. E.g.,

- Watch class recordings together — particularly useful for asynchronous students
- Work on projects together
- Hold study sessions
- Get feedback from each other on their work
- Continue discussions started during class



# Portable HELIX, version 2

Home Insert Design Layout References Mailings Review View Share

You are viewing Alyssa Weber's screen View Options

Recording... View

Participants (2)

Search

ST Support Technician (Co-host)

CSCI 7 (Host)

CSCI 7 (student view)

Alyssa Weber (Co-host)

A Aaron (Guest)

AA Aaryan A (Guest)

A Anshu (Guest)



AN Anthony Narag (he/him) (Guest)

BC Brian Chen (Guest)

Y Chloe Huang (Guest)

## Overview of Resources

1. Short course intro – what you will learn, how fast the course moves, etc
2. Ed discussion board – what is this for?
3. Ed pinned post of Section & OH schedule
4. Really, really read the syllabus - lot of info there!



# The Immersive Classroom



Law 570 - Week 3



Watch later Share

Chat

= Written after the event

**Live Chat**

Hide

Who's online:

**Gabe**  
at 0:00

Leave and Mute Chat

Gabe 10:24 am

Hey, I'm curious what you think about that group project that the prof just introduced.

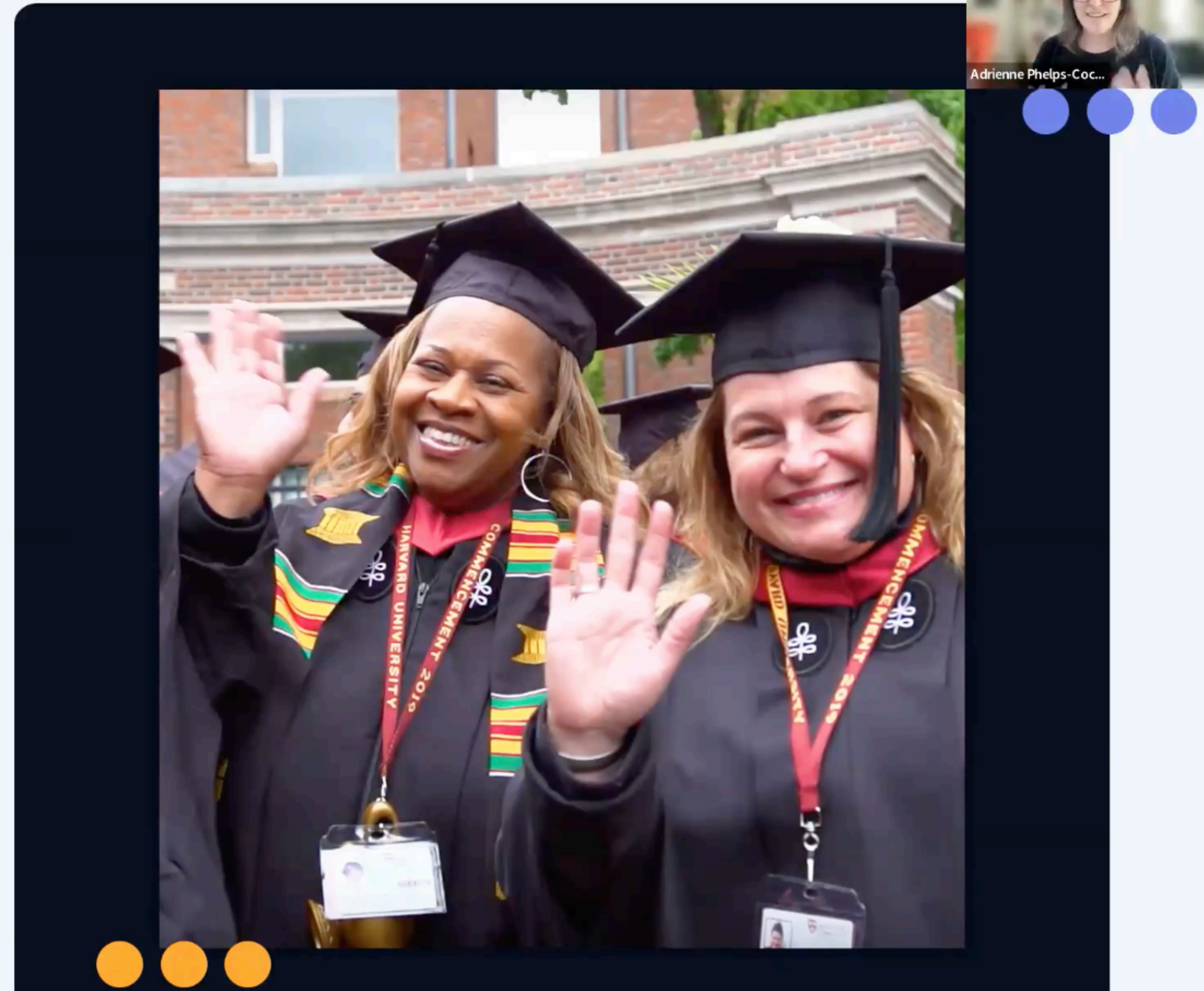
Kina 10:25 am

Yeah it seems really interesting, but I'm confused about what kind of presentation we're supposed to create

**Write message here:**

Write message here

# Connection



01:18 / 14:56



**Video Comments**

Show

# AI in Teaching Computer Science: *Work in Progress*

1. Explain highlighted lines of code.
2. Advise students on how to improve their code's style.
3. Answer (most of the) questions asked online by students.
4. Help students find bugs in their code, via TA-like rhetorical questions.
5. Explain arcane error messages, hint how to solve.
6. Assess design of programs, provide qualitative feedback.
7. Enable other faculty to tailor CS50's tools for their own (non-CS) courses.
8. Assess students' understanding via conversations with AI (a la oral exams).
9. ...

# AI in Teaching Computer Science: *Work in Progress*

1. Explain highlighted lines of code.
2. Advise students on how to improve their code's style.
3. Answer (most of the) questions asked online by students.
4. Help students find bugs in their code, via TA-like rhetorical questions.
5. Explain arcane error messages, hint how to solve.
6. Assess design of programs, provide qualitative feedback.
7. Enable other faculty to tailor CS50's tools for their own (non-CS) courses.
8. Assess students' understanding via conversations with AI (a la oral exams)
9. ...



Show All Commands ⌘ ⌘ P

Go to File ⌘ P

Find in Files ⌘ ⌘ F

Start Debugging F5

Toggle Terminal ⌘ `

```
1 #include <cs50.h>
2 #include <stdio.h>
3
4 int main(void)
5 {
6     string name = get_string("What is your name? ");
7     printf("hello, %s\n", name);
8 }
9 |
```

```
1 #include <cs50.h>
2 #include <stdio.h>
3
4 int main(void)
5 {
6     string name = get_string("What is your name? ");
7     printf("hello, %s\n", name);
8 }
9 |
```


hello.c — hello [Codespaces] — x +

code.cs50.io

hello.c x

```
1 #include <cs50.h>
2 #include <stdio.h>
3
4 int main(void)
5 {
6     string name = get_string("Name: ");
7     printf("hello, %s\n", name);
8 }
9
```

- Go to Definition ⌘F12
- Go to Declaration
- Go to Type Definition
- Go to References ⇧F12
- Peek >
- Find All References ⇧⌘F12
- Explain Highlighted Code ⇧⌘E**
- Rename Symbol F2
- Change All Occurrences ⌘F2
- Format Document ⇧⌘F
- Format Document With...
- Format Selection ⌘K ⌘F
- Refactor... ^⇧R
- Cut
- Copy
- Paste
- Switch Header/Source ⌘O
- Go to Symbol in Editor... ⇧⌘O
- Go to Symbol in Workspace... ⌘T





The image shows a code editor with two panes. The left pane shows the source code for a C program named 'hello.c'. The code is as follows:

```
1 #include <cs50.h>
2 #include <stdio.h>
3
4 int main(void)
5 {
6     string name = get_string("What is your name? ");
7     printf("hello, %s\n", name);
8 }
9
```

The right pane shows the same code snippet but with a dark background, likely representing a dark theme or a specific analysis view. Below the code in the right pane, there is a text explanation of the code snippet:

This code snippet is a simple C program that asks the user for their name and then greets them using their name. Let's break down the code step by step:

1. `#include <cs50.h>`: This line includes the "cs50.h" header file, which is a library developed by Harvard University for their CS50 course. This library provides some useful functions for input and output, such as `get_string()` which we will use later in the code.
2. `#include <stdio.h>`: This line includes the "stdio.h" header file, which is a standard C library for input and output operations. It provides the `printf()` function that we will use to print the greeting message.
3. `int main(void)`: This line defines the main function of the program. In C, the main function is the entry point of the program, where the execution starts. The `int` keyword indicates that the main function returns an integer value, and the `void` keyword inside the parentheses means that the main function takes no arguments.
4. Inside the main function, we have the following lines of code:

# AI in Teaching Computer Science: Work in Progress

1. Explain highlighted lines of code.
2. Advise students on how to improve their code's style.
3. Answer (most of the) questions asked online by students.
4. Help students find bugs in their code, via TA-like rhetorical questions.
5. Explain arcane error messages, hint how to solve.
6. Assess design of programs, provide qualitative feedback.
7. Enable other faculty to tailor CS50's tools for their own (non-CS) courses.
8. Assess students' understanding via conversations with AI (a la oral exams)
9. ...

cat.c — 10591665 [Codespace] x +

cs50.dev

cat.c x style50

```
1 #include<stdio.h>
2
3 int main(void)
4 {
5     for(int i=0; i<3; i++)
6     {
7         printf("meow\n");
8     }
9 }
```

CS50

...

Profile icon

Settings icon

cat.c — 10591665 [Codespace] +

cs50.dev

cat.c style50 cat.c × Explain Changes Apply Changes

```
1- #include<stdio.h>
2
3 int main(void)
4 {
5- for(int i=0; i<3; i++)
6- {
7- printf("meow\n");
8- }
9 }
```

→

```
1+ #include <stdio.h>
2
3 int main(void)
4 {
5+ for (int i = 0; i < 3; i++)
6+ {
7+ printf("meow\n");
8+ }
9 }
```

cs50

cat.c — 10591665 [Codespace] x +

cs50.dev

CS50 DUCK DEBUGGER

you

Explain Changes

ddb

The changes are made at line 1 and lines 5 to 8 in the file.

At line 1, the change is in the inclusion of the `<stdio.h>` library. It was originally written as `#include<stdio.h>`, but it has been modified to `#include <stdio.h>`.

At lines 5 to 8, the changes are in the formatting of a for loop. It was originally written as:

```
for(int i=0; i<3; i++)
{
    printf("meow\n");
}
```

But it has been modified to:

```
for (int i = 0; i < 3;
i++)
{
    printf("meow\n");
}
```

Ask a question

cat.c style50 cat.c x

Explain Changes Apply Changes

```
1- #include<stdio.h>
2
3 int main(void)
4 {
5- for(int i=0; i<3; i++)
6- {
7- printf("meow\n");
8- }
9 }
```

```
1+ #include <stdio.h>
2
3 int main(void)
4 {
5+ for (int i = 0; i < 3; i++)
6+ {
7+ printf("meow\n");
8+ }
9 }
```

# AI in Teaching Computer Science: Work in Progress

1. Explain highlighted lines of code.
2. Advise students on how to improve their code's style.
3. Answer (most of the) questions asked online by students.
4. Help students find bugs in their code, via TA-like rhetorical questions.
5. Explain arcane error messages, hint how to solve.
6. Assess design of programs, provide qualitative feedback.
7. Enable other faculty to tailor CS50's tools for their own (non-CS) courses.
8. Assess students' understanding via conversations with AI (a la oral exams)
9. ...



John Harvard **STUDENT**



What is flask exactly?

[Comment](#) [Edit](#) [Delete](#) [Endorse](#) ...



CS50 Duck BOT



ENDORSED



Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries, it has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

[Comment](#) [Edit](#) [Delete](#) [Unendorse](#) ...





John Harvard STUDENT



For the Caesar problem, I would like to pass only one argument to the main function. However, I get an interesting error when I try this:

```
caesar.c:11:5: error: only one parameter on 'main' declaration [-Werror,-Wmain]
```

If I follow the lecture and add two variables, this code runs. But it seems inefficient, given that one variable is not utilised. I note that there are certain coding "standards" that govern this behaviour, but I'm not sure that I understand the rationale here.

Is there a more efficient way to write this code?

[Comment](#) [Edit](#) [Delete](#) [Endorse](#) [...](#)



CS50 Duck BOT



3



It seems like you're trying to declare the `main` function with only one parameter. However, in C, the `main` function should have either zero parameters or two parameters, which are `int argc` and `char *argv[]`. The first parameter, `argc`, represents the number of command-line arguments, and the second parameter, `argv`, is an array of strings representing each of those arguments.

For the Caesar problem, you'll want to use these parameters to access the command-line argument that represents the key. To do this, you should declare your `main` function as follows:

```
int main(int argc, char *argv[])
{
    // Your code here
}
```

Then, you can check if the user has provided the correct number of arguments (in this case, just one additional argument for the key) and handle the input accordingly.

P.S. I am an experimental bot. Quack. Do not assume that my reply is accurate unless you see that it's been "endorsed" by (human) staff. Quack.

Comment Edit Delete Unendorse ...



John Harvard **STUDENT**

Thank you, weirdly informative robot

❤️ 9 Reply Edit Delete ...

# AI in Teaching Computer Science: *Work in Progress*

1. Explain highlighted lines of code.
2. Advise students on how to improve their code's style.
3. Answer (most of the) questions asked online by students.
4. Help students find bugs in their code, via TA-like rhetorical questions.
5. Explain arcane error messages, hint how to solve.
6. Assess design of programs, provide qualitative feedback.
7. Enable other faculty to tailor CS50's tools for their own (non-CS) courses.
8. Assess students' understanding via conversations with AI (a la oral exams)
9. ...

**ddb**

Quack. I am CS50's duck debugger (ddb), an experimental AI for [rubberducking](#). Quack. My replies might not always be accurate, so always think critically and let me know if you think that I've erred. Conversations are logged for debugging's sake. Quack.

**you**

My code is not working as expected, any ideas?

```
x = input("Give me an integer x: ")
y = input("Give me an integer y: ")
sum = x + y
print("x + y is: " + sum)
```

**ddb**


It seems like you're trying to add two integers, but the `input()` function in Python returns a string. So, when you try to add `x` and `y`, you're actually concatenating two strings, not adding two integers.

To fix this, you can convert the inputs to integers using the `int()` function like so:

```
x = int(input("Give me an integer x: "))
y = int(input("Give me an integer y: "))
```

# Works in Progress

1. Explain highlighted lines of code.
2. Advise students on how to improve their code's style.
3. Answer (most of the) questions asked online by students.
4. Help students find bugs in their code, via TA-like rhetorical questions.
5. Explain arcane error messages, hint how to solve.
6. Assess design of programs, provide qualitative feedback.
7. Enable other faculty to tailor CS50's tools for their own (non-CS) courses.
8. Assess students' understanding via conversations with AI (a la oral exams)
9. ...

 Having trouble with a command?



Source: CS50 Duck (Extension)

Ask for Help

Dismiss

# Works in Progress

1. Explain highlighted lines of code.
2. Advise students on how to improve their code's style.
3. Answer (most of the) questions asked online by students.
4. Help students find bugs in their code, via TA-like rhetorical questions.
5. Explain arcane error messages, hint how to solve.
6. **Assess design of programs, provide qualitative feedback.**
7. Enable other faculty to tailor CS50's tools for their own (non-CS) courses.
8. Assess students' understanding via conversations with AI (a la oral exams)
9. ...



# Virtual Reality Pilot



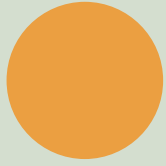


Dr. Adrienne Phelps-Coco

Executive Director of Teaching and Learning

HarvardX





INSTRUCTOR  
DEVELOPMENT  
NIGHT  
2023





THANK YOU





# STUDENT FOCUS GROUPS



# PURPOSEFUL DESIGN/TEACHING

Synchronous  
for Interaction

---

Asynchronous  
for Pacing

---



Engagement  
= Feeling  
Seen

---

# FLEXIBILITY

Predictable  
Policies

---

Participation  
Options

---

Accessibility

---

# DIGITAL ACCESSIBILITY



Harvard's new Digital Accessibility Policy says we should aim to achieve digital accessibility on all newly created content, including Canvas sites, course materials and videos.





# ARTIFICIAL INTELLIGENCE



- Students must acknowledge and cite any use of AI in their work
- If students are unsure whether AI use is permitted, they are responsible for asking
- Instructors may determine their own policies about the use of AI on assignments--and must be clear with students about their policy



# WRITING PROGRAM AI



- The use of AI technologies, including generative AI tools such as ChatGPT, is strictly prohibited in this course. AI technologies may not be used in any phase of the writing process, including brainstorming, outlining, drafting, revising, and editing.



# SYLLABUS POLICY



The Harvard OUE has created sample syllabus language:

- Maximally restrictive policy
- Fully-encouraging policy
- Mixed draft policy

Choose or adapt one for your syllabus.





# SESSION I

- Encouraging Academic Integrity in the Age of ChatGPT
- Teaching Neurodiverse Learners



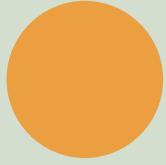
# NETWORKING BREAK

- In person, join us for dessert
- Online, join us for themed breakout rooms



# SESSION 2

- Making the Most of Generative AI in Your Course
- How to Humanize Your Online Course



# DROP IN

- On-the-Spot Accessibility Checks
- Becoming an Instructor for Professional Development Programs
- Brattle Media AV & IT Team Services



# DROP IN

- Introducing the Faculty Portal
- Story Street Studio Tours