Welcome to CSE 143!

Brett Wortzman

Winter 2021

Please make sure your microphone is muted.

If you're willing, turn on your video so we can see you!

Agenda

- About us
- About this course
  - Learning objectives
  - Other similar courses
  - Course components
- Our learning model
- Tools and resources
  - Course Website
  - Zoom
  - Ed
  - PollEverywhere
  - Discord
- Assessment and grading
- Collaboration

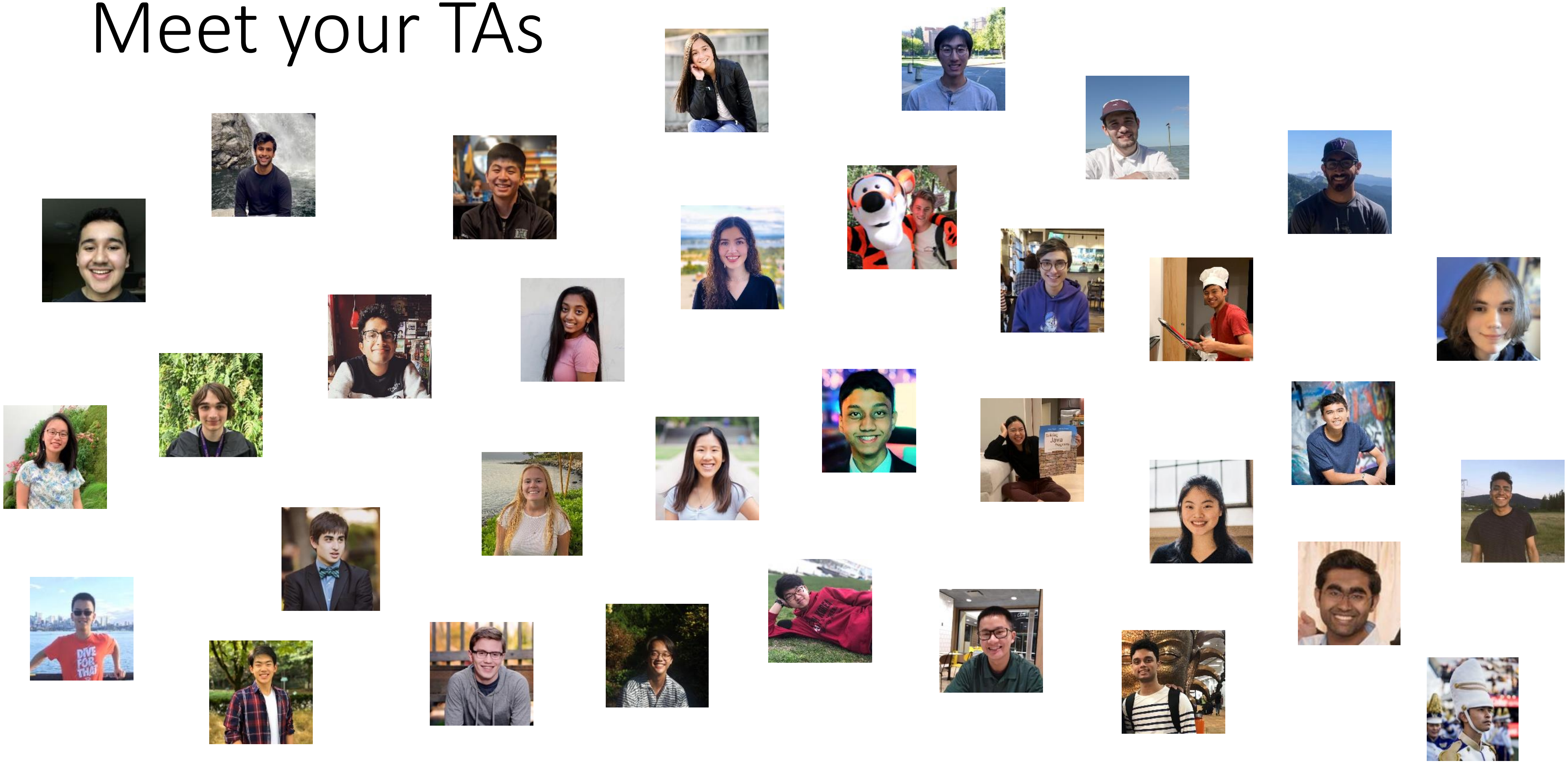
Agenda

- About us ⟵
- About this course
  - Learning objectives
  - Other similar courses
  - Course components
- Our learning model
- Tools and resources
  - Course Website
  - Zoom
  - Ed
  - PollEverywhere
  - Discord
- Assessment and grading
- Collaboration

Hi, I'm Brett! (he/him)

- Assistant Teaching Professor
- Frequent 14X instructor
- Also interested in CS education/pedagogy
- Previously:
  - trained CS teachers
  - developed CS curriculum
  - taught high school CS
  - worked as a software engineer

Meet your TAs



Agenda

- About us
- About this course
  - Learning objectives
  - Other similar courses
  - Course components
- Our learning model
- Tools and resources
  - Course Website
  - Zoom
  - Ed
  - PollEverywhere
  - Discord
- Assessment and grading
- Collaboration

Learning Objectives

or, "What will I learn in this class?"

1. **Functionality/Behavior:** Write functionally correct and efficient Java programs and systems of medium to large length and complexity that meet a provided specification and/or solve a specified problem
2. **Comprehension:** Trace and predict the behavior of programs and systems
3. **Data Abstraction:** Select and apply appropriate abstract data types to manage program state
4. **Data Structures:** Design, implement, and modify data structures to efficiently and effectively provide a defined set of operations
5. **Functional Abstraction:** Document, maintain, and utilize appropriate abstractions between the implementer and client of a library
6. **Decomposition:** Solve problems by breaking them into subproblems and recombining the solutions using techniques such as methods, inheritance, and recursion
7. **Code Quality:** Define programs that are well-written, readable, maintainable, and conform to established standards

Comparison to CSE 142 (or similar)

CSE 142

- Control structures
- Simple (primitive) data
- Client view
- Java as focus
- *How do I do this?*

CSE 143

- Data structures
- Complex data
- Implementer view
- Java as example
- *What can I do with this?*

Other Similar Courses

Course	Good choice if…
CSE 143	• You took CSE 142 OR • You took AP CS A or IB CS in high school OR • You've programmed a fair bit before (especially in Java)
CSE 142	• You've never programmed before OR • You've done a little programming but feel rusty or not confident
CSE 163	• You are interested in programs to manipulate and analyze data AND • You are in, or interested, a major that doesn't require CSE 143
CSE 154	• You are interested in learning to develop web applications

Course Components

Lessons (aka Lectures)

- MWF, 12:30 or 2:30
- Held live via Zoom; recordings released after
- First introductions to course concepts
- Mix of presentation of content and practice activities/problems
- Some required pre-work

Sections

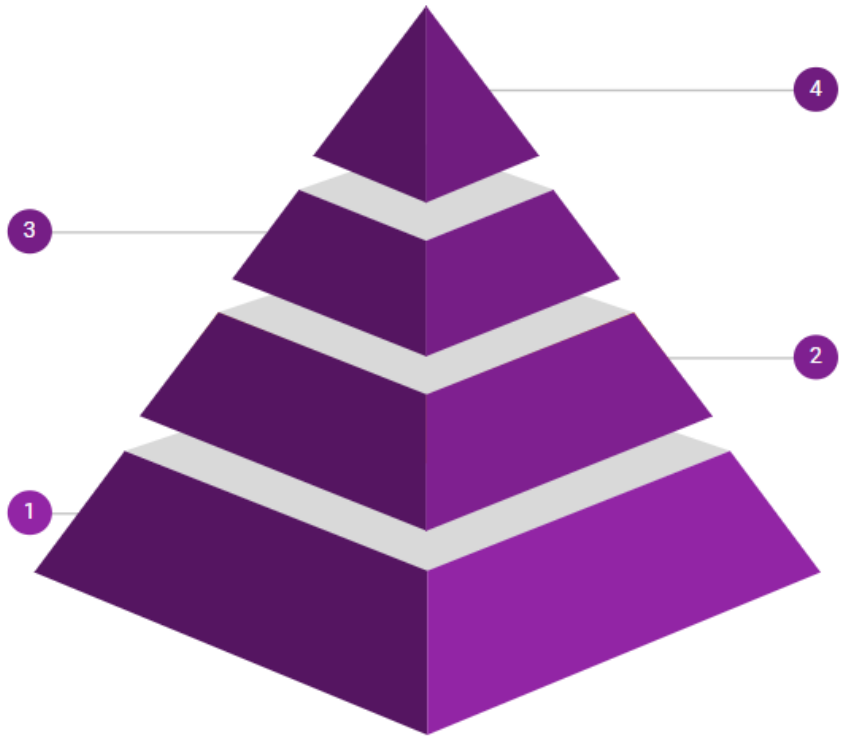
- TTh, various times
- Led by TAs
- Held live via Zoom; ***not*** recorded
  - Short videos will be released on occasion when important material is covered
- Additional review, discussion, and practice
- Mostly practice problems

Attendance is not taken, but you are responsible for all material (including announcements).

Agenda

- About us
- About this course
  - Learning objectives
  - Other similar courses
  - Course components
- Our learning model ⟵
- Tools and resources
  - Course Website
  - Zoom
  - Ed
  - PollEverywhere
  - Discord
- Assessment and grading
- Collaboration

Learning in CSE 143 (or anywhere)



Assessment
Take-home assessments

Build on the scaffolding from 1-3.
Still learning by doing: you're not done!
Demonstrate your mastery
(even if it's still developing).

Guided Practice
Lesson activities, sections

Practice with support from course staff.
Learn by doing: make mistakes and learn from them.
Start to develop mastery.

Independent/Group Practice
Checkpoints, section problems, additional practice

Practice on your own or with classmates.
Continue to **learn by doing.**
Get close to mastery.

Exposure
Lessons, readings, textbook

Encounter concepts for the first time.
See examples and ask questions.
Nowhere near mastery!

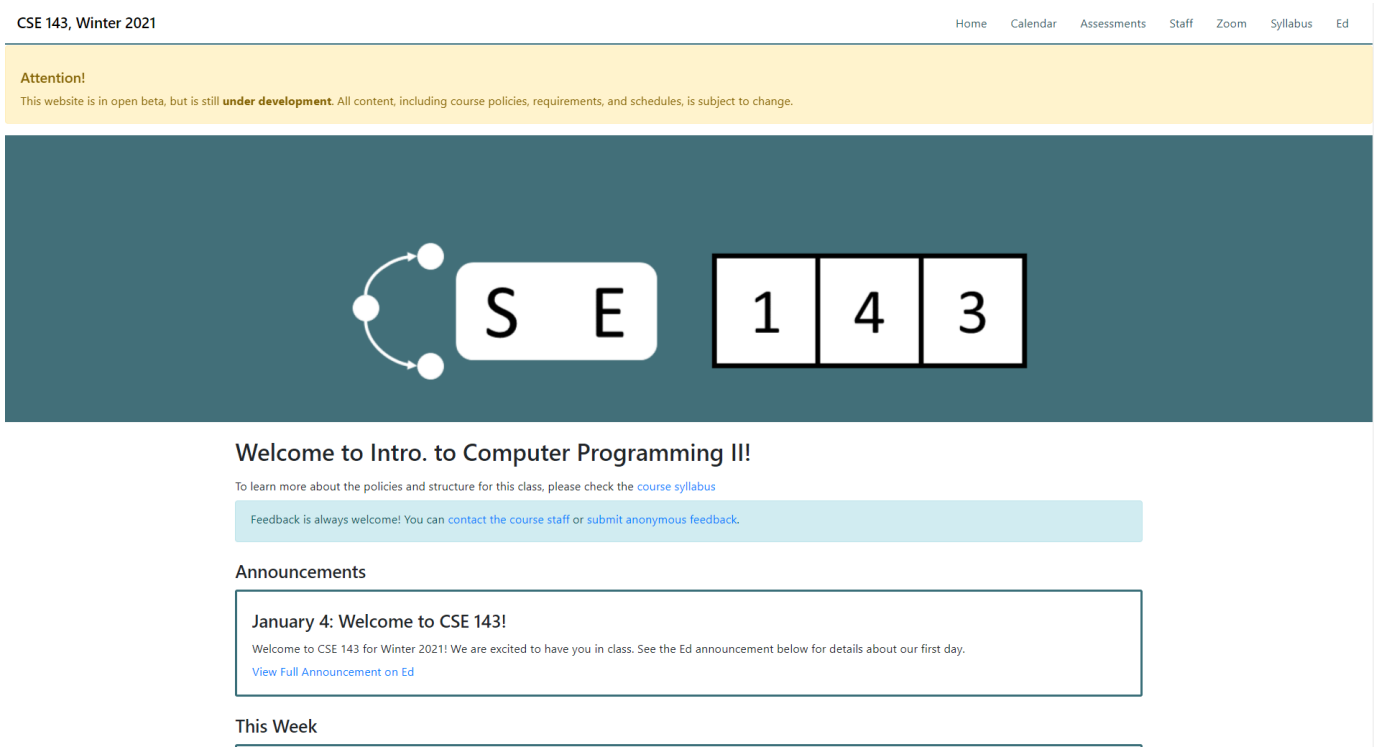
Agenda

- About us
- About this course
  - Learning objectives
  - Other similar courses
  - Course components
- Our learning model
- Tools and resources ⟵
  - Course Website
  - Zoom
  - Ed
  - PollEverywhere
- Assessment and grading
- Collaboration

Course Website

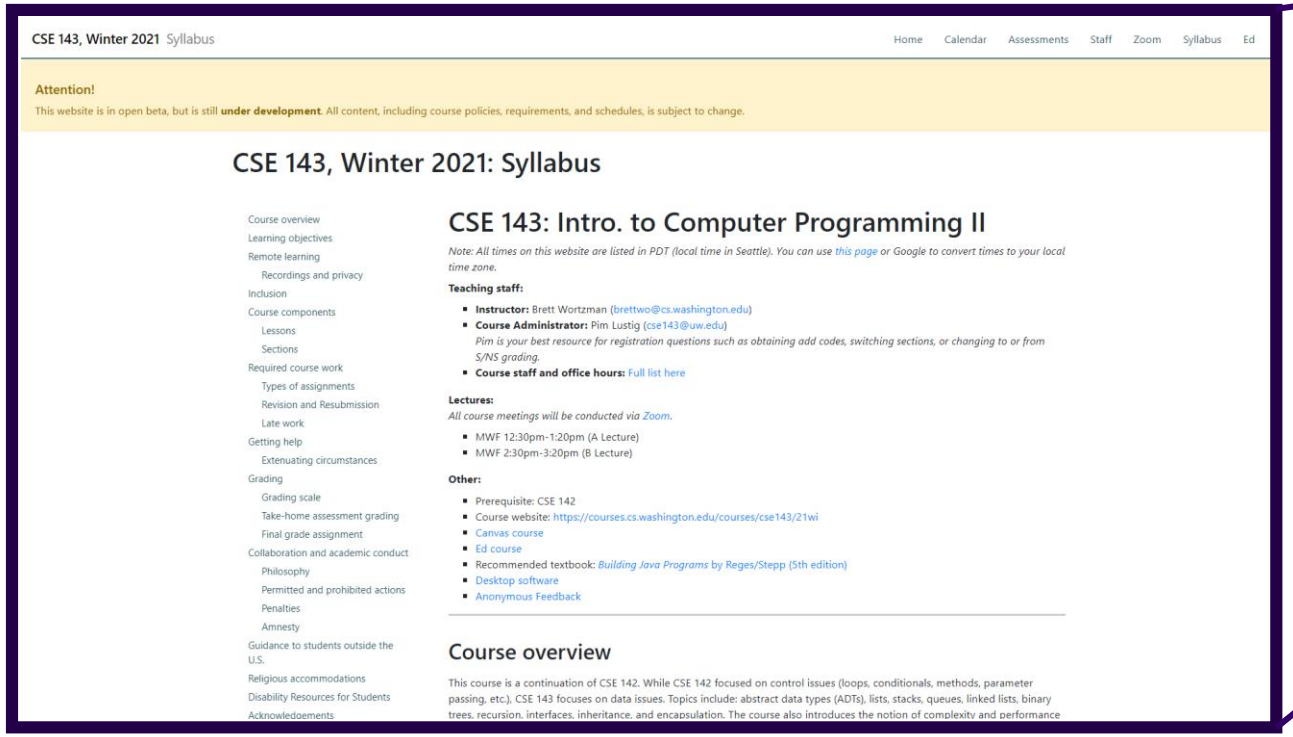
cs.uw.edu/143

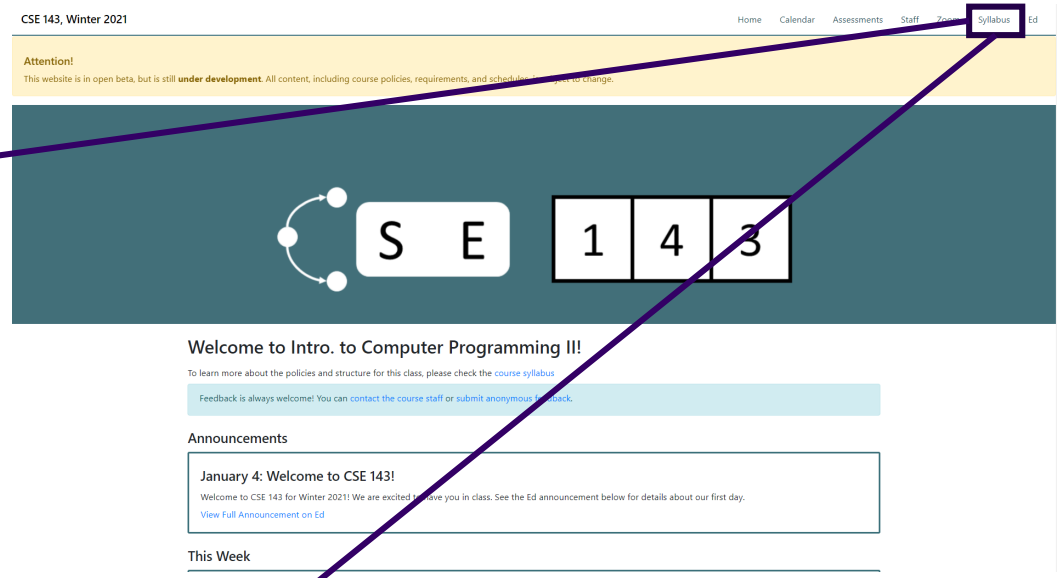
- Primary source of course information (*not* Canvas)
- Calendar will contain links to (almost) all resources



Course Website

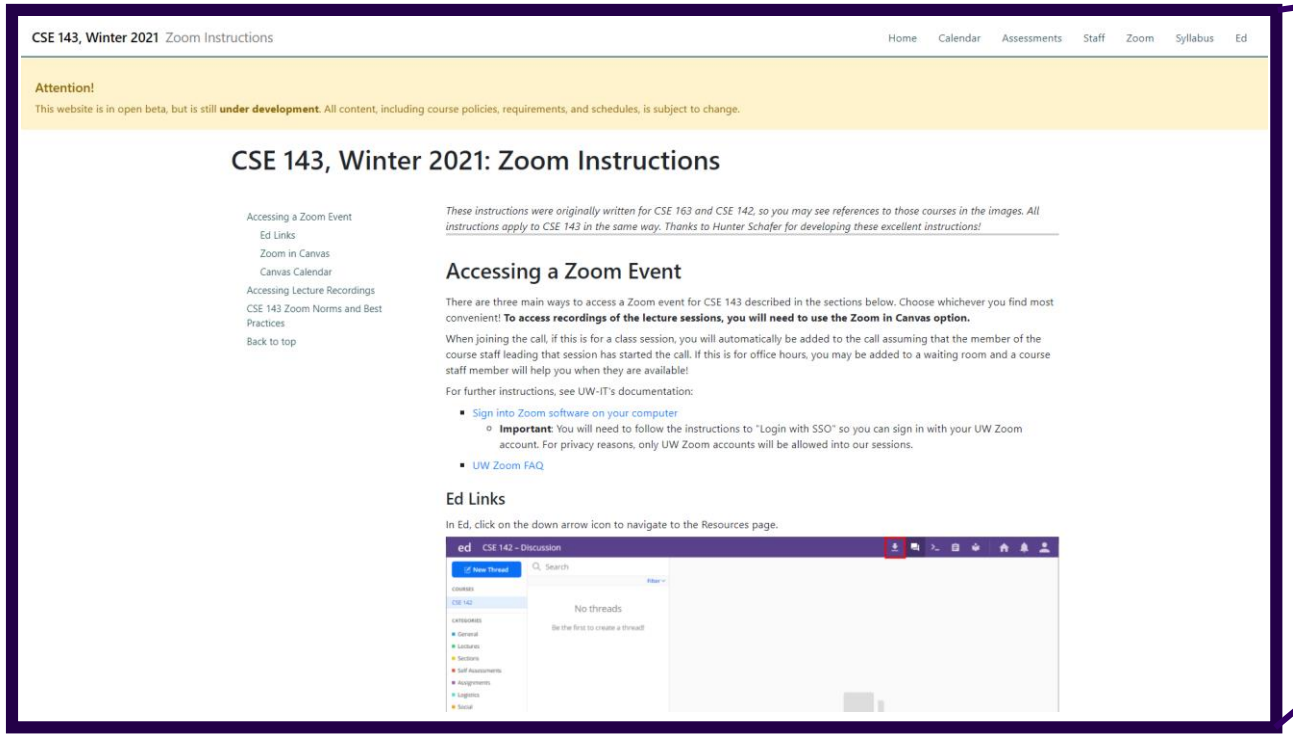
Please review the syllabus ASAP.

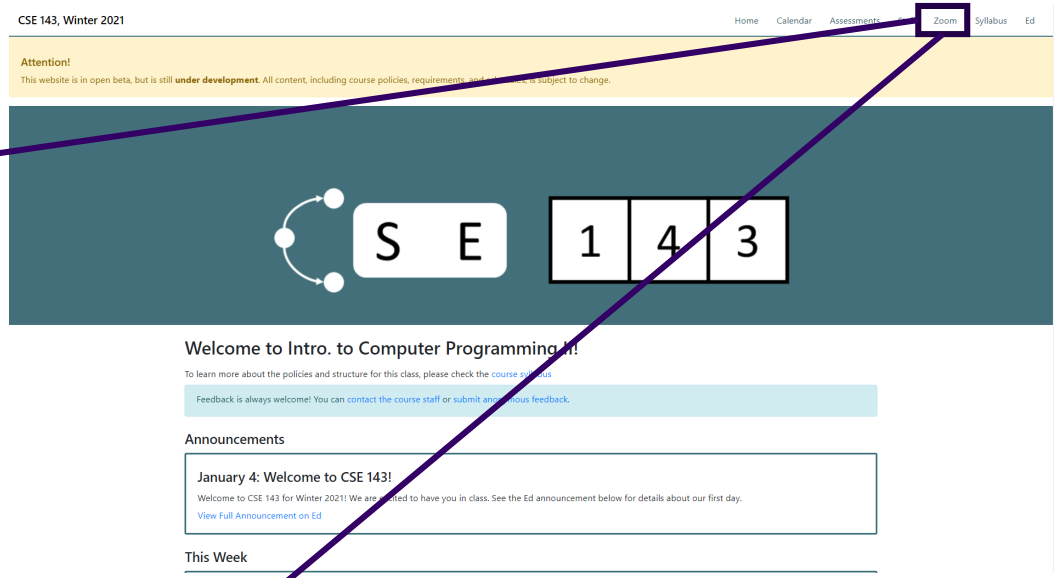




Course Website

Make sure you're aware of Zoom norms.





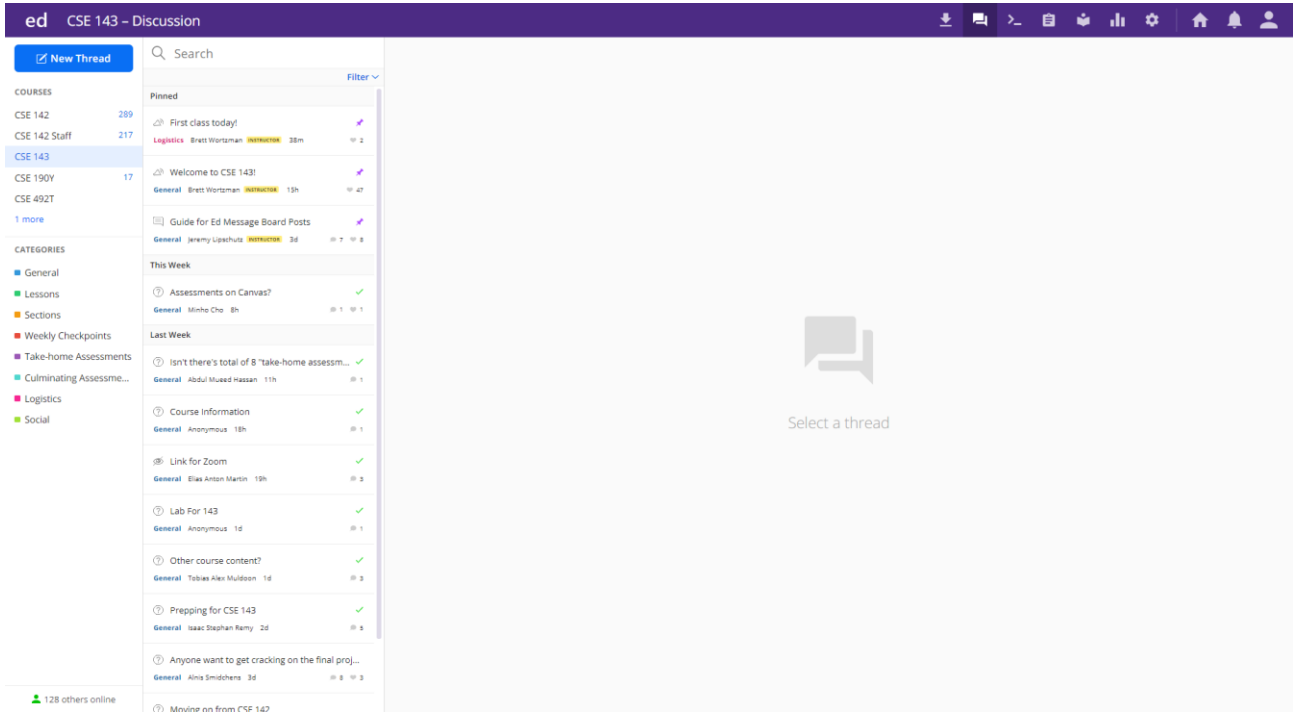
Zoom

- Keep you microphone muted until called on
- Use the “Raise Hand” button to ask to speak

- Turn video on if you’re comfortable!
  - I like seeing your faces. ☺
- Chat will usually be disabled in lessons
  - But we’ll have other options…

Ed

- Our online learning platform
- Lessons, sections, labs, assessments all here
- Intro and walkthrough video forthcoming

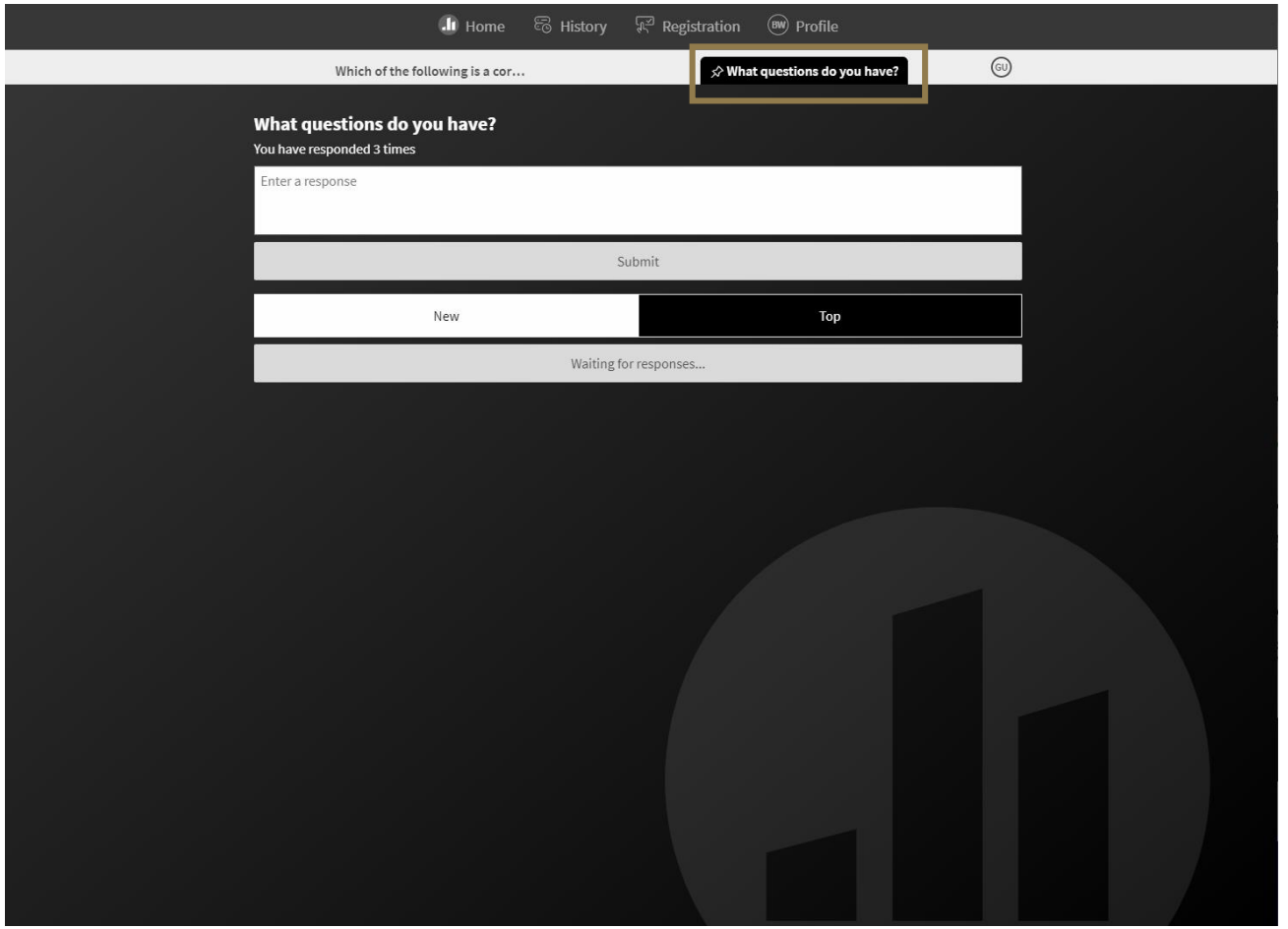


PollEverywhere

pollev.com/brettwo

Two purposes (at least):

- Questions backchannel
  - Ask questions at any time
  - I'll check periodically and respond
  - Some may be deferred
    - Answers will be posted on Ed after class

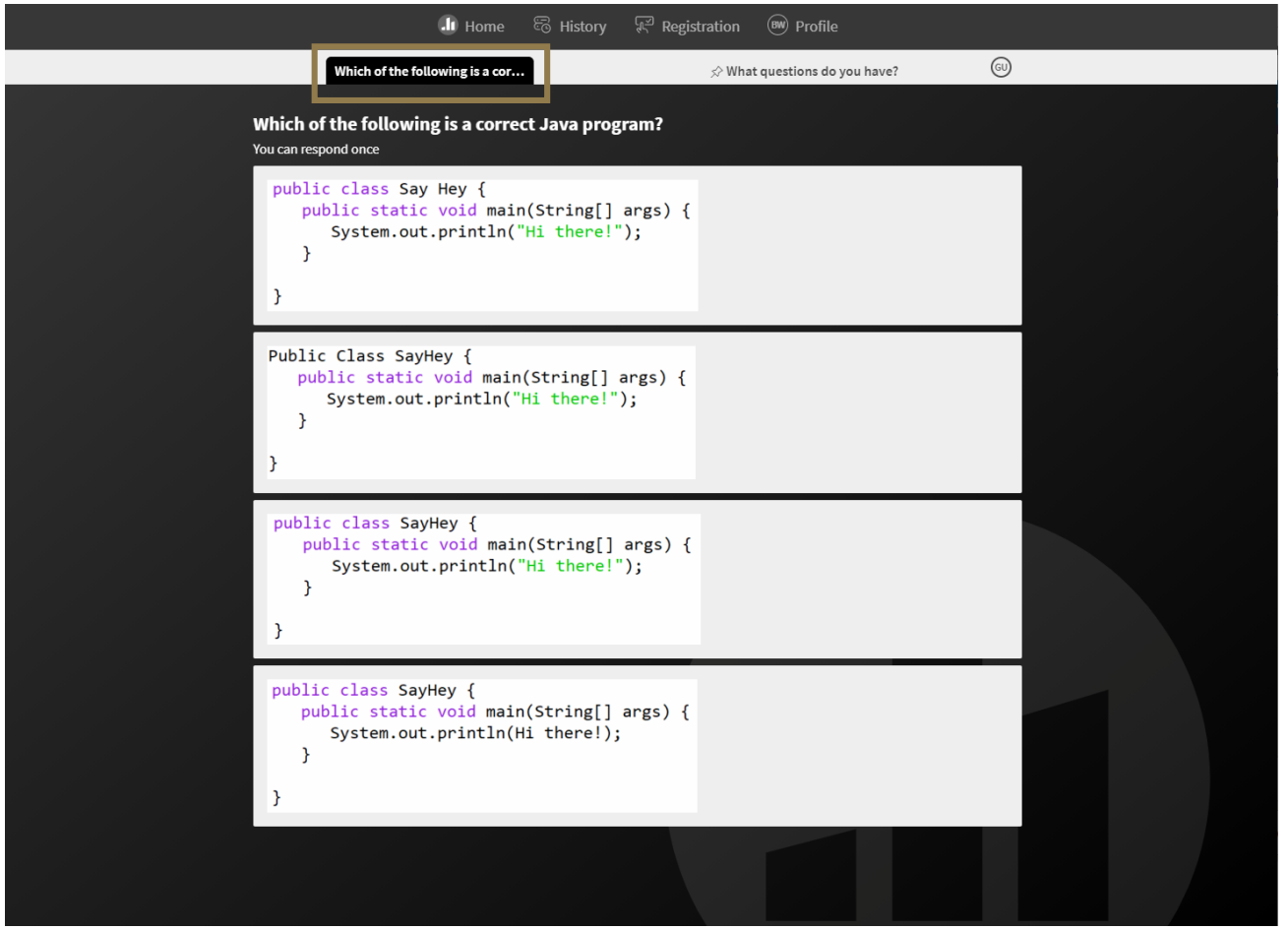


PollEverywhere

pollev.com/brettwo

Two purposes (at least):

- In-class activities
  - Short questions, problems, etc.
  - Usually multiple choice
  - *Not* graded
    - Not even on participation



Agenda

- About us
- About this course
  - Learning objectives
  - Other similar courses
  - Course components
- Our learning model
- Tools and resources
  - Course Website
  - Zoom
  - Ed
  - PollEverywhere
- Assessment and grading ⟵
- Collaboration

Assessment and Grading

- Our goal in the course is for you to **master the concepts and skills** we teach
- We assess your mastery by asking you to apply the concepts and skills on tasks or problems
- By necessity, we are assessing your *work* as a proxy for your *mastery*
- Your final grade should reflect **the extent to which you have demonstrated mastery of the course objectives**

Assessment

- Your learning in this course will be assessed in four ways:
  - **Take-home assessments (~weekly, 8 total)**
    - **Large programming assignments to assess your full mastery of that week's concepts (plus some previous material)**
  - Checkpoints (~weekly, 9-10 total)
    - Short problems to help you practice and make sure you've got the basics for the week
  - Culminating assessments (2 total)
    - Series of problems covering all material up to that point
  - Reflections (w/other assignments, 8-10 total)
    - Written assignments to help you think critically about your learning and progress

Resubmission

Learning takes time, and doesn't always happen on the first try

- One previous take-home assessment can be **resubmitted** each week
  - Initial submission must have been made by original due date and passed verification tests
  - Must be accompanied by a write-up describing changes
  - Grade on resubmission will **completely** replace original grade

- See the syllabus for more details

Grading

Grades should reflect your mastery of the course objectives

- Checkpoints, culminating assessments, and reflections are graded **S (Satisfactory) or N (Not yet)**
  - If you submit on time and meet all requirements, you'll get an S
- Take-home assessments will be grade **E (Exemplary), S (Satisfactory), N (Not yet), or U (Unassessable)** on four dimensions.
  - Specifics forthcoming
- Final grades will be assigned based on the **amount of work at each level**

- See the syllabus for more details

Collaboration Policy

Learning is hard, but it's easier when you learn from each other

- You are encouraged to form study groups, work together on practice and review, and discuss your ideas and approaches **at a high level**
- If you discuss your ideas with others, you must **cite them**
- All work you submit for grading **must be your own**
- Any work found to not be your own will receive a grade of **U on all dimensions and may not be resubmitted**
  - If it's not your work, we can't assess your mastery from it

- See the syllabus for more details

Amnesty

Sometimes, we make bad choices that we regret

- "If you submit work that is in violation of the academic conduct policy, you bring the action to Brett's attention within 72 hours of submission and request amnesty. If you do so, you will receive a grade of U for the initial submission, but you **will be allowed to resubmit your work under the normal resubmission process**."

- See the syllabus for more details