

Welcome to CSE 142!

Omar Ibrahim

Summer 2021

Please make sure your microphone is muted.

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

Agenda

- About us
 - What is Computer Science?
 - About this course
 - Learning objectives
 - Other similar courses
 - Course components
 - Our learning model
- (Wed) →
- Tools and resources
 - Course Website
 - Zoom
 - Ed
 - PollEverywhere
 - Discord
 - Assessment and grading
 - Collaboration

Agenda

- About us ←
 - What is Computer Science?
 - About this course
 - Learning objectives
 - Other similar courses
 - Course components
 - Our learning model (Wed) →
- Tools and resources
 - Course Website
 - Zoom
 - Ed
 - PollEverywhere
 - Discord
 - Assessment and grading
 - Collaboration

Hi, I'm Omar! (he/they)

- Instructional Lecturer
- Recent UW grad!
- B.S. in Computer Science
- Previously...
 - Frequent TA and Head TA for CSE 142



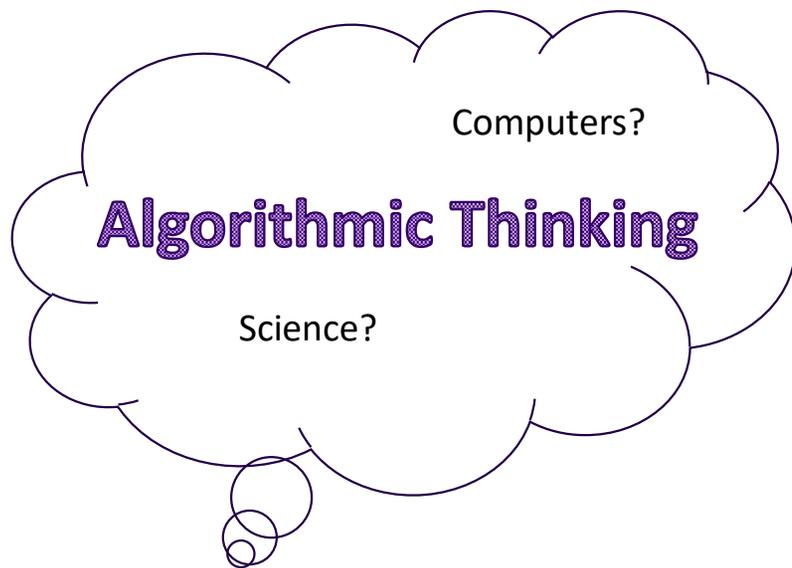
Meet your TAs



Agenda

- About us
- What is Computer Science?
- About this course ←
 - Learning objectives
 - Other similar courses
 - Course components
- Our learning model (Wed) →
 - Assessment and grading
 - Collaboration
- Tools and resources
 - Course Website
 - Zoom
 - Ed
 - PollEverywhere
 - Discord

What is Computer Science?



Algorithm:

a step-by-step procedure for solving a problem or accomplishing some end *especially by a computer*

**Programming is like
a building block**

Learning Objectives

or, “What will I learn in this class?”

- **Functionality/Behavior:** Write functionally correct Java programs that meet a provided specification and/or solve a specified problem
- **Functional Decomposition:** Break down problems into subproblems that are modular and reusable, and define methods to represent those subproblems
- **Control Structures:** Select and apply control structures (e.g. methods, loops, conditionals) to manage the flow of control and information in programs
- **Data Abstraction:** Select and apply basic data abstractions (e.g. variables, parameters, arrays, classes) to manage and manipulate data in programs
- **Code Quality:** Define programs that are well-written, readable, maintainable, and conform to established standards

Other Similar Courses

Course	Good choice if...
CSE 142	<ul style="list-style-type: none">• You've never programmed before OR• You've done a little programming but feel rusty or not confident AND• You are, or want to be, in a major such as CS, CE, EE, Info, etc. that requires Java programming
CSE 143	<ul style="list-style-type: none">• You've programming in Java before OR• You took AP CS A or IB CS in high school
CSE 143X	<ul style="list-style-type: none">• You've programmed a lot before <i>in a language other than Java</i> OR• You are confident you can pick up new concepts very quickly OR• You <i>really, really</i> need to get through two courses in one quarter
CSE 160	<ul style="list-style-type: none">• You've never programmed before AND• You're interested in data science and analysis OR• You'd rather learn Python than Java* OR• You are, or want to be, in a major such as Physics, Bio, Stat, etc. where analyzing data through programming is useful

Course Components

Lectures (*aka Lessons*)

- MWF, 12:00 PM PDT
- 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

- Th, 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).

Course Components

Labs (optional)

- T, various times
- Problems released online, support from TAs over Zoom
- CSE 190 sections **X & Y**
- 1 credit course
- Credit/No Credit grading

Agenda

- About us
- What is Computer Science?
- About this course
 - Learning objectives
 - Other similar courses
 - Course components
- Our learning model ←

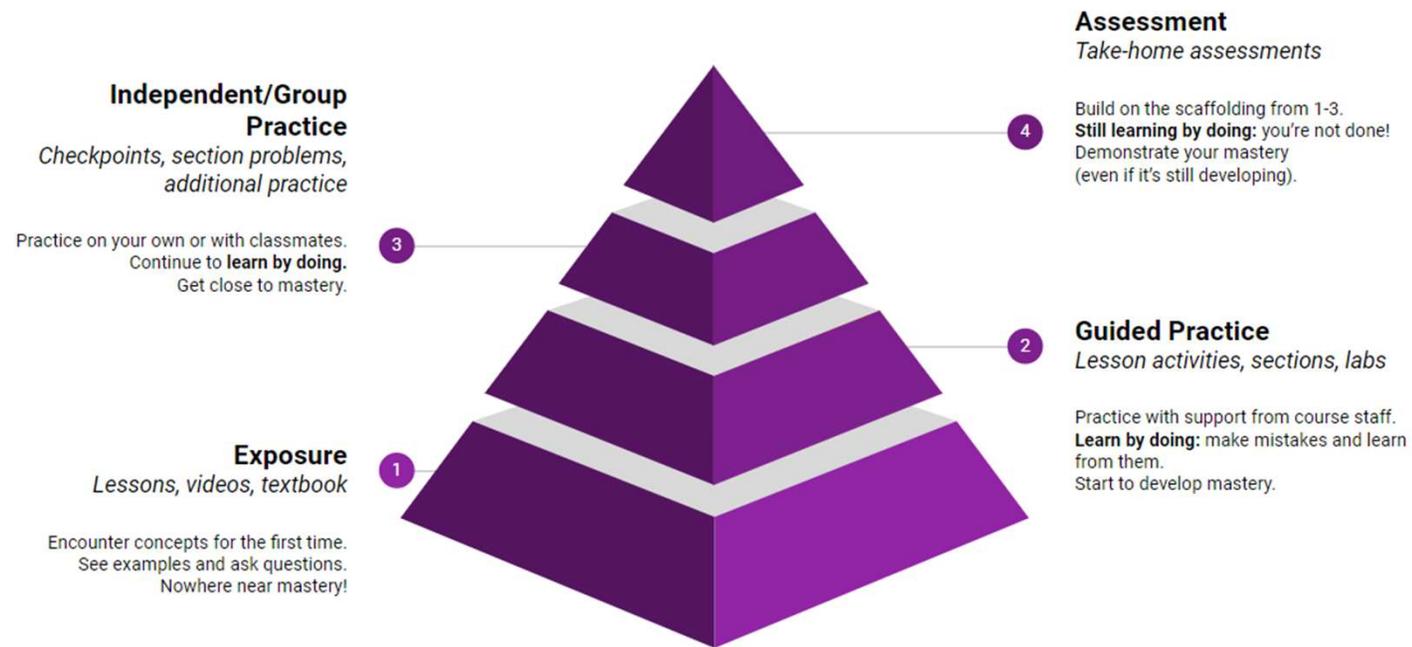
- Tools and resources
 - Course Website
 - Zoom
 - Ed
 - PollEverywhere
 - Discord

- Assessment and grading
- Collaboration

← (Wed)



Learning in CSE 142 (or anywhere)



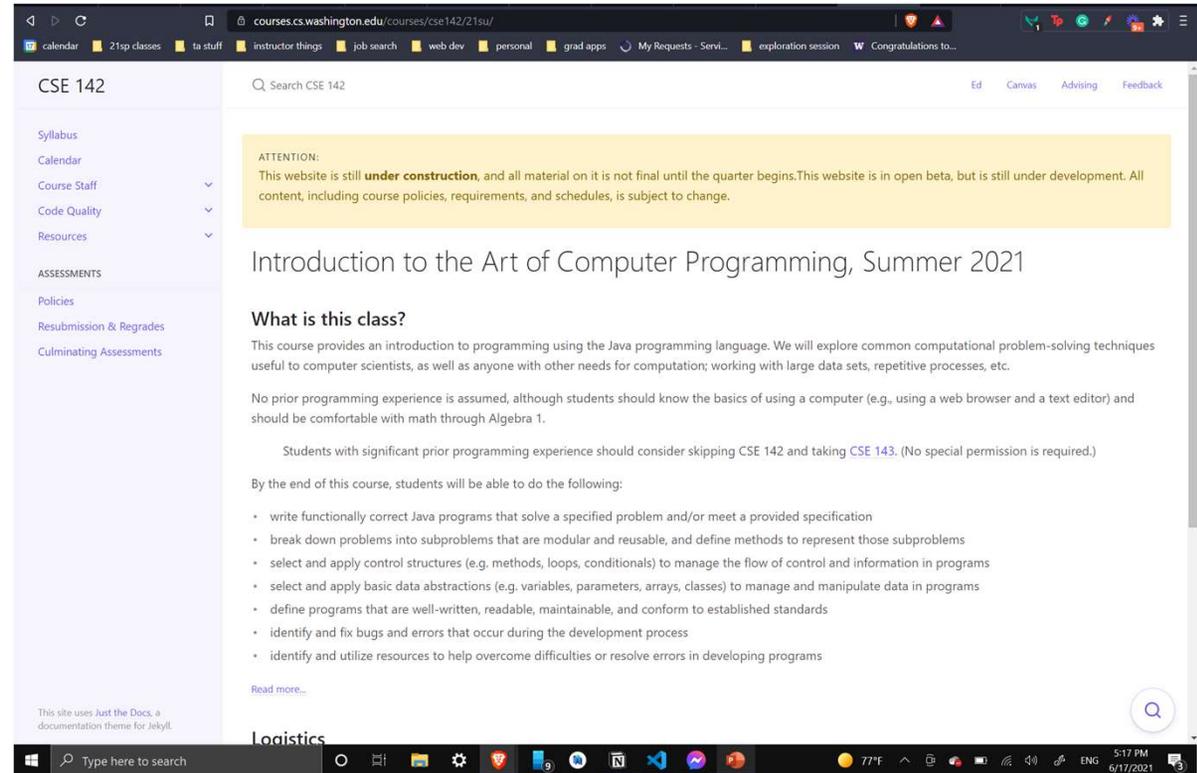
Agenda

- About us
 - What is Computer Science?
 - About this course
 - Learning objectives
 - Other similar courses
 - Course components
 - Our learning model
- (Wed) →
- Tools and resources ←
 - Course Website
 - Zoom
 - Ed
 - PollEverywhere
 - Discord
 - Assessment and grading
 - Collaboration

Course Website

cs.uw.edu/142

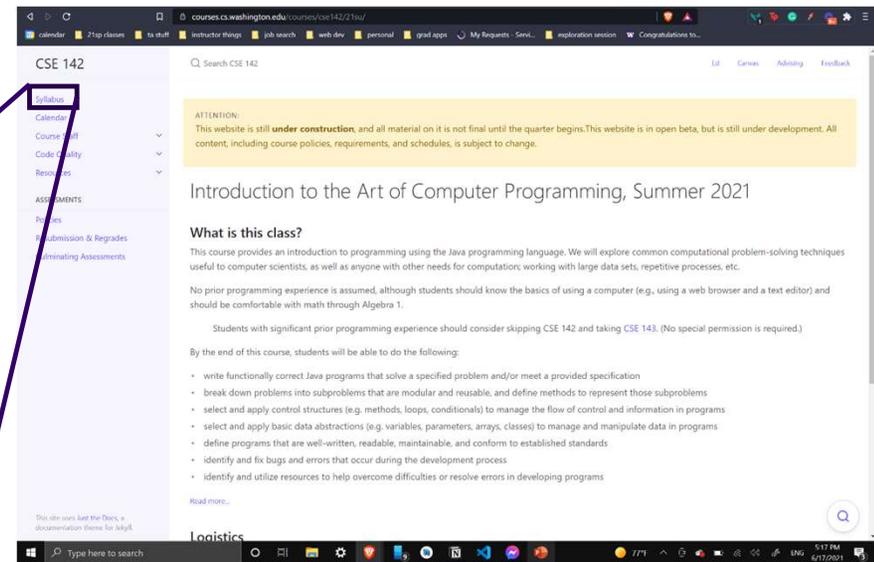
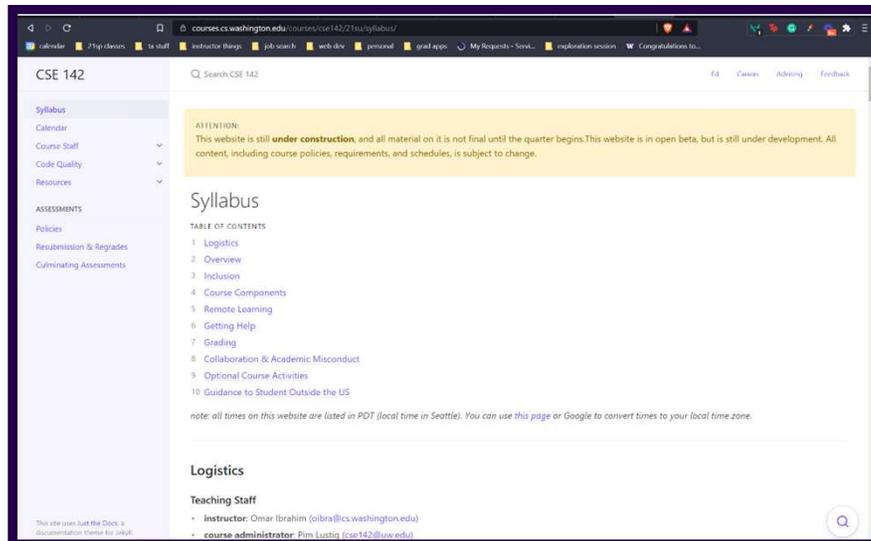
- Primary source of course information (*not* Canvas)
- Calendar will contain links to (almost) all resources



The screenshot shows a web browser displaying the course website for CSE 142 at Washington University in St. Louis. The URL is courses.cs.washington.edu/courses/cse142/21su/. The page features a navigation menu on the left with links for Syllabus, Calendar, Course Staff, Code Quality, Resources, and Assessments. A prominent yellow attention box at the top states: "ATTENTION: This website is still **under construction**, and all material on it is not final until the quarter begins. This website is in open beta, but is still under development. All content, including course policies, requirements, and schedules, is subject to change." Below this, the page title is "Introduction to the Art of Computer Programming, Summer 2021". The section "What is this class?" describes the course as an introduction to programming using Java, covering computational problem-solving techniques. It notes that no prior programming experience is assumed, though basic computer skills are required. A list of learning objectives follows, including writing functional Java programs, breaking down problems into subproblems, applying control structures and data abstractions, and identifying/fixing bugs. A "Logistics" section is partially visible at the bottom.

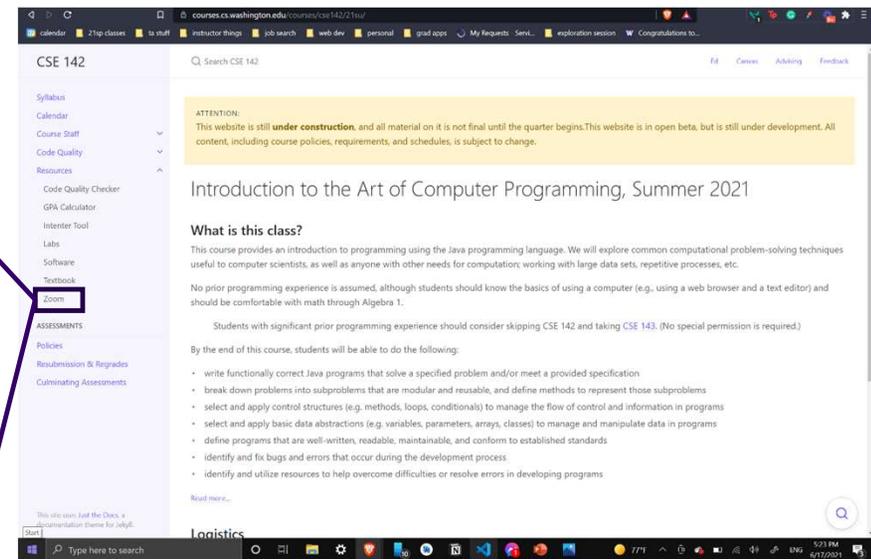
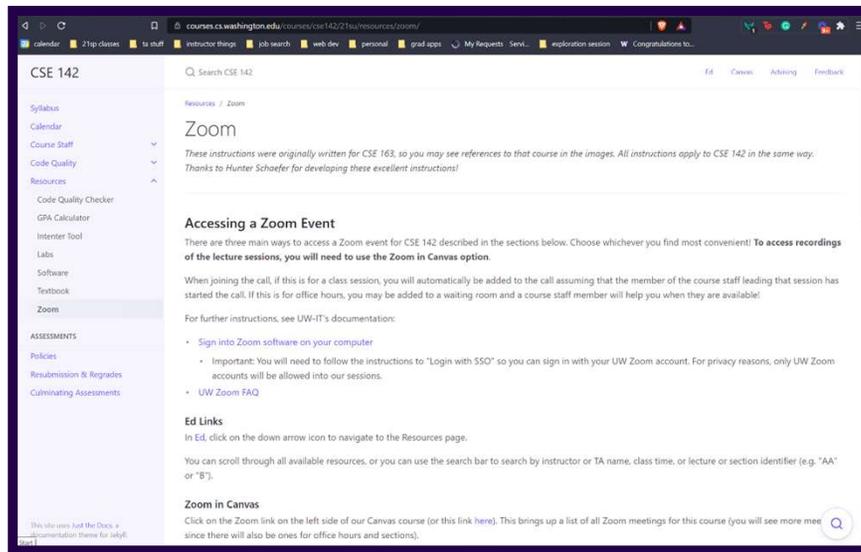
Course Website

Please review the syllabus ASAP.



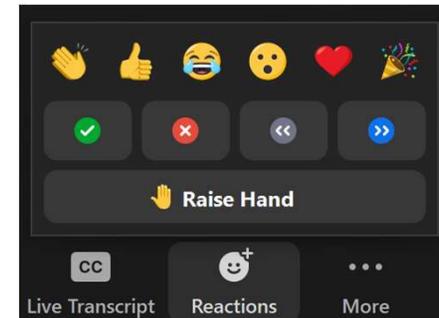
Course Website

Make sure you're aware of Zoom norms.



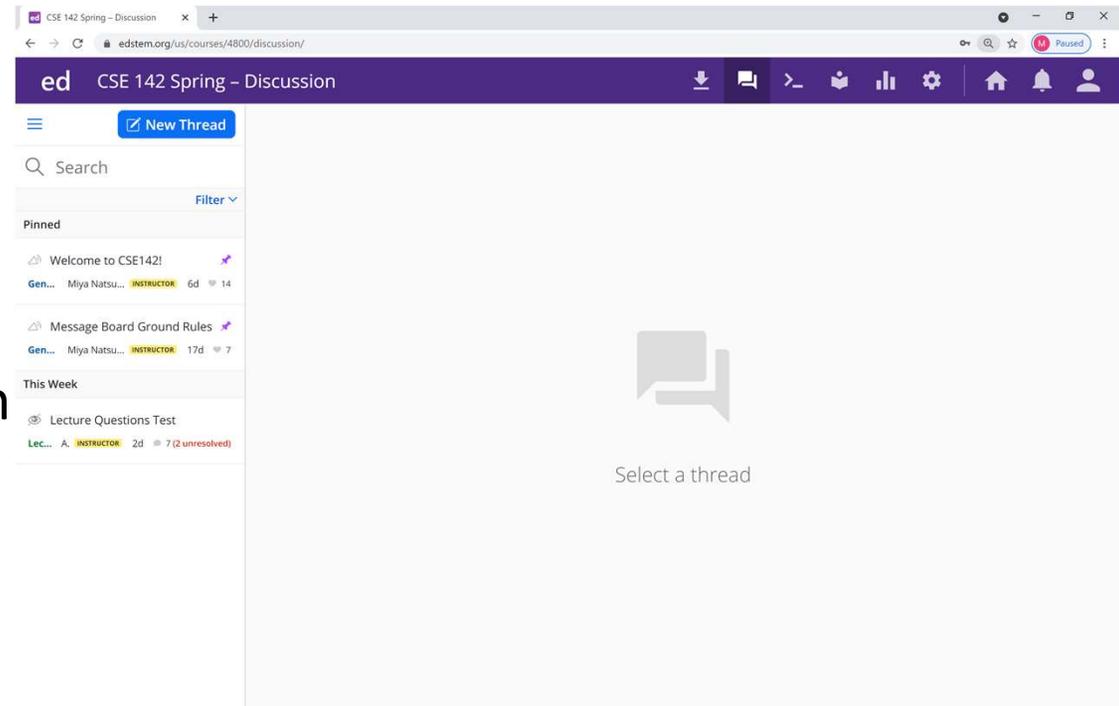
Zoom

- Keep your 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 lectures
 - But we’ll have other options...



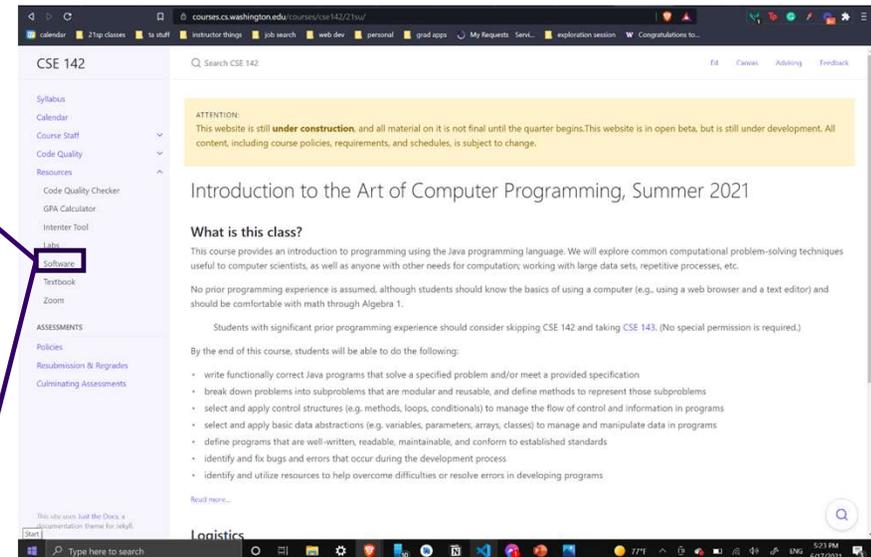
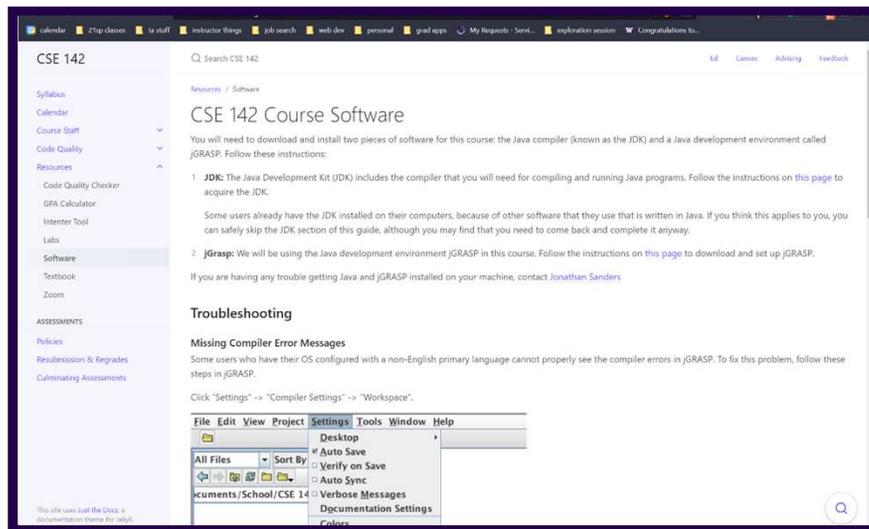
Ed

- Our online learning platform
- Lessons, sections, labs, assessments all here
- Lecture Megathreads for each day that will be monitored by TAs during lecture
- Intro and walkthrough video forthcoming



Software

You will need to install the JDK and jGRASP

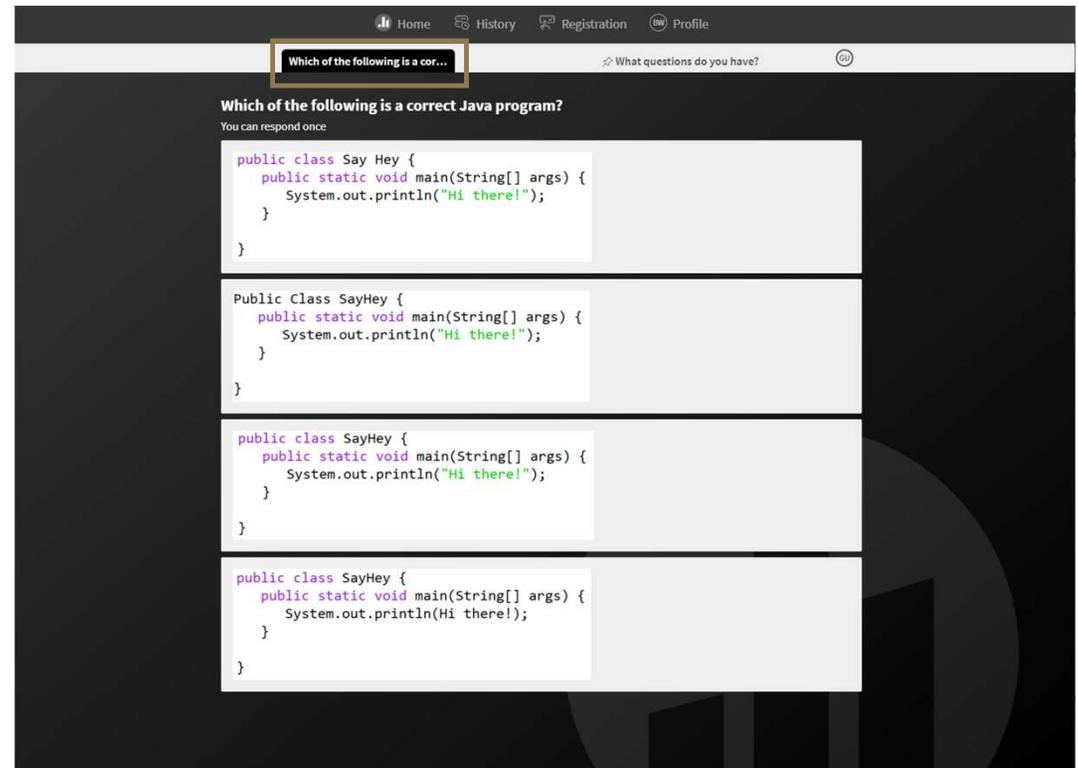


PollEverywhere

pollev.com/oibra

Two purposes (at least):

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

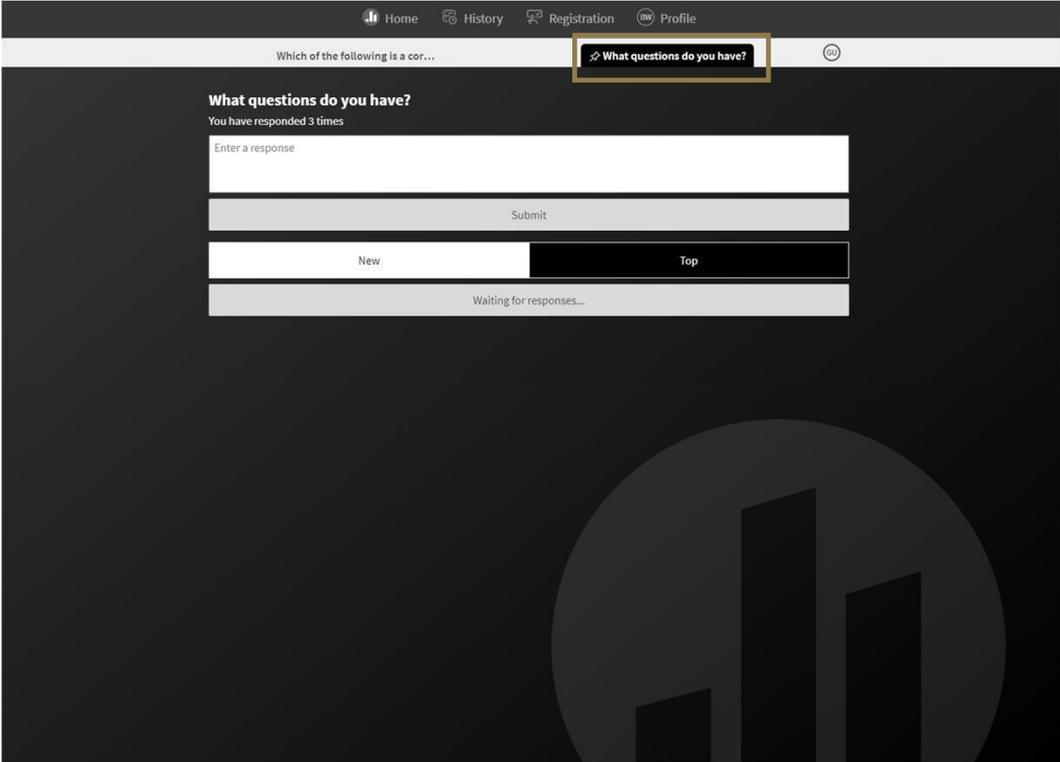


PollEverywhere

pollev.com/oibra

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



The screenshot shows a web browser interface for a poll. At the top, there are navigation links: Home, History, Registration, and Profile. Below the navigation bar, a poll question is displayed: "Which of the following is a cor...". A yellow box highlights the poll title "What questions do you have?". Below the question, there is a text input field labeled "Enter a response", a "Submit" button, and a "New" button. A "Top" button is also visible. At the bottom of the poll area, it says "Waiting for responses...". In the background, there is a faint bar chart graphic.

Agenda

- About us
- What is Computer Science?
- 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

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:
 - Checkpoints (~weekly)
 - Short problems to help you practice and make sure you've got the basics for the week
 - **Take-home assessments (~weekly)**
 - **Large programming assignments to assess your full mastery of that week's concepts (plus some previous material)**
 - Culminating assessments (2/quarter)
 - Series of problems covering all material up to that point
 - Reflections (w/other assignments)
 - 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
 - Must be accompanied by a write-up describing changes
 - Grade on resubmission will replace original grade
 - Must meet (low) minimum bar of effort on original submission
- 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 **U (Unassessable)**
 - 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:
 - Behavior
 - Structure & Design
 - Use of Language Features
 - Documentation & Readability
- 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 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