

# Welcome to CSE 142!

Brett Wortzman

Autumn 2021

# Agenda

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

# Agenda

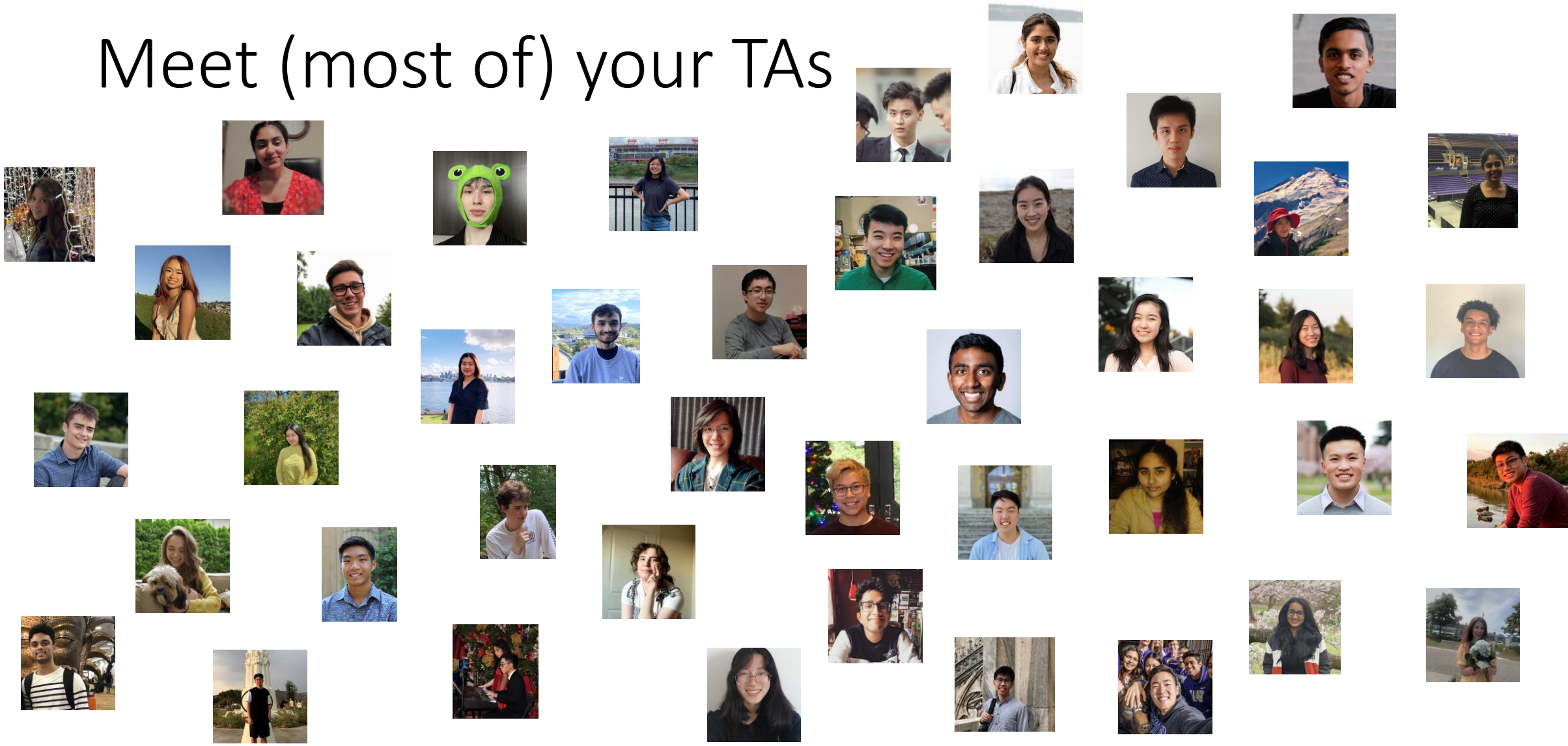
- **About us** ←
- About this course
  - Learning objectives
  - Other similar courses
  - Course components
- Our learning model
- Tools and resources
  - Course Website
  - Ed
  - PollEverywhere
- Assessment and grading
- Collaboration

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

- Assistant Teaching Professor
- Frequent 142 instructor
- Also interested in CS education/pedagogy
- Previously:
  - trained CS teachers
  - developed CS curriculum
  - taught high school CS
  - worked as a software engineer



# Meet (most of) 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?”*

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



# Course Components

## Lessons (aka Lectures)

- MWF, 11:30 or 3:30
- Held live in KNE; 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 in person; **not** recorded
  - Materials will be released online afterwards
- 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
  - Ed
  - PollEverywhere
- Assessment and grading
- Collaboration



# Digression: My Pandemic Hobby

*Amigurumi*: Japanese art of creating crocheted or knitted stuffed toys



# Digression: My Pandemic Hobby

*Amigurumi*: Japanese art of creating crocheted or knitted stuffed toys

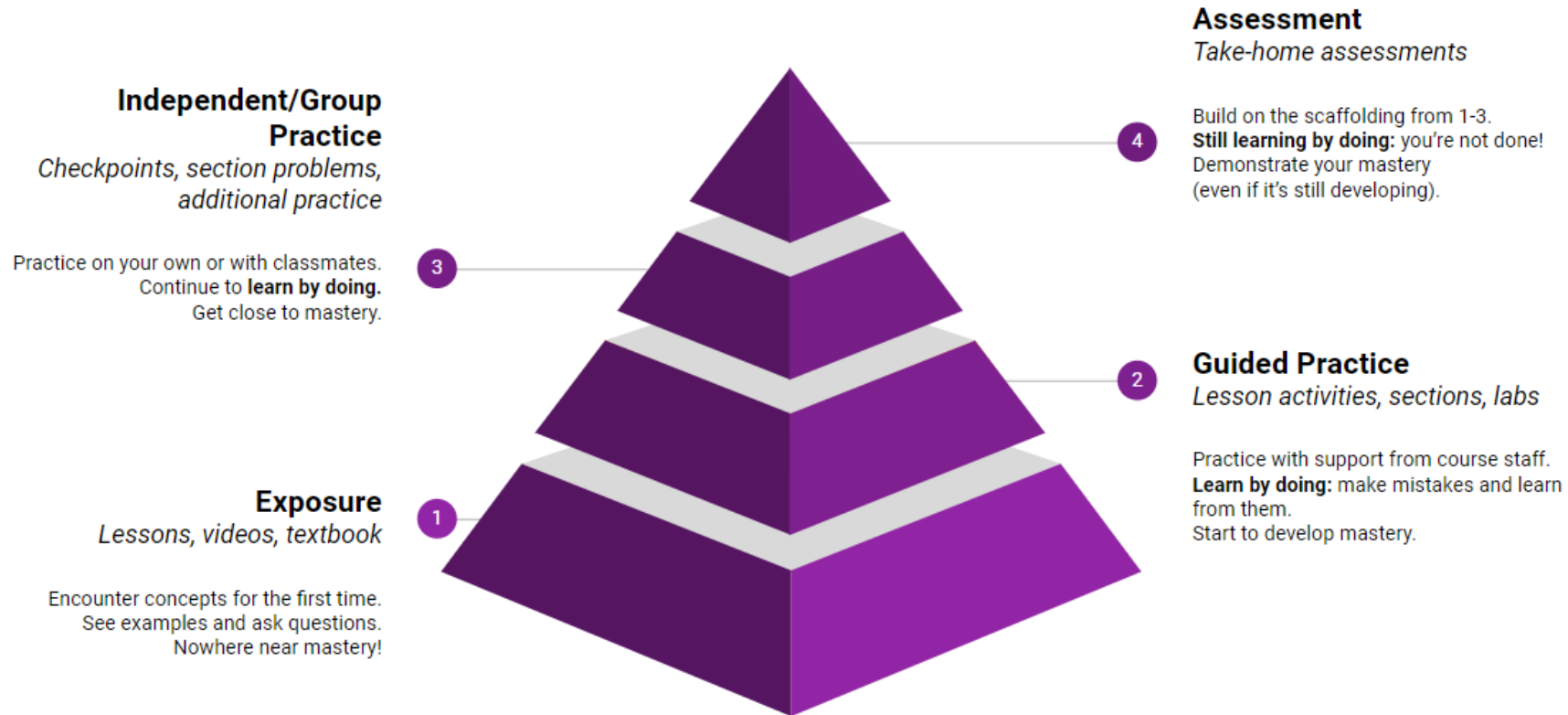


# Digression: My Pandemic Hobby

*Amigurumi*: Japanese art of creating crocheted or knitted stuffed toys



# Learning in CSE 142 (or anywhere)



# Agenda

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



# Course Website

[cs.uw.edu/142](https://cs.uw.edu/142)

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

The screenshot shows the course website for CSE 142, Autumn 2021. The page has a teal header with the course title "CSE 142 Computer Programming I". Below the header, there is a yellow "Attention!" banner stating that the website is in open beta and under development. The main content area is white and contains a "Welcome to Intro. to Computer Programming I!" message, a feedback link, an "Announcements" section with a "September 29: Welcome to CSE 142!" announcement, and a "This Week" section with "Week 1: Welcome!; Introduction to Java; Printing; Static Methods".

CSE 142, Autumn 2021 Home Calendar Assessments Staff Syllabus Ed

**Attention!**  
This website is in open beta, but is still **under development**. All content, including course policies, requirements, and schedules, is subject to change.

## CSE 142

Computer Programming I

### Welcome to Intro. to Computer Programming I!

To learn more about the policies and structure for this class, please check the [course syllabus](#)

Feedback is always welcome! You can [contact the course staff](#) or [submit anonymous feedback](#).

#### Announcements

**September 29: Welcome to CSE 142!**  
Welcome to CSE 142 for Autumn 2021! We are excited to have you in class. See the Ed announcement below for details about our first day.

#### This Week

**Week 1: Welcome!; Introduction to Java; Printing; Static Methods**

# Course Website

*Please review the syllabus ASAP.*

CSE 142, Autumn 2021 Syllabus

Home Calendar Assessments Staff Syllabus Ed

**Attention!**  
This website is in open beta, but is still **under development**. All content, including course policies, requirements, and schedules, is subject to change.

## CSE 142, Autumn 2021: Syllabus

Course overview  
Learning objectives  
Inclusion  
Course components  
Lessons  
Sections  
Getting help  
Exenuating circumstances: "Don't suffer in silence!"  
Disability Resources for Students  
Religious accommodations  
Required course work  
Types of assignments  
Revision and Resubmission  
Late work  
Grading  
Grading scale  
Take-home assessment grading  
Final grade assignment  
Optional course activities  
Labs (CSE 190)  
Exploration sessions  
Collaboration and academic conduct  
Philosophy  
Permitted and prohibited actions  
Penalties  
Amnesty  
Guidance to students outside the U.S.

### CSE 142: Intro. to Computer Programming I

**Teaching staff:**

- **Instructor:** Brett Wortzman ([brettwo@cs.washington.edu](mailto:brettwo@cs.washington.edu))
- **Course Administrator:** Pim Lustig ([cse143@uw.edu](mailto:cse143@uw.edu))  
*Pim is your best resource for registration questions such as obtaining add codes, switching sections, or changing to or from S/NS grading.*
- **Course staff and support hours:** [Full list here](#)

**Class meetings:**

- MWF 11:30am-12:20pm, KNE 130 (A Lecture)
- MWF 3:30pm-4:20pm, KNE 120 (B Lecture)

**Other:**

- Course website: <https://courses.cs.washington.edu/courses/cse142/21au>
- [Ed course](#)
- Recommended textbook: *Building Java Programs* by Reges and Stepp (5th edition)
- [Desktop software](#)
- [Anonymous Feedback](#)

### Course overview

This course provides an introduction to programming using the Java programming language. We will explore common computational problem-solving techniques useful to computer scientists, but also to anyone who has large data sets, repetitive processes or other needs for computation. No prior programming experience is assumed, although students should know the basics of using a computer (e.g., using a web browser and a text editor) and should be comfortable with math through Algebra 1. Students with significant prior programming experience should consider skipping CSE 142 and taking [CSE 143](#) or [CSE 143X](#). (No special permission is required.)

### Learning objectives

CSE 142, Autumn 2021

Home Calendar Assessments Staff Syllabus Ed

**Attention!**  
This website is in open beta, but is still **under development**. All content, including course policies, requirements, and schedules, is subject to change.

# CSE 142

## Computer Programming I

### Welcome to Intro. to Computer Programming !!

To learn more about the policies and structure for this class, please check the [course syllabus](#)

Feedback is always welcome! You can [contact the course staff](#) or [submit anonymous feedback](#).

### Announcements

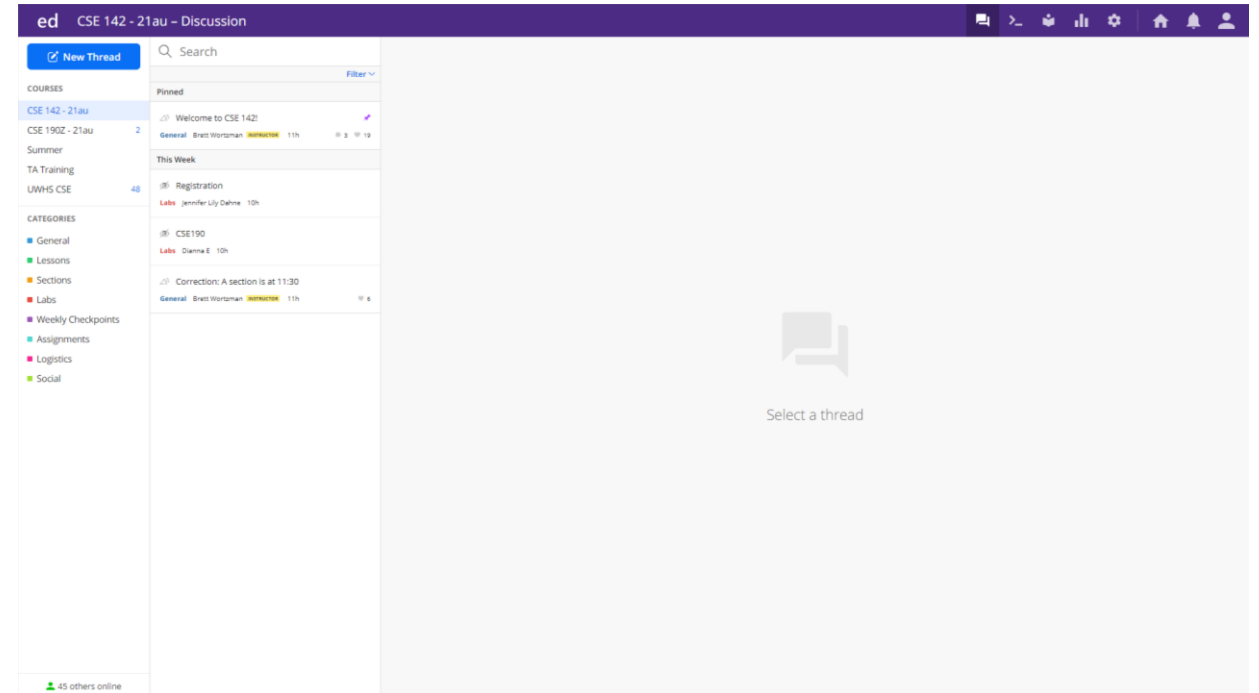
**September 29: Welcome to CSE 142!**  
Welcome to CSE 142 for Autumn 2021! We are excited to have you in class. See the Ed announcement below for details about our first day.

### This Week

**Week 1: Welcome! Introduction to Java, Printing; Static Methods**

# Ed

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

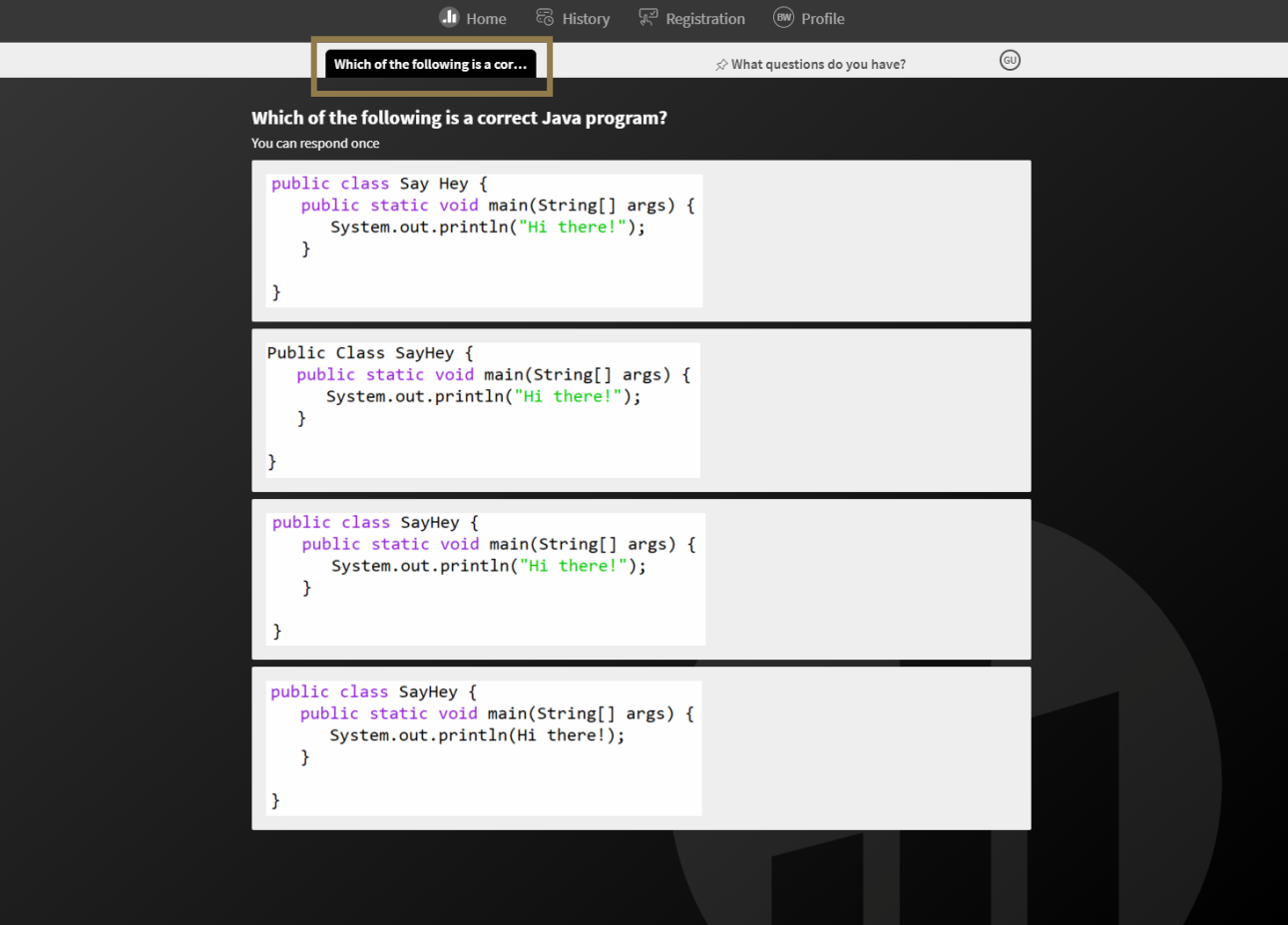


# PollEverywhere

[pollev.com/brettwo](http://pollev.com/brettwo)

Two purposes (at least):

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



The screenshot shows a web browser interface for a poll. At the top, there are navigation links: Home, History, Registration, and Profile. Below that, a search bar contains the text "Which of the following is a cor..." and a link "What questions do you have?". The main content area displays the question: "Which of the following is a correct Java program?" and a note "You can respond once". There are four code snippets listed as options:

```
public class Say Hey {  
    public static void main(String[] args) {  
        System.out.println("Hi there!");  
    }  
}
```

```
Public Class SayHey {  
    public static void main(String[] args) {  
        System.out.println("Hi there!");  
    }  
}
```

```
public class SayHey {  
    public static void main(String[] args) {  
        System.out.println("Hi there!");  
    }  
}
```

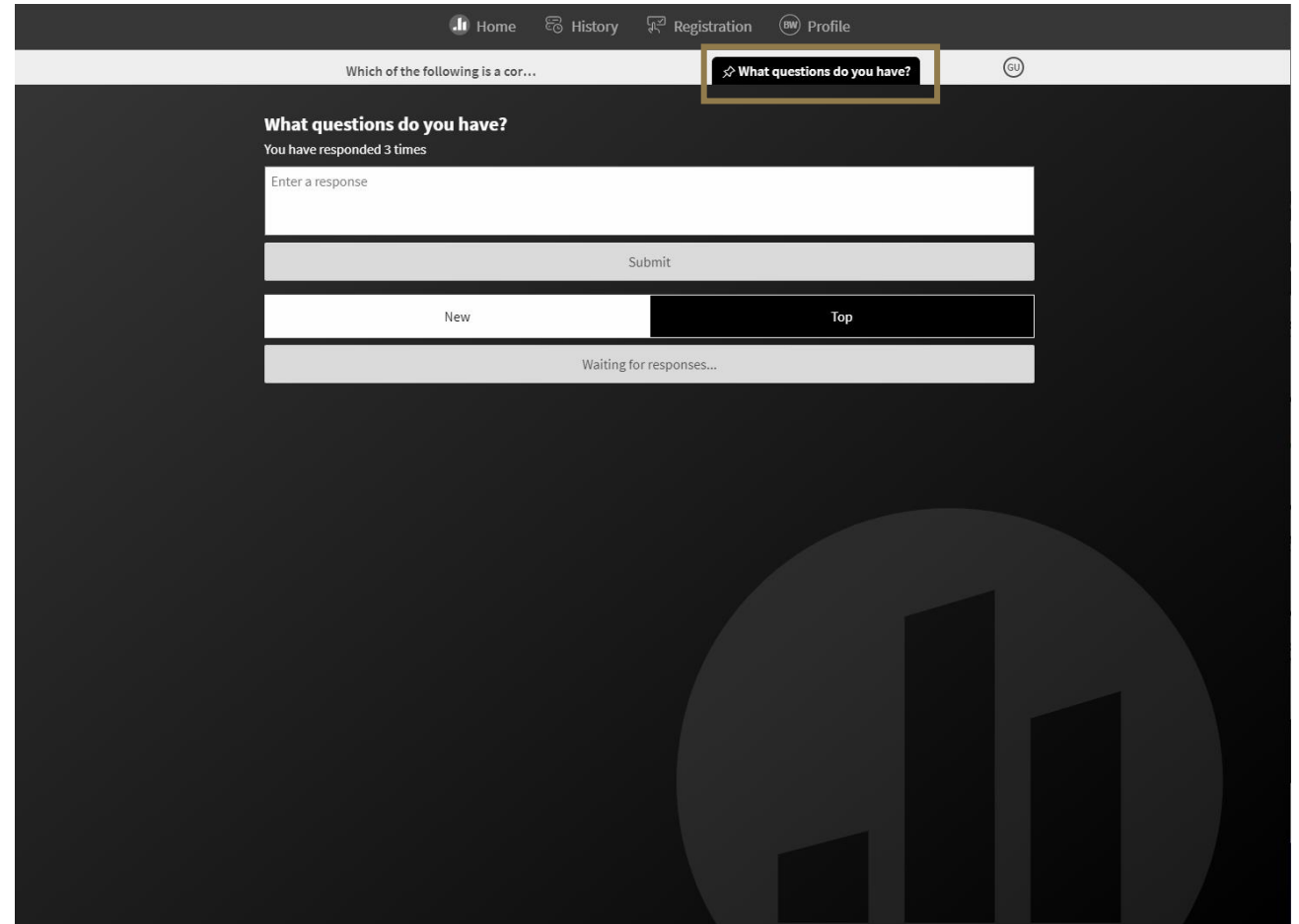
```
public class SayHey {  
    public static void main(String[] args) {  
        System.out.println(Hi there!);  
    }  
}
```

# PollEverywhere

[pollev.com/brettwo](https://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



# Agenda

- About us
- About this course
  - Learning objectives
  - Other similar courses
  - Course components
- Our learning model
- Tools and resources
  - Course Website
  - 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
  - Must be accompanied by a write-up describing changes
  - Grade on resubmission will 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)**, or **N (Not yet)** on four dimensions:
  - Behavior
  - Structure and Design
  - Use of Language Features
  - Code Quality
- Under certain circumstances, a grade of U (Unassessable) may be assigned
- Final grades will be assigned based on the **amount of work at each level**
- See the [syllabus](#) for more details

# Agenda

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

# 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