

LEC 00

CSE 122

Welcome!



Questions during Class?
Raise hand or send here

sli.do #cse122



BEFORE WE START

Talk to your neighbors:
Introduce yourself to your neighbor!

What is your name? Major? Why are you taking this class?

Music: [cse 122 su 23 tunes](#)

Instructor	Melissa Lin	
TAs	Poojitha Arangam	Audrey Lin
	Darel Gunawan	Di Mao
	Colton Harris	Steven Nguyen
	Atharva Kashyap	Ben Wang
	Eesha Kunisetty	Jaylyn Zhang

Lecture Outline

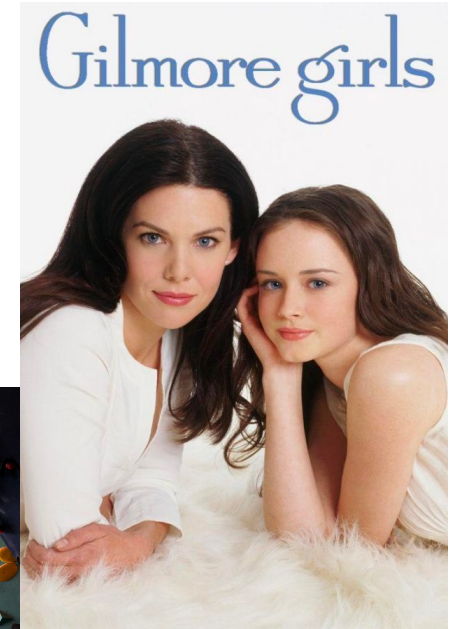
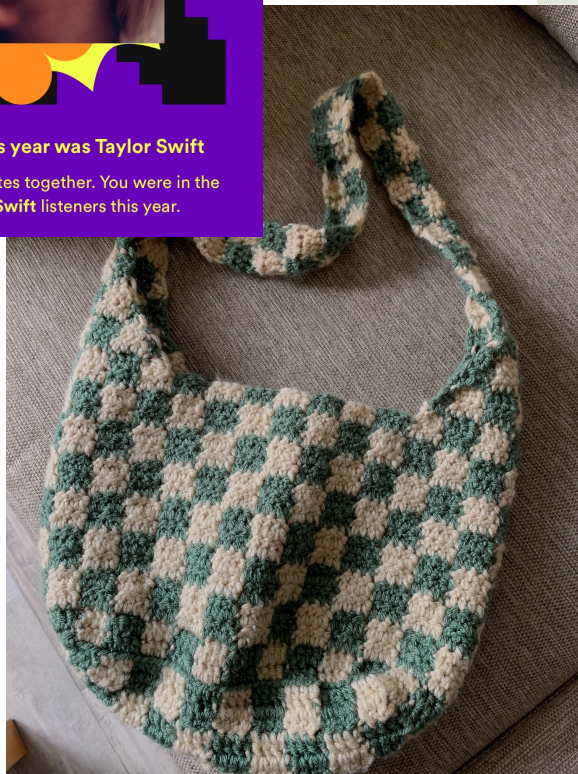
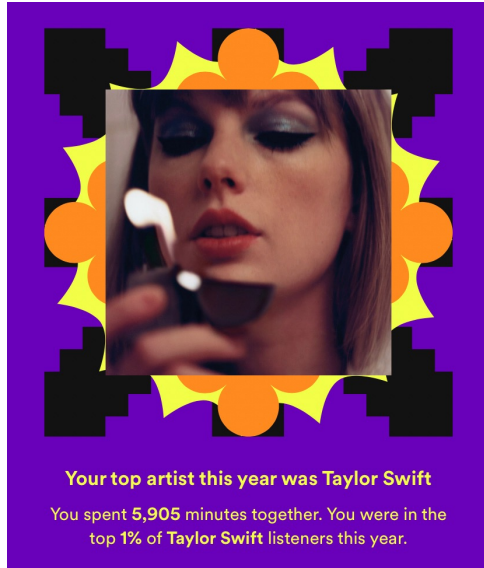
- **Introductions** 
- About this Course
 - Course Components & Tools
 - Policies
 - Making the Most of this Class
- Intro/Review Java

Course Staff

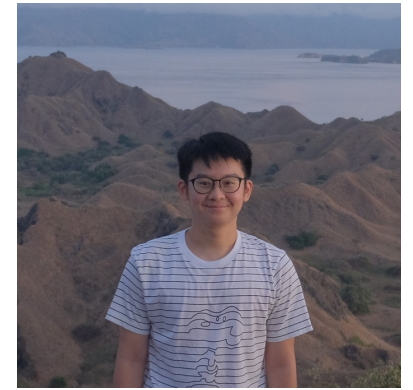
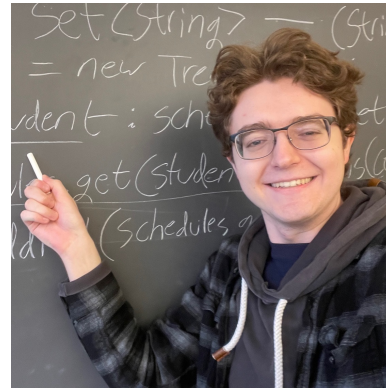
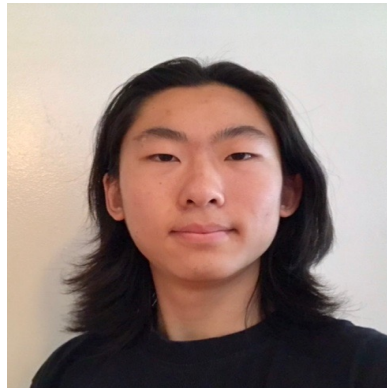
- Instructor: Melissa Lin
- Teaching Assistants:
 - Available in section, office hours, and discussion board
 - Invaluable source of information & help in this course
- We're excited to get to know you!
 - Our goal is to help you succeed 😊

About me

me ↓



Tas <3



Students

- 79 of you!
- Strength in numbers
 - If you're confused about something, I guarantee someone else is too!
 - Students come from all different backgrounds & majors & interests in future career goals
- Focus on us trying to help you build community
 - Meet others in the class to form study groups or have people you can work with.

What is this Class?

CSE 121 – Computer Programming I or Other Programming Experience

- Print statements
- Data types (int, String, boolean)
- Methods / Functions
 - Parameters
 - Returns
- Control structures
 - Loops
 - Conditionals
- File I/O
- Arrays
- **Computational Thinking**
(language agnostic)

CSE 122 – Computer Programming II

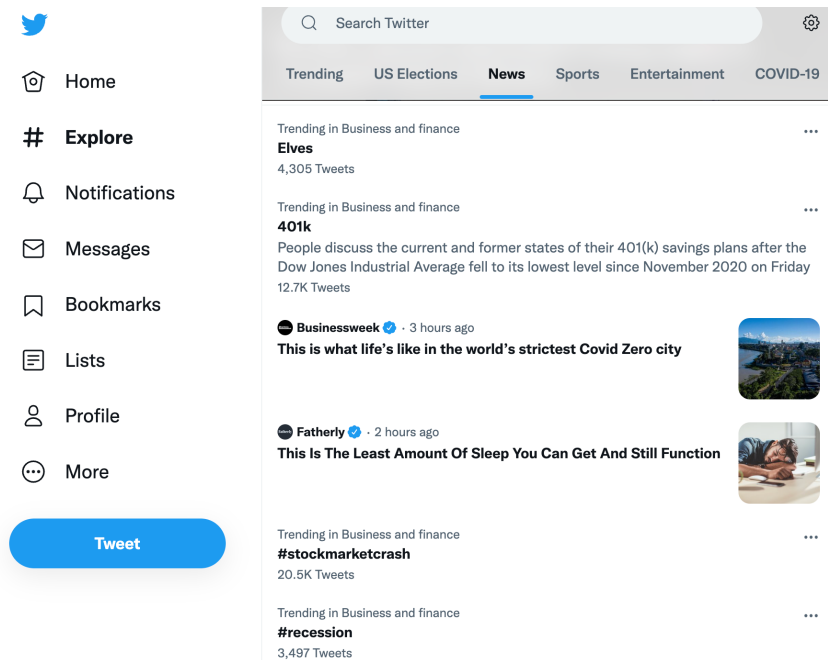
- Decomposing large problems into smaller, manageable, subproblems
- Using data structures
 - List
 - Stacks / Queues
 - Sets
 - Maps
- Object Oriented Programming
 - Interfaces

Prerequisite Knowledge

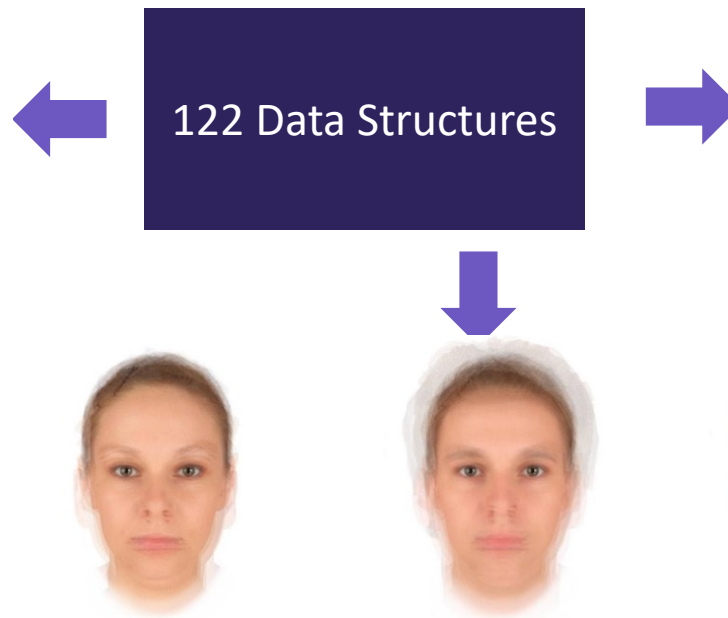
- Students entering CSE 122 are coming from many of different backgrounds
 - UW: CSE 121 or other intro programming course
 - Community College: Intro Programming Course
 - High School Programming Course (e.g., UWHS, AP CS, IB CS, etc.)
 - Self-taught or other previous experience
- Importantly: CSE 122 is in Java, but we **do not expect prior experience in Java!** Do expect knowing the list of CSE 121 topics in some language.
 - Students who do not have experience in Java will be focusing on practicing the programming skills you know in a new language!
 - You will find the [Java Tutorial](#) and Programming Assignment 0 very helpful!
- If you want to know if this class is the right fit for you, take the [Allen School Self-Placement Test](#)

Why 122?

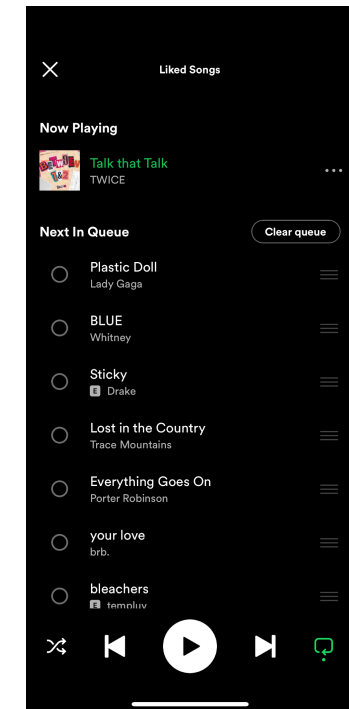
1. Build a strong foundation of data structures that will let you tackle the biggest problems in computing



Source: Twitter 9/23



Source: Ethical CS

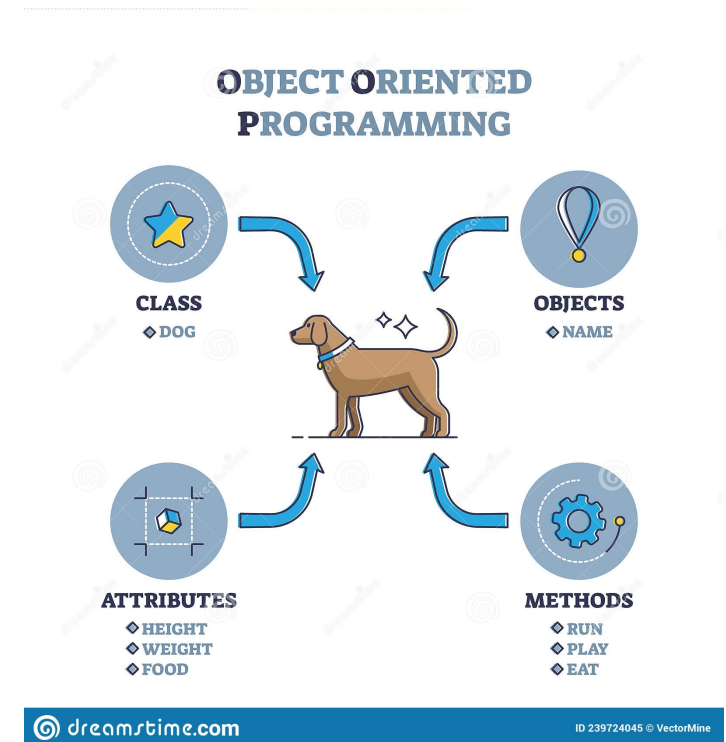


Source: Hunter's Spotify

Why 122?

2. Learn an important structural pattern for representing **objects** in code to make our code more **reusable** and **maintainable** and **easier to understand**.

- Java is designed around this idea of objects. We haven't been leveraging that yet!
- Used in almost every real-world software project.




Poll: Why are you taking this class?

- Answer on [sli.do #cse122](https://sli.do/#cse122) or using this QR code:



Lecture Outline

- Introductions
- **About this Course**
 - **Course Components & Tools** 
 - Policies
 - Making the Most of this Class
- Intro/Review Java

Course Components

Meetings

LECTURES

(x19)

- We're here!
- Introduce concepts, practice ideas, discuss applications.
- Pre-class materials to prepare for class each day. Due **before** class.

SECTIONS

(x18)

- Held in person
- More practice, reviews, applications
- TA advice, how to be an effective student
- Preparation for quizzes / exams

Assessments

PROGRAMMING ASSIGNMENTS

(x4)

- Structured assignments
- Programming in Java
- Applying & implementing course concepts

CREATIVE PROJECTS

(x4)

- More open-ended assignments
- Explore new ideas and applications

QUIZZES

(x3)

- Take home, open-note, open-internet
- ~45 minutes

EXAM

(x1)

- Culminating exam
- More details to come

Course Website

cs.uw.edu/122

The screenshot shows the course website for CSE 122. On the left is a navigation sidebar with links for Home / Calendar, Programming Assignments, Creative Projects, Exam, Staff, Office Hours, Syllabus, Grading Rubric, COVID-19 Safety, Resources, Course Tools, EdStem, and Anonymous Feedback. The main content area is titled "Introduction to Computer Programming II Summer 2023". It features a "Feedback" section, a "Welcome to CSE 122: Introduction to Computer Programming II" section with two expandable items: "What is this class? What will I learn?" and "Prior Experience and Expectations". Below this is a link to the course syllabus. A "This Week (at a glance)" section lists activities for Tuesday (06/20) and Wednesday (06/21), including a note that there is no section today on Tuesday and a class session on Wednesday at 10:50am in PCAR 192 with pre-class material.

The screenshot shows the instructor's bio page for Melissa Lin. It includes a profile picture, her name "Melissa Lin" with a pronoun indicator "(SHE/HER/HERS)", and her email "mlin23@uw". The bio text reads: "Hi everyone, I'm Melissa! I'm currently a BS/MS computer science student in the Allen School. Previously, I was a TA for 8 quarters (3 quarters for 122, 5 quarters on some 300 level CSE courses). This will be my first time being the instructor for a course, so I'm super excited for this quarter! I've spent my whole life in the Seattle area, from Redmond where I was born and raised, to Seattle when I started at UW as a direct admit to the Allen School. After doing some computer science tutoring and realizing how much I enjoyed teaching, I became a teaching assistant here and have been doing it ever since. Outside of school and teaching, I enjoy crocheting, bouldering, watching TV shows, reading, going on walks, and exploring Seattle. My favorite weekend activity is going to a cafe to read or do work, so please let me know your favorite cafes so I can try them! Looking forward to a great quarter with y'all!" The page also lists "Office Hours" as "Wednesday and Friday 12:00 - 1:00pm CSE 206". At the bottom, there is a section for "Teaching Assistants".

Get to know the staff

Contains most course info – check frequently!

- Announcements, Calendar, Lecture Slides, Office Hours schedule, Staff Bios, Important Links

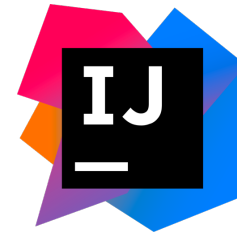
Other Course Tools



Ed

- Community & Information
 - Discussion Board
(please ask & answer!; anonymous option)
 - Chat
 - Announcements
- Pre-Class Materials / Section Handouts
- Assignments
 - Online IDE
 - Submit assignments
 - View Feedback

My Digital Hand



My Digital Hand

- Queueing in office hours

IntelliJ

- Develop offline
- Visual debugger



Canvas


- Gradebook
- Lecture recordings



Sli.do

- In-class activities
(ungraded)
- No account needed

Lecture Outline

- Introductions
- **About this Course**
 - Course Components & Tools
 - **Policies** 
 - Making the Most of this Class
- Intro/Review Java

Resubmissions

Learning is a challenging process that takes time, it doesn't always happen on your first try.

- Each week, one previous Programming Assignment or Creative Project can be resubmitted
 - Must be accompanied by a short write up explaining changes
 - Grade on resubmission completely replaces original grade.

See syllabus for more details

Collaboration

- These concepts are challenging: we strongly encourage discussion + collaboration!
 - Don't attempt to gain credit for something you didn't do
 - In general, share ideas and work together, but don't copy work. Never show someone else your code or solution write up.
 - For any ungraded work (e.g., pre-class materials) there is no concern about academic misconduct! You should be collaborating on those without reservation.
 - On graded assignments you should still collaborate, but the code you write should be of your own creation.
 - Always cite the help you receive on graded work
- [Withdrawal Policy](#)
- **Read full policy in Syllabus**

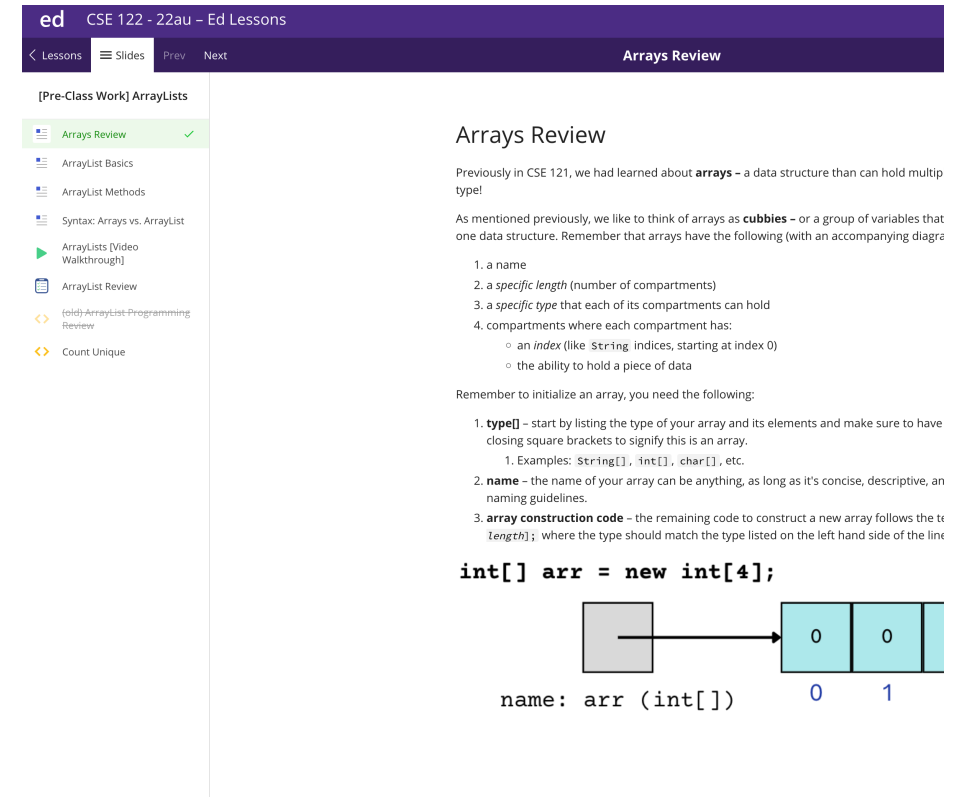
Textbook

Pre-class Materials

- All required readings are available free on Ed!
- Should be finished before class (not graded)

Optional Textbook

- [Building Java Programs by Reges and Stepp \(5th Edition\)](#)
- Not required but does add another perspective. Will reference relevant chapters.
- Advice: only purchase if you learn best with a textbook, otherwise not recommended.



ed CSE 122 - 22au - Ed Lessons

< Lessons Slides Prev Next

Arrays Review

[Pre-Class Work] ArrayLists

- ArrayLists Review ✓
- ArrayList Basics
- ArrayList Methods
- Syntax: Arrays vs. ArrayList
- ArrayLists [Video Walkthrough]
- ArrayList Review
- [old] ArrayList Programming Review
- Count Unique

Arrays Review

Previously in CSE 121, we had learned about **arrays** – a data structure that can hold multiple types!

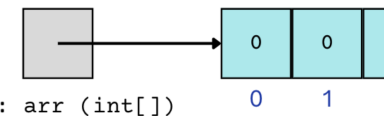
As mentioned previously, we like to think of arrays as **cubbies** – or a group of variables that are all of the same type. Remember that arrays have the following (with an accompanying diagram):

1. a **name**
2. a **specific length** (number of compartments)
3. a **specific type** that each of its compartments can hold
4. compartments where each compartment has:
 - an **index** (like `String` indices, starting at index 0)
 - the ability to hold a piece of data

Remember to initialize an array, you need the following:

1. **type[]** – start by listing the type of your array and its elements and make sure to have closing square brackets to signify this is an array.
 1. Examples: `String[]`, `int[]`, `char[]`, etc.
2. **name** – the name of your array can be anything, as long as it's concise, descriptive, and follows naming guidelines.
3. **array construction code** – the remaining code to construct a new array follows the type and `length`; where the type should match the type listed on the left hand side of the line.

```
int[] arr = new int[4];
```



name: arr (int[]) 0 1

Lecture Outline

- Introductions
- **About this Course**
 - Course Components & Tools
 - Policies
 - **Making the Most of this Class** ◀
- Intro/Review Java

How Learning Works

- Learning requires **active participation** in the process. It's not as simple as sitting and listening to someone talk at you.
 - Requires **deliberate practice** in **learning by doing**
 - Benefits from **collaborative learning**
- Hybrid classroom model
 - Asks you to do some preparation before class in the form of readings and practice problems.
 - Should take ~30 minutes a day
 - Class will start with brief recap, then pick up where the reading and practice problems leave off.
 - Attendance isn't graded, but showing up and trying is the first step in succeeding in the class!
- Pre-class materials are ungraded, but
 - It's okay if you find them challenging! That means you are learning!



Metacognition

- **Metacognition**: asking questions about your solution process.
- Examples:
 - **While debugging**: explain to yourself why you're making this change to your program.
 - **Before running your program**: make a prediction of what you expect to see.
 - **When coding**: be aware when you're not making progress, so you can take a break or try a different strategy.
 - **When designing**:
 - Explain the tradeoffs with using a different data structure or algorithm.
 - Reflect on how you ruled out alternative ideas along the way to a solution.
 - **When studying**: what is the relationship of this topic to other ideas in the course?
 - **After coding**: reflect on the process – what were the challenges?

Getting Help

- Discussion Board
 - Feel free to make a public or private post on Ed
 - We encourage you to answer other peoples' questions! A great way to learn
- Introductory Programming Lab (Office Hours)
 - TAs can help you face to face in office hours, and look at your code
 - You can go to the IPL with **any** course questions, not just assignments
- Section
 - Work through related problems, get to know your TA who is here to support you
- Your Peers
 - We encourage you to form study groups! Discord or Ed are great places to do that
- Email
 - We prefer that all content and logistic questions go on the Ed discussion board (even if you make them private).
 - For serious personal circumstances, you can email me directly. It never hurts to email me, but if it's a common logistic question, I may politely ask you to post on the discussion board.

Help Us Improve!

- This is a very new course! We are always looking for feedback on how to improve the class for you and for future students! Thank you in advance for your patience and understanding as we develop everything. 😊
 - We *really* value your feedback!
 - Let us know what's working and what isn't working for you
 - Something that went well in another course? Tell us about it!
- Post on the discussion board (can be public/private).
 - Note: Anonymous here is anonymous to other students, not to the staff.
- Submit feedback via the **Anonymous Feedback Tool** (linked under “Course Tools” on the website)

The World Around CSE 122

- Our goal is to give you a great CSE 122 experience
 - But CSE 122 does not exist in a vacuum – there's a lot going on in the world right now that can impact your education
- We've designed course policies for maximum flexibility
 - But we cannot cover every individual situation
- **Please reach out** if you need accommodations of any kind to deal with these unfamiliar situations

Lecture Outline

- Introductions
- About this Course
 - Course Components & Tools
 - Policies
 - Making the Most of this Class
- **Intro/Review Java** 

Hello World

- Java Specifics


- Every program needs a **class**
- Runnable programs need a **main** method (*signature* must exactly match)
- `System.out.println` to print
- **"Hello world"** is a `String`

- Running on Ed

- **Run** runs your program
- **Mark** submits and runs autograder
 - Submit as many times as you like
 - "Shotgun submission" = Unhelpful habit
- **Solution** shows solution (if applicable)

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello world");  
    }  
}
```

“Homework” for Next Time

- First assignment will be released Friday, but there are some things to do in the meantime.
- TODOs this week
 - Go meet your TA and classmates in Thursday’s quiz section
 -  Complete the pre-class material for Friday (see calendar)
 - Read over syllabus