LEC 01

CSE 373

Welcome! 👋

BEFORE WE START

***Use the Zoom chat*: Introduce yourself! What are you most excited about in this class? What's your least favorite vegetable?**

Instructor Aaron Johnston

TAs
Timothy Akintilo
Brian Chan
Joyce Elauria
Eric Fan
Farrell Fileas
Leona Kazi
Keanu Vestil
Howard Xiao
Siddharth Vaidyanathan

Lecture Outline

- Introductions
- About this Course
  - Course Components & Tools
  - Policies
  - Making the Most of this Class
- Abstract Data Types

Course Staff

- Instructor: Aaron Johnston
  - Grad student from UW CSE, previously taught CSE 333 (Systems Programming) and CSE 390B (Academic Skill-Building)
- Teaching Assistants:

  - Available in section, office hours, discussion board, and 1:1 meetings
  - Invaluable source of information & help in this course
- We’re excited to get to know you!
  - Our goal is to help you succeed

Students

- Currently 205 students registered for the course
  - Over double the size of last year's summer 373 offering (!)
- If you're waiting to register, unfortunately there are no overloads available, and the course staff does not have add codes
  - Reach out to `ugrad-advisor@cs.washington.edu` with any registration questions
- Strength in numbers
  - With 205 students, if you're confused about something, I guarantee someone else is too!
  - Students come from all different backgrounds & majors

What is this Class?

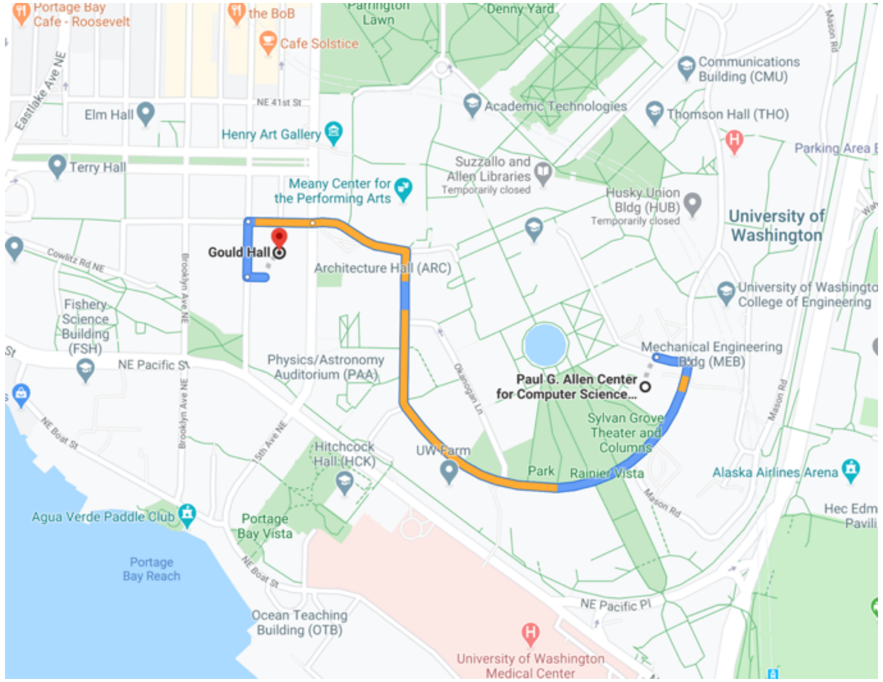
- **CSE 143 – Object Oriented Programming**
  - Classes and Interfaces
  - Methods, variables and conditionals
  - Loops and recursion
  - Linked lists and binary trees
  - Sorting and Searching
  - O(n) analysis
  - Generics

- **CSE 373 – Data Structures and Algorithms**
  - Design decisions
  - Design analysis
  - Implementations of data structures
  - Debugging and testing
  - Abstract Data Types
  - Code Modeling
  - Complexity Analysis
  - Software Engineering Practices

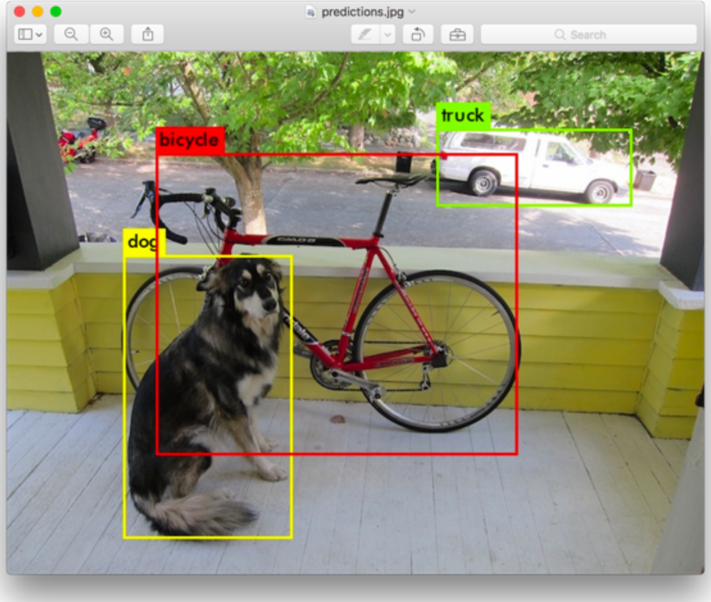
Why 373?

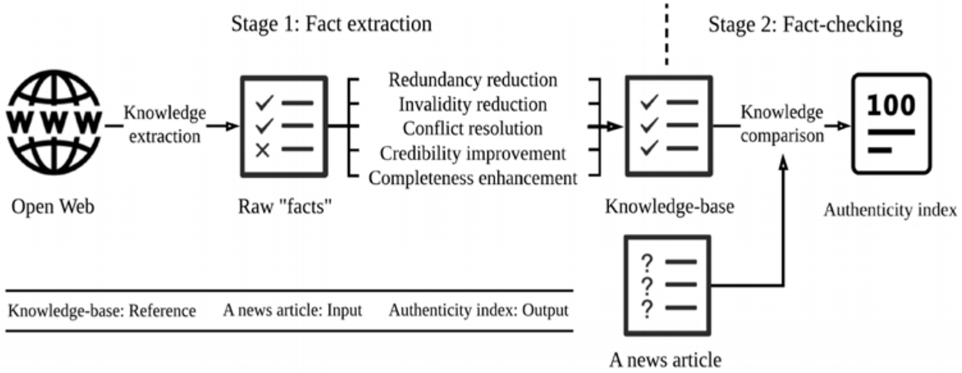
1. Build a strong foundation of data structures and algorithms that will let you tackle the biggest problems in computing

373 Data Structures & Algorithms



Portage Bay Cafe - Roosevelt
the BoB
Cafe Solstice
Denny Yard
Communications Building (CMU)
Academic Technologies
Thomson Hall (THO)
Elm Hall
Henry Art Gallery
Terry Hall
Suzzallo and Allen Libraries
Meany Center for the Performing Arts
Husky Union Bldg (HUB)
University of Washington
Gould Hall
Architecture Hall (ARC)
University of Washington College of Engineering
Fishery Science Building (FSH)
NE Pacific St
Physics/Astronomy Auditorium (PAA)
Mechanical Engineering Bldg (MEB)
Paul G. Allen Center for Computer Science...
Sylvan Grove Theater and Columns
Hitchcock Hall (HCK)
UW Farm
Rainier Vista
Alaska Airlines Arena
Agua Verde Paddle Club
Portage Bay Vista
Portage Bay Reach
Ocean Teaching Building (OTB)
NE Pacific Pl
University of Washington Medical Center





Fake News: A Survey of Research, Detection Methods, and Opportunities (Xinyi Zhou, Reza Zafarani/arXiv:1812.00315)

Why 373?

2. Pick up the vocabulary, skills, and practice needed to make **design decisions**. Learn to **evaluate** the tools in your CS toolbox

- Differences between technical implementations
- Evaluation can mean many different things!

Lecture Outline

- Introductions
- About this Course
  - Course Components & Tools
  - Policies
  - Making the Most of this Class
- Abstract Data Types

Course Components

LECTURES (x26)

- Held live via Zoom
- Recordings available after
- In-lecture activities
- Introduction to course concepts

SECTIONS (x9)

- Held live via Zoom
- Review videos available after
- Practice problems, reviews, TA advice
- Preparation for exams

PROJECTS (x5)

- Partner recommended
- Programming in Java
- Applying & implementing course concepts
- More practical

EXERCISES (x4)

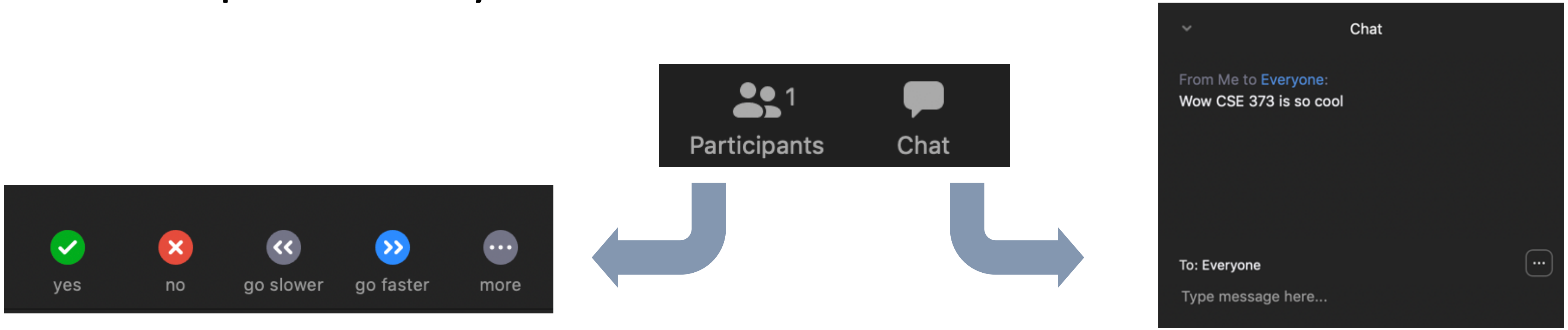
- Individual
- Written problems, focusing on the "why"
- More conceptual

EXAMS (x2)

- Available over a multi-day window, complete whenever works for you
- More details as we get closer

Using Zoom

- Two important ways to interact in lecture:



Chat

From Me to Everyone:
Wow CSE 373 is so cool

To: Everyone
Type message here...

- Open `Participants` Pane
  - Use the feedback buttons for quick cues
- Open `Chat` Pane
  - Type your questions in the chat
    - Other students, TAs, or I can answer!
  - Please conduct yourself as you would in a classroom – be respectful!

- Sign in early to chat with other students, warm up for the day

Poll Everywhere

pollev.com/uwcse373

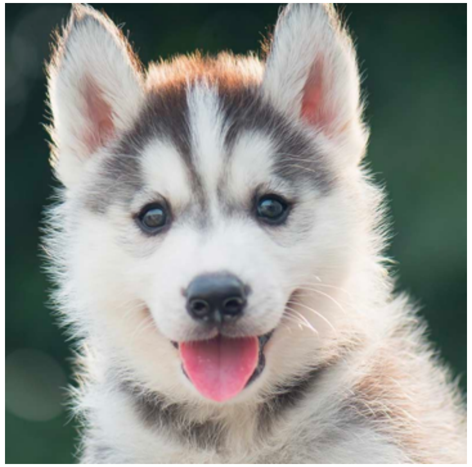
Using PollEverywhere

- Sometimes I'll ask for more involved feedback or we'll pause to do an active learning activity
- Go to `pollev.com/uwcse373` to register and participate
- Let's practice: which puppy is cutest?

A

B

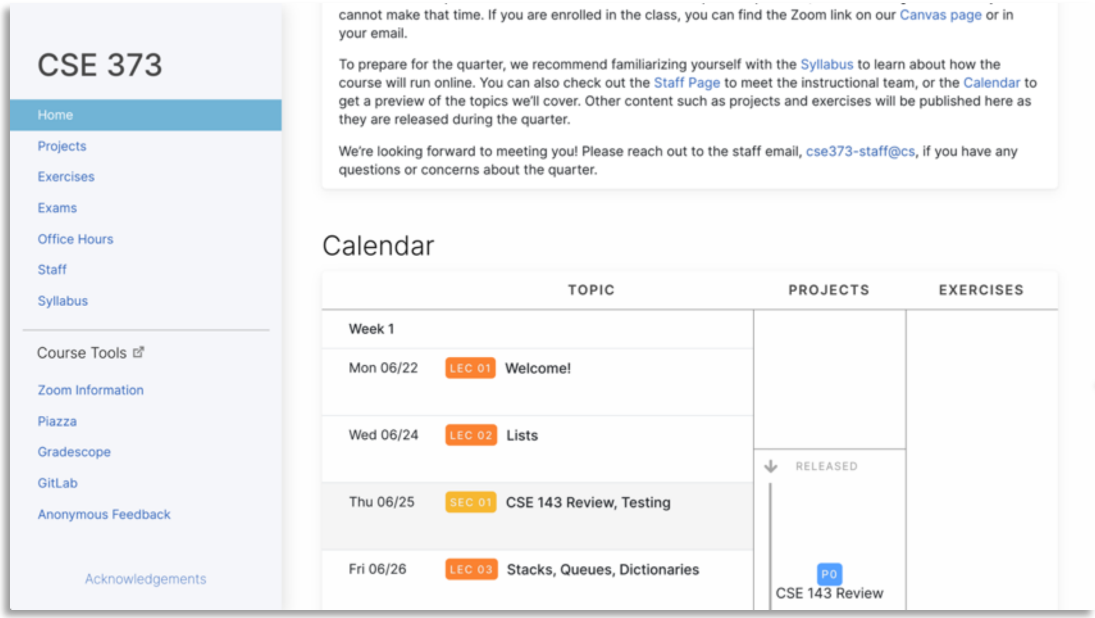
C



D

Course Website

cs.uw.edu/373



cannot make that time. If you are enrolled in the class, you can find the Zoom link on our Canvas page or in your email.

To prepare for the quarter, we recommend familiarizing yourself with the Syllabus to learn about how the course will run online. You can also check out the Staff Page to meet the instructional team, or the Calendar to get a preview of the topics we'll cover. Other content such as projects and exercises will be published here as they are released during the quarter.

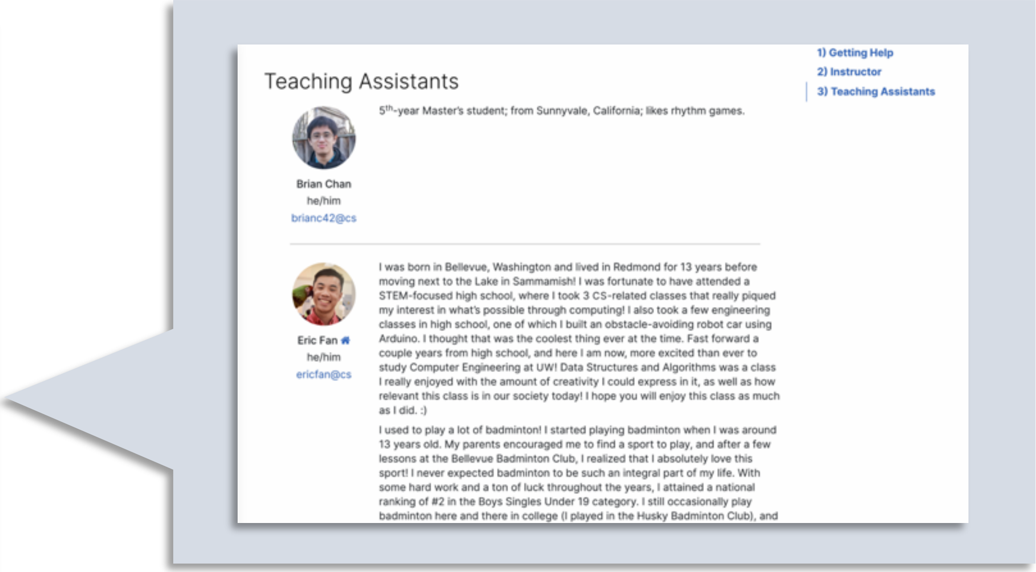
We're looking forward to meeting you! Please reach out to the staff email, cse373-staff@cs, if you have any questions or concerns about the quarter.

Calendar

	TOPIC	PROJECTS	EXERCISES
Week 1			
Mon 06/22	LEC 01 Welcome!		
Wed 06/24	LEC 02 Lists		
Thu 06/25	SEC 01 CSE 143 Review, Testing	↓ RELEASED	
Fri 06/26	LEC 03 Stacks, Queues, Dictionaries	P0 CSE 143 Review	

Contains most course info – check frequently!

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



Teaching Assistants

Brian Chan
he/him
brianc42@cs

5th-year Master's student; from Sunnyvale, California; likes rhythm games.

Eric Fan
he/him
ericfan@cs

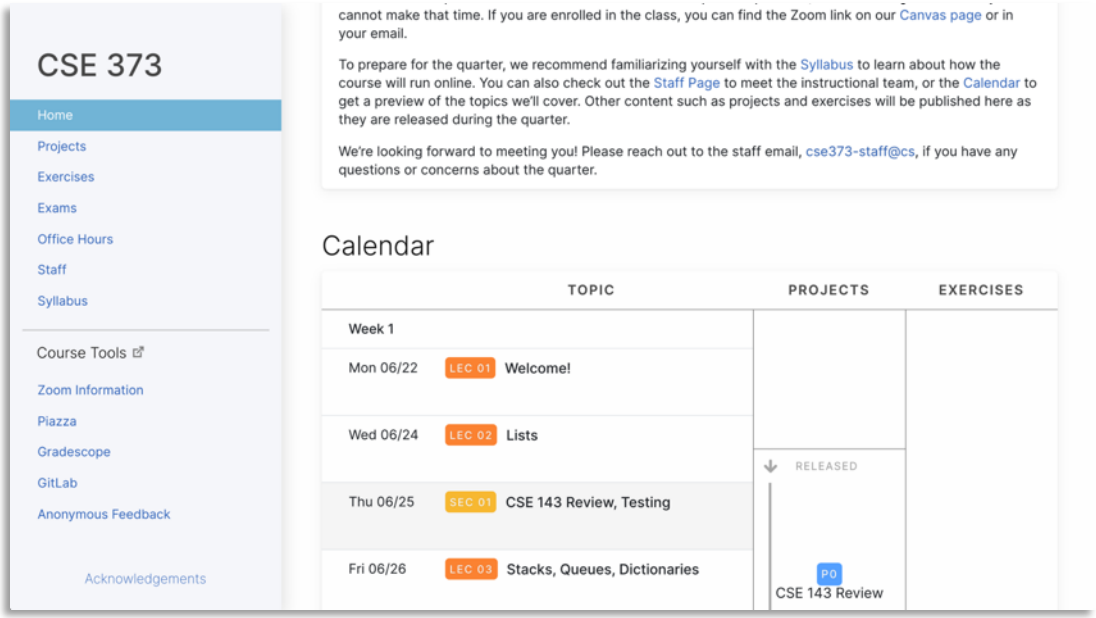
I was born in Bellevue, Washington and lived in Redmond for 13 years before moving next to the Lake in Sammamish! I was fortunate to have attended a STEM-focused high school, where I took 3 CS-related classes that really piqued my interest in what's possible through computing! I also took a few engineering classes in high school, one of which I built an obstacle-avoiding robot car using Arduino. I thought that was the coolest thing ever at the time. Fast forward a couple years from high school, and here I am now, more excited than ever to study Computer Engineering at UW! Data Structures and Algorithms was a class I really enjoyed with the amount of creativity I could express in it, as well as how relevant this class is in our society today! I hope you will enjoy this class as much as I did. :)

I used to play a lot of badminton! I started playing badminton when I was around 13 years old. My parents encouraged me to find a sport to play, and after a few lessons at the Bellevue Badminton Club, I realized that I absolutely love this sport! I never expected badminton to be such an integral part of my life. With some hard work and a ton of luck throughout the years, I attained a national ranking of #2 in the Boys Singles Under 19 category. I still occasionally play badminton here and there in college (I played in the Husky Badminton Club), and

Get to know the staff

Course Website

cs.uw.edu/373



CSE 373

Home
Projects
Exercises
Exams
Office Hours
Staff
Syllabus

Course Tools

Zoom Information
Piazza
Gradescope
GitLab
Anonymous Feedback

Acknowledgements

cannot make that time. If you are enrolled in the class, you can find the Zoom link on our Canvas page or in your email.

To prepare for the quarter, we recommend familiarizing yourself with the Syllabus to learn about how the course will run online. You can also check out the Staff Page to meet the instructional team, or the Calendar to get a preview of the topics we'll cover. Other content such as projects and exercises will be published here as they are released during the quarter.

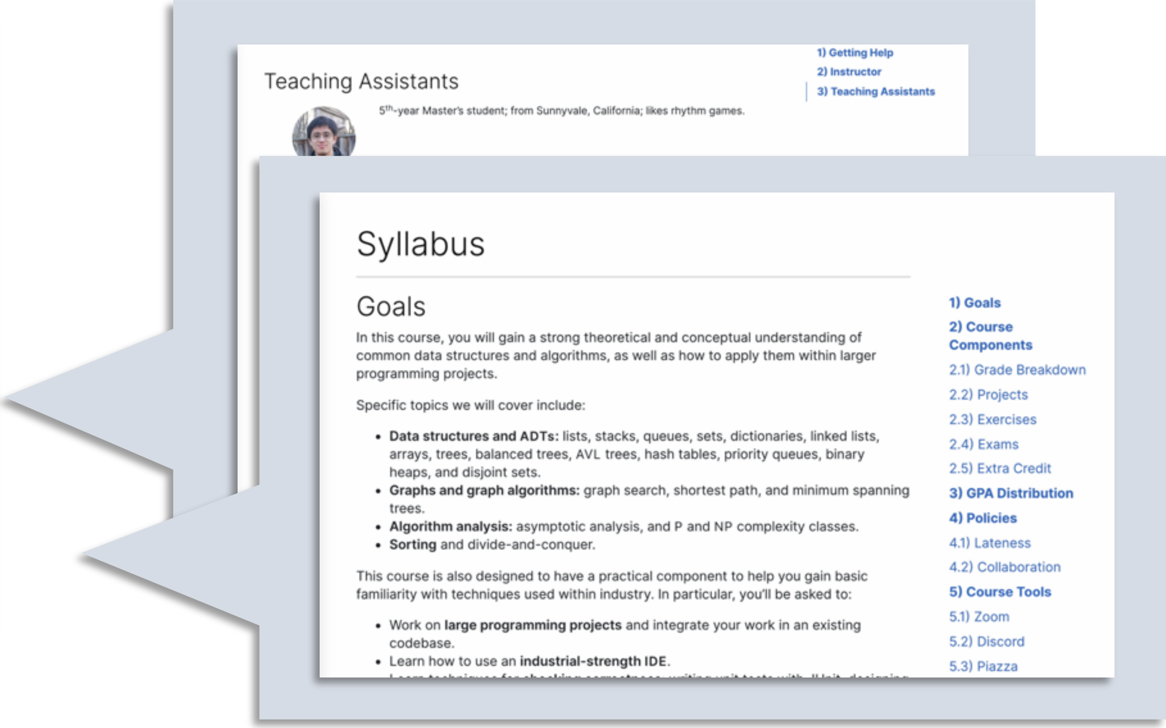
We're looking forward to meeting you! Please reach out to the staff email, cse373-staff@cs, if you have any questions or concerns about the quarter.

Calendar

TOPIC	PROJECTS	EXERCISES
Week 1		
Mon 06/22 LEC 01 Welcome!		
Wed 06/24 LEC 02 Lists	RELEASED	
Thu 06/25 SEC 01 CSE 143 Review, Testing		
Fri 06/26 LEC 03 Stacks, Queues, Dictionaries	P0 CSE 143 Review	

Contains most course info – check frequently!

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



Teaching Assistants

5th-year Master's student; from Sunnyvale, California; likes rhythm games.

1) Getting Help
2) Instructor
3) Teaching Assistants

Syllabus

Goals

In this course, you will gain a strong theoretical and conceptual understanding of common data structures and algorithms, as well as how to apply them within larger programming projects.

Specific topics we will cover include:

- **Data structures and ADTs:** lists, stacks, queues, sets, dictionaries, linked lists, arrays, trees, balanced trees, AVL trees, hash tables, priority queues, binary heaps, and disjoint sets.
- **Graphs and graph algorithms:** graph search, shortest path, and minimum spanning trees.
- **Algorithm analysis:** asymptotic analysis, and P and NP complexity classes.
- **Sorting** and divide-and-conquer.

This course is also designed to have a practical component to help you gain basic familiarity with techniques used within industry. In particular, you'll be asked to:

- Work on **large programming projects** and integrate your work in an existing codebase.
- Learn how to use an **industrial-strength IDE**.

1) Goals
2) Course Components
2.1) Grade Breakdown
2.2) Projects
2.3) Exercises
2.4) Exams
2.5) Extra Credit
3) GPA Distribution
4) Policies
4.1) Lateness
4.2) Collaboration
5) Course Tools
5.1) Zoom
5.2) Discord
5.3) Piazza

Please familiarize yourself with the course syllabus this week!

Other Course Tools

Piazza

- Discussion Board & Announcements
- Please ask AND answer!
- Anonymous option
- Opt out of Piazza Network

Discord

- Community: meet other students, form study groups
- Most Office Hours held here
- *More details to come*

Gitlab

- Everyone gets a git repo
- We'll distribute starter code, you'll push your work
- *More details to come*

Gradescope

- Submit all your assignments
- Get feedback

Canvas

- Only used for Zoom recordings and gradebook

Lecture Outline

- Introductions
- **About this Course**
  - Course Components & Tools
  - **Policies**
  - Making the Most of this Class
- Abstract Data Types

Grading Breakdown

- Your grade will consist of the following weighted categories:

Category	Weight
Programming Projects	45%
Individual Exercises	25%
Exam I	15%
Exam II	15%

- Instead of curving the class as usual, we'll use a bucket system:
  - These are *minimum* GPA guarantees – may adjust upward

Percentage	GPA
95%	4.0
90%	3.5
80%	3.0
60%	2.0
50%	0.7

Assignment Policies

Collaboration & Academic Integrity

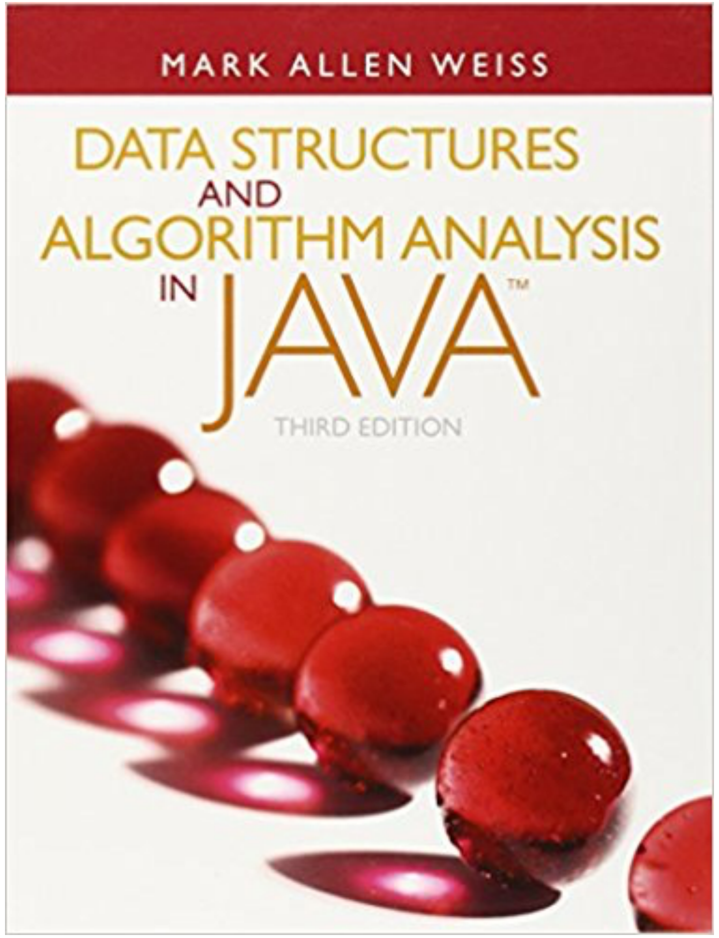
- These concepts are hard: we strongly encourage discussion + collaboration!
  - Don’t attempt to gain credit for something you didn’t do
  - In general, share ideas and work together, but don’t copy work. Never show someone else your code or solution write up.
  - Always cite the help you receive
  - Full collaboration with your partner on projects!
- **Read full policy in Syllabus**

Lateness

- You get 7 “free” late days for the quarter – submit 24 hours late with no penalty
  - Use on projects or exercises
- After that, -5% each day late
- No assignment can be submitted >72 hours late
  - Except with instructor permission

Textbook

- Data Structures and Algorithm Analysis in Java by Mark Allen Weiss
- Completely **optional**
  - Nothing assigned out of the textbook
  - No readings
- Advice: only purchase if you learn best with a textbook, otherwise not recommended



Lecture Outline

- Introductions
- **About this Course**
  - Course Components & Tools
  - Policies
  - **Making the Most of this Class**
- Abstract Data Types

Getting Help

- Discussion Board
  - Feel free to make a public or private post on Piazza
  - We encourage you to answer other peoples' questions! A great way to learn
- Office Hours
  - TAs can help you face to face in office hours, and look at your code
  - Discord gives great flexibility – feel free to join your peers in the OH chat and listen in to see who else has a similar problem
- 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 Piazza are great places to do that
- Email the Staff List
  - You can always email cse373-staff@cs.washington.edu if you don't know how to get help – we'll work to get you the support you need

Help Us Improve!

- We’re still learning how to do this online ☺
  - Thank you in advance for your patience and understanding
  - 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!

- Email the course staff at `cse373-staff@cs.uw.edu`

- Submit feedback via the **Anonymous Feedback Tool** (linked under “Course Tools” on the website)

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 an explicit 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.
    - If one or more requirements change, how would the solution change as a result?
    - 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?

The World Around 373

- Our goal is to give you a great 373 experience
  - But CSE 373 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: plenty of late days, take-home exams, no participation
  - 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
- **Abstract Data Types**

Data Structures & Algorithms

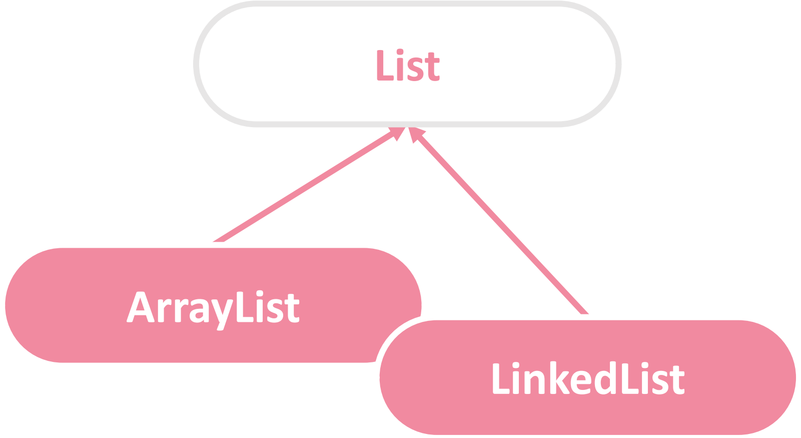
- Data Structure:
  - A way of organizing, storing, accessing, and updating data
  - **Examples from CSE 14x**: Arrays, Linked Lists, Stacks, Queues, Trees

- Algorithm:
  - A series of precise instructions to produce a specific outcome
  - **Examples from CSE 14x**: Binary Search, Merge Sort, Recursive Backtracking

Review Interface vs. Implementation

- In Java, an **interface** is a data type that specifies what to do but not how to do it.
  - **List**: an ordered sequence of elements.

- A **subtype** implements all methods required by the interface.
  - **ArrayList**: Resizable array implementation of the List interface.
  - **LinkedList**: Doubly-linked implementation of the List interface.

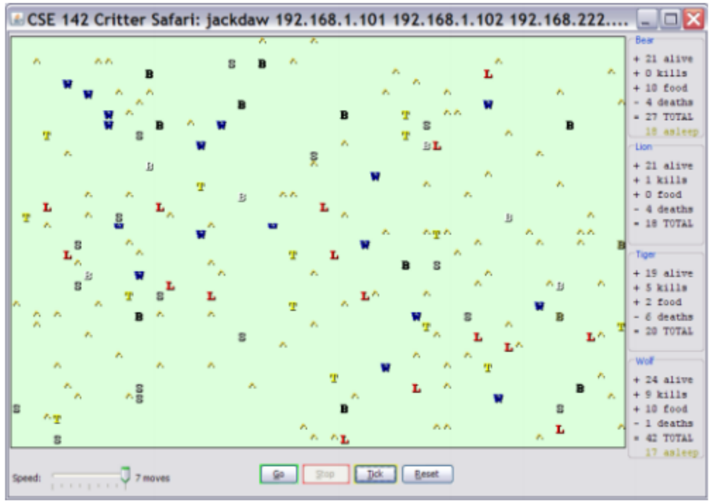


Review Client vs. Object

Client Classes

- A class that is executable, in Java: contains a Main method

```
public static void main(String[] args)
```

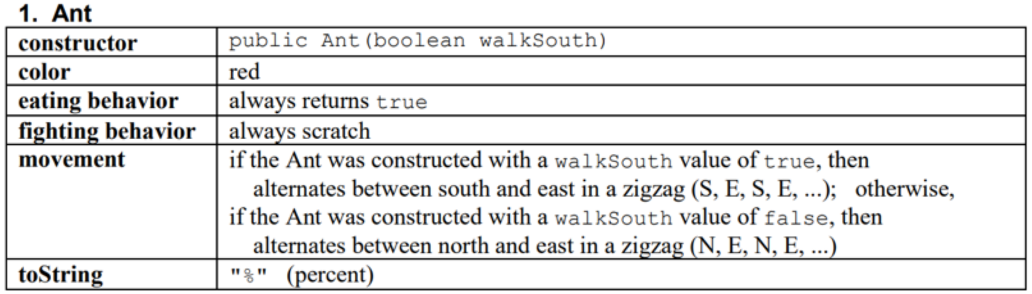


CSE 142 Critter Safari: jackdaw 192.168.1.101 192.168.1.102 192.168.222....

```
Bear
+ 21 alive
+ 0 kills
+ 10 food
- 4 deaths
= 27 TOTAL
  18 asleep
Lion
+ 21 alive
+ 1 kills
+ 0 food
- 4 deaths
= 18 TOTAL
Tiger
+ 19 alive
+ 5 kills
+ 2 food
- 6 deaths
= 20 TOTAL
Wolf
+ 24 alive
+ 9 kills
+ 10 food
- 1 deaths
= 42 TOTAL
  17 asleep
```

Object Classes

- A coded structure that contains data and behavior
- Start with the data you want to hold, organize the things you want to enable users to do with that data



1. Ant

constructor	`public Ant(boolean walkSouth)`
color	red
eating behavior	always returns `true`
fighting behavior	always scratch
movement	if the Ant was constructed with a `walkSouth` value of `true`, then alternates between south and east in a zigzag (S, E, S, E, ...); otherwise, if the Ant was constructed with a `walkSouth` value of `false`, then alternates between north and east in a zigzag (N, E, N, E, ...)
toString	`"%"` (percent)

ADTs: Abstract Data Types

- Java interfaces represent the concept of abstract data types.
- An **abstract data type** is a data type that does not specify any one implementation.
- **Data structures** implement ADTs.
  - **Resizable array** can implement List, Stack, Queue, Deque, PQ, etc.
  - **Linked nodes** can implement List, Stack, Queue, Deque, PQ, etc.

List ADT

A collection storing an ordered sequence of elements.

- Each element is accessible by a zero-based index.
- A list has a size defined as the number of elements in the list.
- Elements can be added to the front, back, or any index in the list.
- Optionally, elements can be removed.

Where we're Headed: ADTs we'll look at

- List
- Set
- Map
- Stack
- Queue
- Priority Queue
- Graph
- Disjoint Set

Learning to Bake in a CSE Class

- Think of what you'll learn this quarter as a cookbook
  - ADTs are the chapters/category: Soups, Salads, Cookies, Cakes, etc
    - High-level descriptions of a category of functionality
    - You don't serve a soup when guests expect a cookie!
  - Data structures are the recipes: chocolate chip cookies, snickerdoodles, etc
    - Step-by-step, concrete descriptions of an item with specific characteristics
    - Understand your tradeoffs before replacing carrot cake with a wedding cake
- When you go out into the world …
  - Figure out which category is required
  - Choose the specific recipe that best fit the situation