CS 2112 / ENGRD 2112 Object-Oriented Design and Data Structures — Honors

Fall 2022

Cornell University

CORNELL UNIVERSITY FOUNDED A.D. 1865

Course staff

Instructor: Nate Foster

Course Staff:

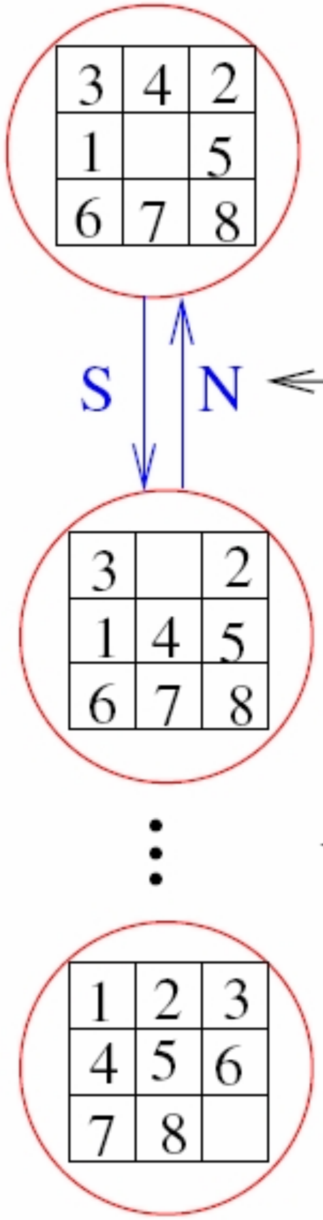
- Omkar Bhalerao
- Charles Sherk
- Jerry Xu
- Esther Wang
- Stanley Jiang
- Shiyuan Huang
- William Wang
- Vivian Ding
- Kevin Cui
- Arnav Parashar
- Jessica Cho
- Noah Schiff
- Kushal Kedia

What it's about

Introduction to computer science and software engineering

- **Programming language features**
  - data abstraction, subtyping, generic programming
  - concurrency and threads
  - Not a course *about* Java, but you will become comfortable with Java
- **Object-oriented design — organizing large programs**
  - specifications
  - design patterns
  - frameworks and event-driven programming
- **Data structures and algorithms**
  - recursive algorithms and data structures
  - reasoning about algorithm correctness and efficiency
    - induction, asymptotic complexity
  - arrays, lists, stacks, queues, trees, graphs, hash tables, and associated algorithms

Sam Loyd's 8 Puzzle



3	4	2
1		5
6	7	8

Initially scrambled configuration

Transition

(N/S/E/W means tile moves North/South/East/West)

3		2
1	4	5
6	7	8

Sequence of moves

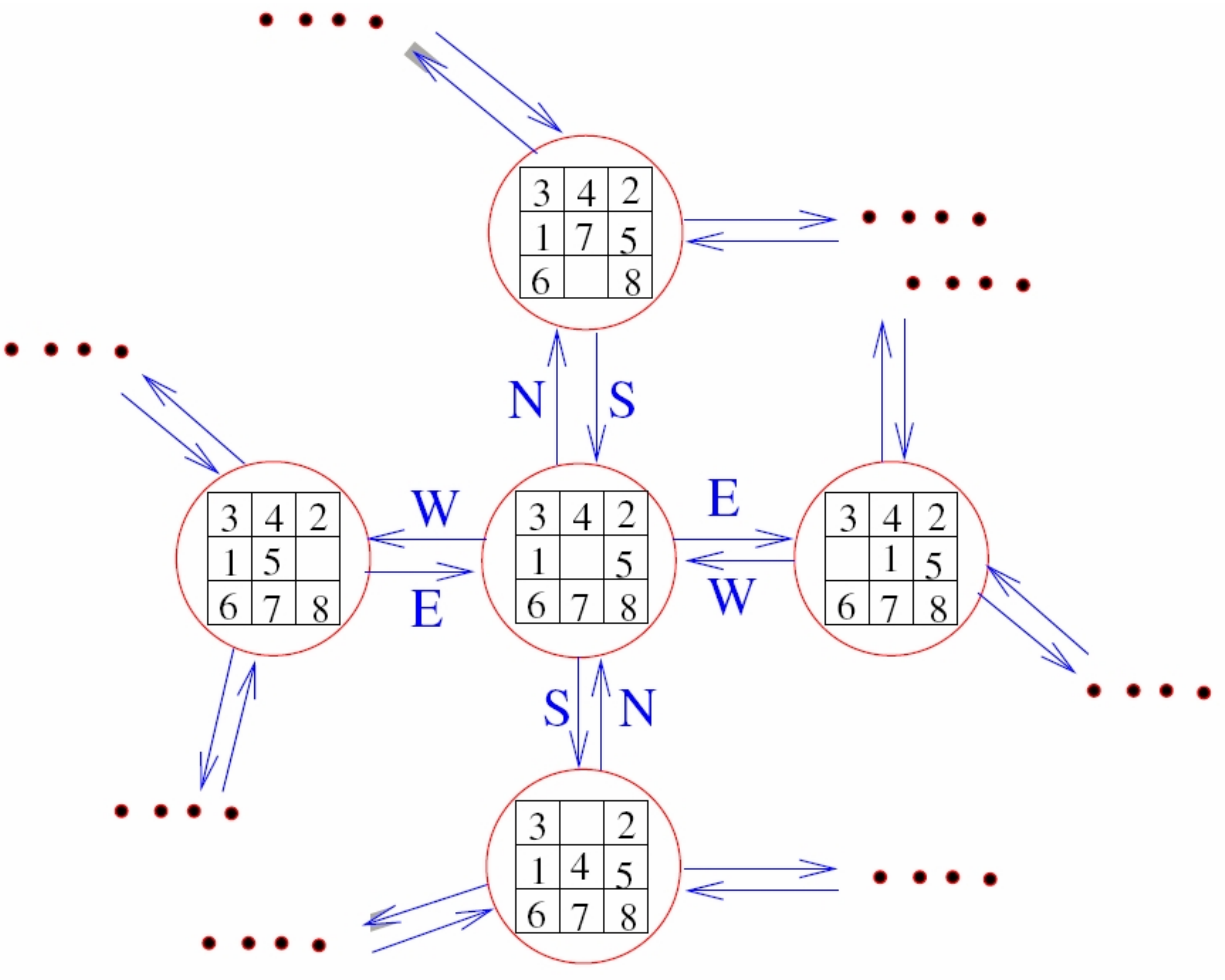
1	2	3
4	5	6
7	8	

Sorted configuration

Goal: Given an initial configuration of tiles, find a sequence of moves that will lead to the sorted configuration.

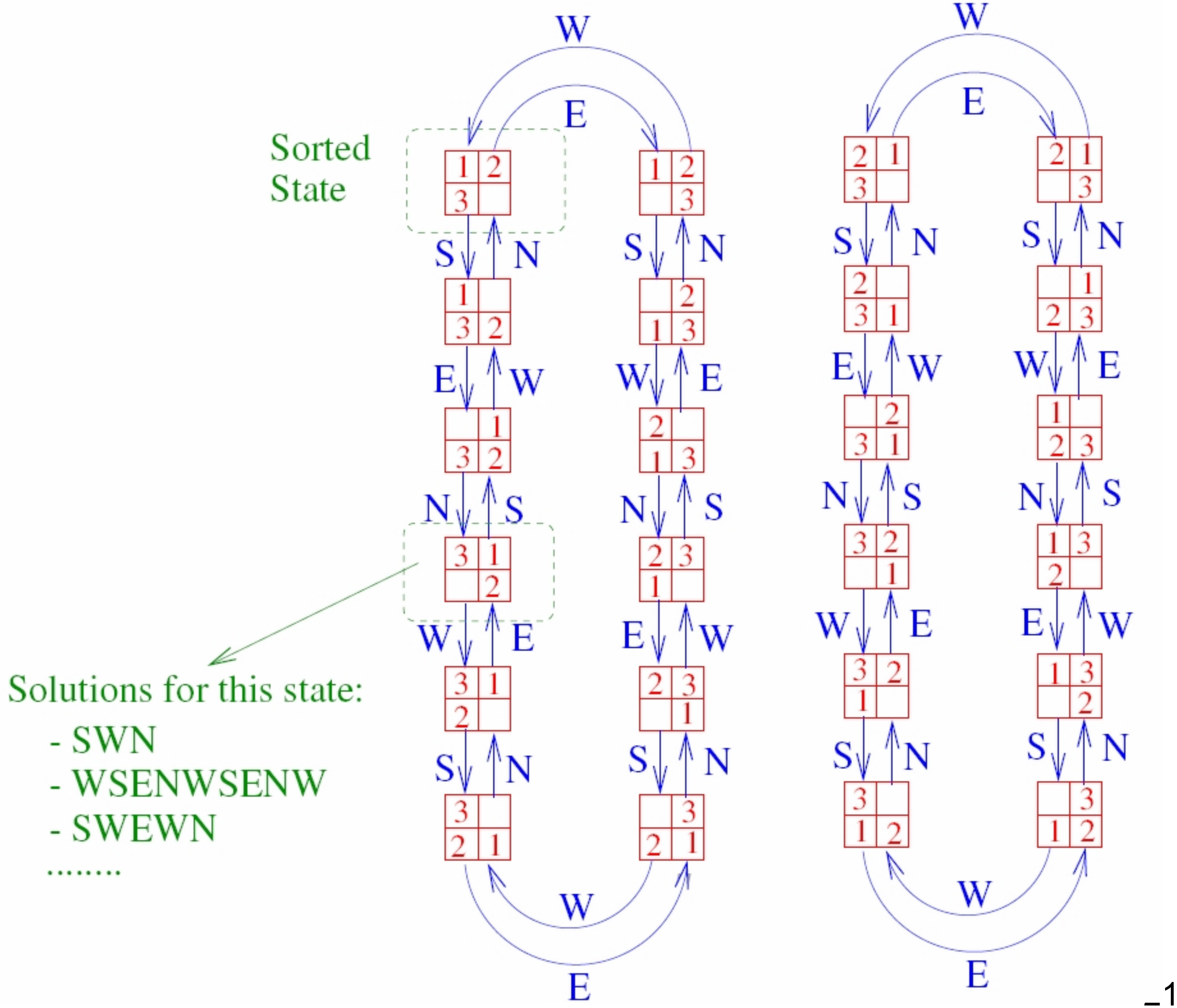
A particular configuration is called a ***state*** of the puzzle.

State Transition Diagram of 8-Puzzle



State Transition Diagram: picture of adjacent states.
A state Y is ***adjacent*** to state X if Y can be reached from X in one move.

State Transition Diagram for a 2x2 Puzzle



Sorted State

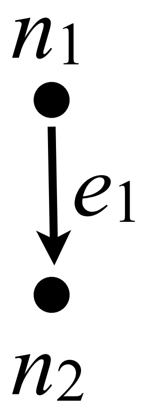
Solutions for this state:

- SWN
- WSENWSENW
- SWEWN

....

Graphs

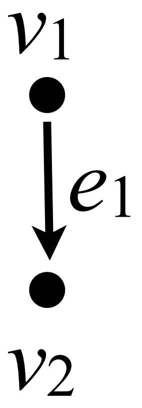
- State transition diagram in previous slide is an an example of a graph: a mathematical abstraction
    - –nodes (or vertices) : the puzzle states
    - –edges (or arcs) : the transitions, possibly labeled



- Graphs are all around us: Internet paths, airline routes, roadmaps, org charts, pipelines, family trees, ...

Graph algorithms

- Large toolbox of efficient algorithms for graphs help us solve problems:
  - searching for best nodes/shortest paths
  - finding maximum flow through graph
  - minimum spanning trees
  - . . .



- And known hardness results (e.g., finding Hamiltonian cycles) tell you what you can’t solve—at least, not efficiently.

Software design choices

- What operations should puzzle objects have?
- How do we represent states? The initial state? Transitions between states?
- How do we present information to the user and support interaction?
- How do we break the coding up into parts that can be coded independently?
- How to structure code so it can be maintained and upgraded in the future?

Why you need CS 2112

- Data structures and algorithms to solve problems efficiently and effectively
- Design techniques to produce code that works quickly and keeps working
- Computer science:
  - algorithms, data structures, programming languages, design principles, knowledge of what is possible and feasible
- Good programmers have more fun!
  - 10x more productive
  - better able to adapt, see opportunities, grow
  - change the world!

CS 2112 or ENGRD 2112?

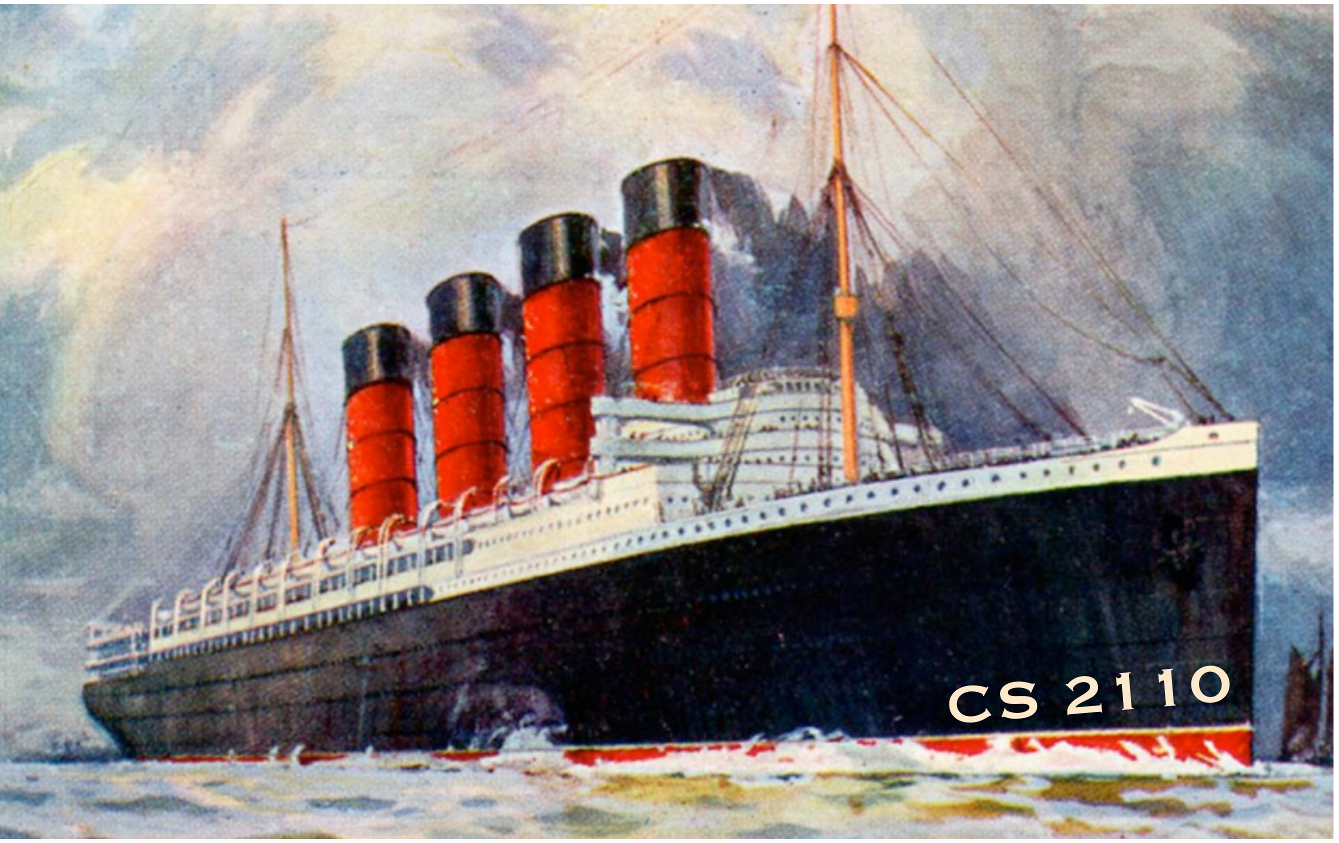
Doesn’t matter

CS/ENGRD 2110 or CS/ENGRD 2112?

- 2112 is an honors version of 2110
  - –aimed at CS majors
  - –much smaller (<100 vs. ~700)
  - –more material
    - more algorithms and their analysis (theory)
    - more about design and design patterns (practice)
  - –more difficult and more interesting assignments, with more programming and building code from scratch.
  - –cool final project spanning 3 assignments and a final tournament.
  - –more credits (4 vs 3)

2110 vs 2112

Warning: you will be challenged here



It's fine to switch between 2110 and 2112—in either direction—during the first three weeks.

Web site

- Your best source for information:

  `http://courses.cs.cornell.edu/cs2112`

  - Lecture notes: you are expected to read
    - may not include *everything* covered in lecture
    - may include extra material *not* covered in lecture
    - often updated after the lecture based on questions
  - Assignments
    - may occasionally be updated (w/ datestamp and changelog) after initial release
  - Pointers to resources

Communicating with staff

- Best: Ed
  - Answering other questions (well) counts as participation
  - Watch out for violations of academic integrity!
- Course announcements ☛ Ed (email if urgent)
- See website for office hours
  - Front line for answering questions—consulting hours start next week

CMS(X)

- `https://cmsx.cs.cornell.edu/`
- Assignments ☛ CMS
- Grades, solutions ☛ CMS
- Regrade requests ☛ CMS soon after receiving grade (remind us if necessary...)
- You should be registered if you are here

Meetings

- **Lectures**: TR 10:10–11am, Gates G01
- **Discussions** (attend **one** per week, as assigned)
  - –Tuesdays 12:25-1:15pm, Olin Hall 165
  - –Wednesdays 1:30-2:20pm, Hollister Hall 110
- **Labs** (attend **one** per week)
  - –Mondays 7:30-8:20pm, Thurston Hall 205
  - –Wednesdays 7:30-8:20pm, Olin Hall 165
- Starts *next* week
- Attendance is expected in the section you are registered for (confusing if you move around)

Assignments

- 6 assignments
  - mostly programming but some written problems
  - 50% of total score
- First assignment done alone
- Second and third alone or with one partner
- Final project (last 3 assignments) with one or two partners
- Late submissions: 10% penalty per day

Exams

- 2 x take-home exams (one due 9/29, the other on a registrar-assigned date during finals)
- 24-hours for each exam in a ~2 day window
- Open book, open notes
- Solved individually
- Worth 50% of total score, with higher score weighted slightly higher

Labs

- Programming exercises, solve problems, learn about tools
- Starts next week (IntelliJ demo and working with Java I/O)

Optional Textbook

Data Structures and Abstractions with Java, 4th ed., Frank M. Carrano and Timothy Henry, Pearson Education, 2014

- Recommended, not required
- Not heavily used
- Most helpful for extra details about implementations of complex data structures
- Earlier editions are fine

Software

- Java 11 (current LTS version of Java)
- IntelliJ IDEA

CS 2110 Java HyperText

- http://www.cs.cornell.edu/courses/JavaAndDS/
- David Gries's online text for OO programming in Java and data structures
- excellent resource, especially for students with little Java experience
- searchable glossary of terms
- information on Java, Eclipse, data structures, code style, program correctness, recursion, …
- short tutorials, pdf files, videos

Academic integrity

- You must never misrepresent someone else's work as your own or let others misrepresent your work as theirs
  - Copying code or answers is never okay
  - Aiding others' AI violations is also a violation
    ⇒ Letting others copy you is also a violation
  - You must be able to explain your answers fully
  - Discussions with others are perfectly fine if they could have happened in a lightless room
- We use highly effective tools for detecting plagiarism
- Report any discussions about assignments and any use of external code or tool, besides basic editors
- Our goal: spend time on course content

Social integrity

Everyone is to be treated with **respect**, regardless of background, experience, religion, ethnicity, citizenship, gender identity, or sexual orientation.

If you are made to feel unwelcome or disrespected, please contact me.

If you become aware of anyone else being made to feel unwelcome or disrespected, it is good to speak up! Also encourage them to contact course staff.

Covid-19

- If you have to stay home due to a positive Covid-19 test, please let me know, but otherwise just follow along with the materials on the course website.
- The lecture notes are very well done, and the canonical version of course content.
- As CS 2112 exams are take-home, we hope that positive tests with mild symptoms (the common case) shouldn't be a problem.
- Of course, we can make accommodations if you turn out to have a severe case and/or feel so unwell that you are unable to work.

Next steps

- Log in to Ed and say hi!
- Keep an eye on the 2112 website
- Download the first programming assignment when it is released (soon!)
- Get IntelliJ downloaded and working — can get in touch with course staff
- Attend lab next M/W if you need help
- Have fun!