Lecture 1: Introduction

CS 2110

August 26, 2025

Matt Eichhorn

- PhD (Applied Math) at Cornell
- BS (Computer Science, Math)

Teaching:
- CS 2110, CS 2800, ENGRI 1101

Research:
- Algorithm Design: optimization with unknown / random inputs
- Statistics on networks: how do interventions cascade through populations

Interests:
- Origami, crosswords, board games, Buffalo Bills

Leah Perlmutter (she/her)

Colby

amazon

EnergySavvy

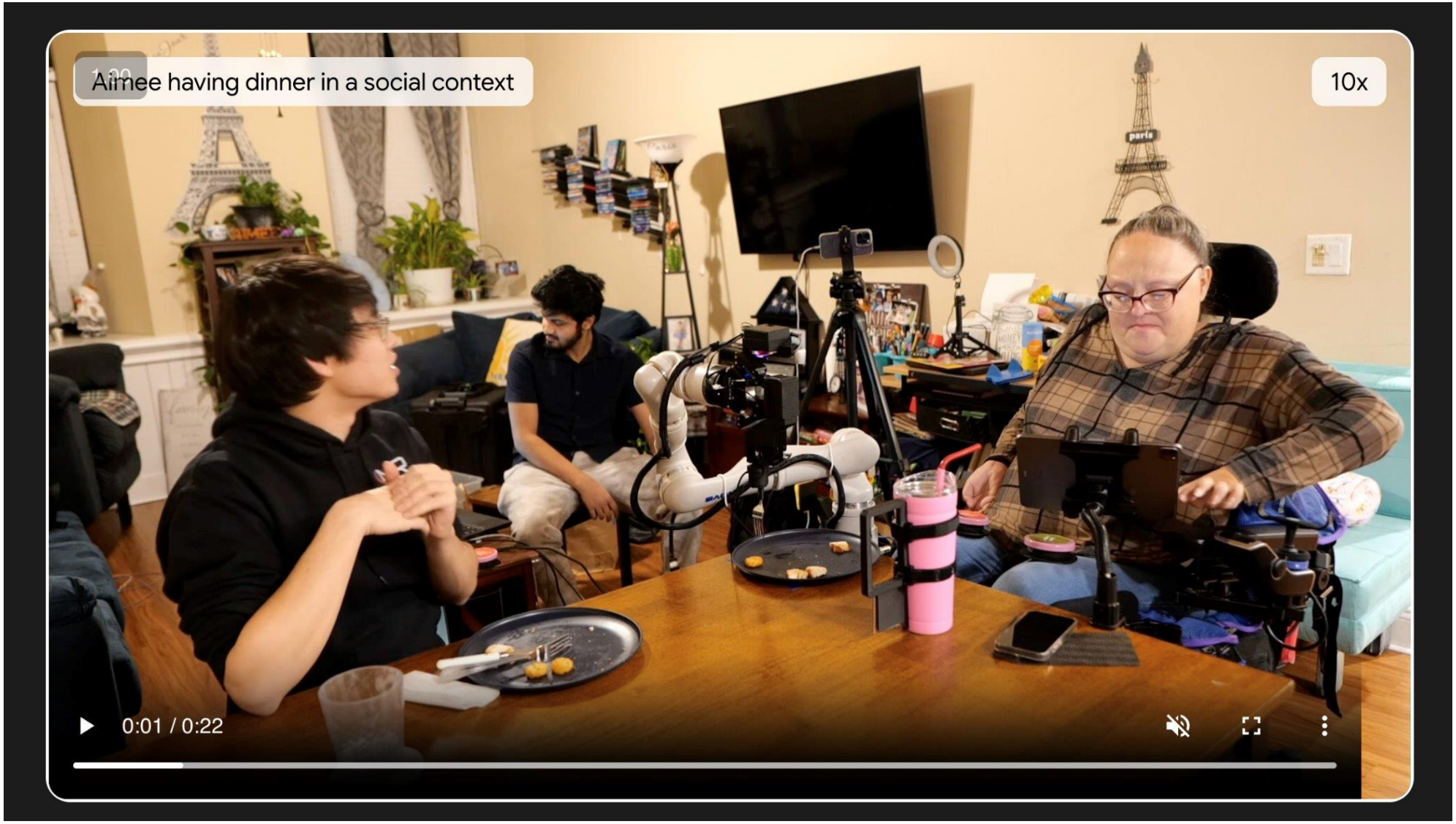
UNIVERSITY of WASHINGTON
PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

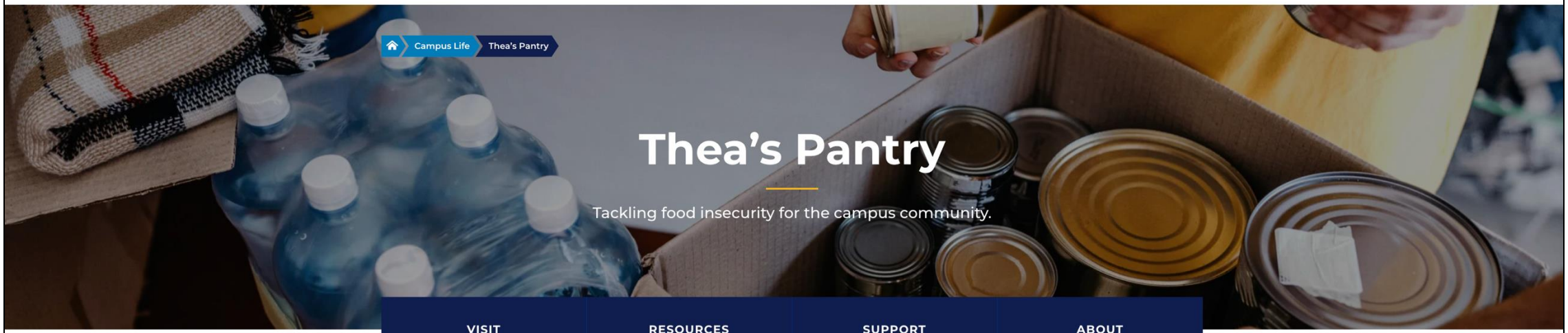
Grinnell College

Cornell Bowers CIS
College of Computing and Information Science

Technology for social good

- FEAST





Thea's Pantry

Tackling food insecurity for the campus community.

Campus Life
Campus Safety
Arts & Culture
Athletics
Clubs & Organizations +
Health Services +
Alcohol & Drug Prevention Education
Commuter Services +
Counseling Services +
Orientation +
Diversity, Inclusion & Equal Opportunity
Information Technology

A food pantry for the Worcester State community.

Thea's Pantry provides food and other essentials to students, staff, and faculty who are food insecure or at risk of food insecurity. Located in the Student Center, Thea's Pantry provides confidential services for those in need.

Sign up for the Thea's Pantry newsletter!

HOURS OF OPERATION

Summer 2025

Tuesday/Wednesday/Thursday 9am-3pm, by appointment only.

Thea's Pantry is located on the first floor of the Student Center, room 113. Thea's Pantry is not open on weekends. During school breaks, access to the pantry is by appointment only.

Search or go to...

Group

Theas Pantry
Manage
Plan
Code
Deploy
Operate
Analyze

What's new 8

Theas Pantry

Integrated software solution for Thea's Pantry campus food pantry at Worcester State University.

See the Documentation project for details.

Subgroups and projects | Shared projects | Shared groups | Inactive

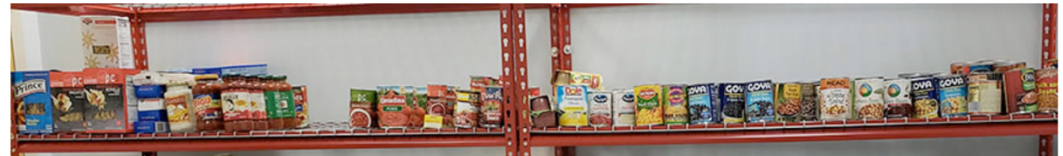
Search (3 character minimum) Name

A	**Archived** System components that are no longer being used. These are archived for historical purposes, and for possible future repurposing.	1	0	1
D	**Deployment** Building deployment and staging servers.	0	5	1
G	**GuestInfoSystem** The GuestInfoSystem provides services for tracking pantry guest information.	1	5	1
I	**IAMSystem** System for user authentication and access management.	0	9	1
I	**InventorySystem Category-Based** The InventorySystem provides services for tracking pantry inventory based on categories.	0	5	1
I	**InventorySystem Culling** System for removing inventory based on expiration dates.	0	4	1
I	**InventorySystem Weight-Based** The InventorySystem provides services for tracking pantry inventory based on weight.	0	7	1
R	**ReportingSystem** The ReportingSystem provides services for reporting on pantry usage.	0	6	1

D	**Documentation** Contains details about the Thea's Pantry client solution, including design, development processes, and licensing.	★ 0	2 months ago
G	**General** Used for general issues for Thea's Pantry.	★ 0	4 months ago

README.md

Thea's Pantry



decidim
free open-source democracy

First steps | Features | Community | Book | About | Donate | EN

Decidim is a digital platform for citizen participation

Free/libre, open and safe technology.
With all democratic guarantees.
Reprogramming democracy is now possible with Decidim.

Get started

Ajuntament de Barcelona | NYC | European Commission | Helsinki | Senat | Generalitat de Catalunya

- Strategic planning
- Participatory processes

Deadly systems may seem durable, but they're not inevitable! And we don't simply have to click "submit." We can each work to strengthen the social fabric in our own locale and create a shared vision where no one is left behind.
- Ruha Benjamin

Technology for social good

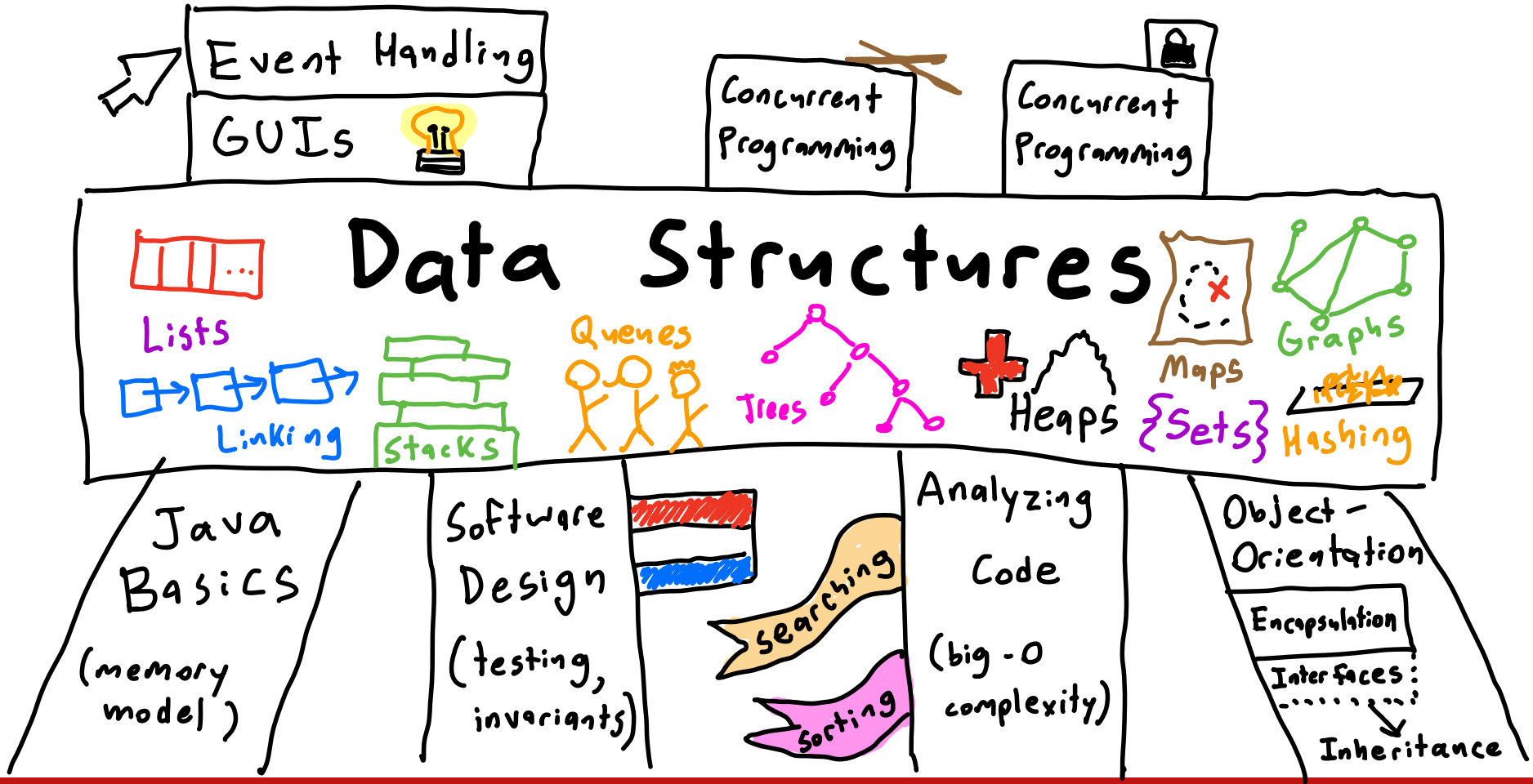
- FEAST
- Thea's Pantry / LibreFoodPantry
- Decidim
- Ruha Benjamin Ted Talk

Goal of CS 2110: Write *Better* Code

What makes code *better*?

- more efficient (speed, energy use, memory)
- easier to read, adapt, maintain, extend, develop
- user-friendly with good features
- more reliable, fewer bugs
- robust to security vulnerabilities
- ⋮

Course Map


Event Handling
GUIs
Concurrent Programming
Concurrent Programming
Data Structures
Lists
Linking
Stacks
Queues
Trees
Heaps
Maps
{Sets}
Graphs
Hashing
Java Basics
(memory model)
Software Design
(testing, invariants)
searching
sorting
Analyzing Code
(big-O complexity)
Object-Orientation
Encapsulation
Interfaces:
Inheritance

Course Logistics

Everything you need is linked from our course website

courses.cis.cornell.edu/courses/cs2110/2025fa/

(find a link on our Canvas page)

- Syllabus
- Lecture notes
- Slides and demo code (after lecture)
- Discussion materials
- Assignments
- Grades (coming soon)

"process over product"

metacognition

Let the learning begin!

Today's Learning Outcomes

1. Explain the difference between *statically* and *dynamically* typed programming languages.

2. Determine the static type of an expression involving one of more operators, method calls, and/or implicit/explicit type coercions.

3. Draw a memory diagram that visualize the state of execution of a program involving multiple method calls.

Poll Everywhere

PollEv.com/2110fa25 text 2110fa25 to 22333

What is your "go to" programming language?

Java (A)	Python (B)
C / C++ (C)	Matlab (D)
Javascript / Typescript (E)	R (F)
Something Else (G)	

A Basic Python Program

```
# Returns the number of minutes in `days` days
def days_to_minutes(days):
  hours = 24 * days
  minutes = 60 * hours
  return minutes

if __name__ == "__main__":
    print("Enter a number of days: ", end="")
    days = input() # get user input from the console
    mins = days_to_minutes(days)
    print("There are " + mins + " minutes in " + days + " days.")
```

start here

Poll Everywhere

PollEv.com/2110fa25 text 2110fa25 to 22333

What will be the output of this program?

There are 1440 minutes in 3 days. **(A)**

There are 4320 minutes in 3 days. **(B)**

(The program will crash because of an error) **(C)**

Something else... **(D)**

A Basic Python Program

```
# Returns the number of minutes in `days` days
def days_to_minutes(days):
  hours = 24 * days
  minutes = 60 * hours
  return minutes

if __name__ == "__main__":
    print("Enter a number of days: ", end="")
    days = input() # get user input from the console
    mins = days_to_minutes(days)
    print("There are " + mins + " minutes in " + days + " days.")
```

String concatenation

days is a String, not a number

Dynamic vs. Static Typing

The type of a variable (or expression) determines its possible values and how it can be used.

Dynamically Typed languages infer types at runtime
(Python) - shorter code, but less "safe"

Statically Typed languages check types before
(Java) code is run

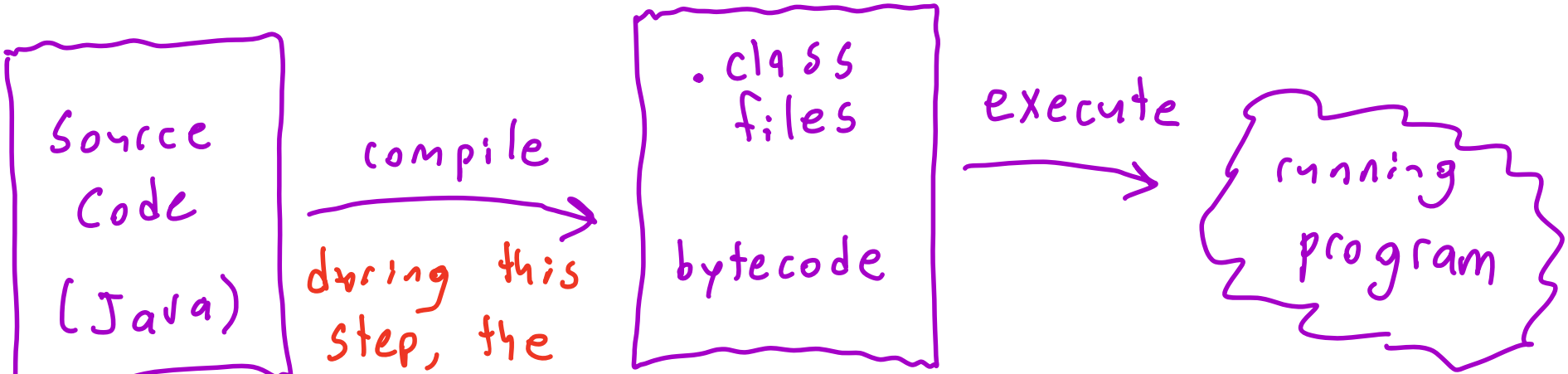
- usually require explicit type declarations by programmer

Coding Demo: Translation to Java

(We include slides like this one in case you want a place to take notes during/about the demo.)

Compilation

↳ intermediary "code translation" step that happens before code runs



Primitive Types

8 in Java, most important are:

- int (4 byte) (whole numbers)
- double (8 byte) (decimal numbers)
- char (2 byte) (characters)
- boolean (1 bit) (true, false)

- Fixed size, values stored directly in their variables

```
int age = 27;
```

27: int literal

Expressions

An expression is a unit of code with type/value.

- literals: 27 (int) 30.5 (double) true (boolean) 'A' (char)
- variables: int age=27; age (int, value=27)
- using operators: 27*12 (int, value=324)
- method call: static int daysToMins(int days) daysToMins(1) (int, value=1440)

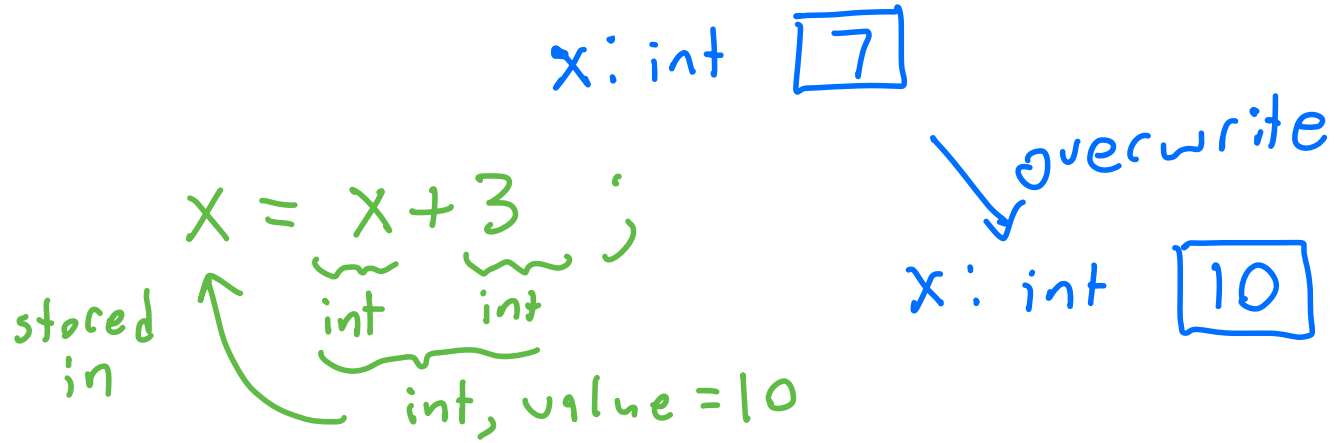
Compiler determines types of all expressions and checks for agreement.

Statements and Assignment

A statement is a unit of code that describes an action (i.e., produce a side-effect)

Ex. Assignment statements

```
int x = 7; // store RHS value in LHS container
```



Primitive Type Coercion

Most operations preserve primitive types:

$3_{\text{int}} + 4_{\text{int}} \rightarrow 7_{\text{int}}$

$3.0_{\text{double}} + 4.0_{\text{double}} \rightarrow 7.0_{\text{double}}$

Coercion lets us alter expression types

Implicit (widening): $3_{\text{int}} + 4.0_{\text{double}} \rightarrow 3.0_{\text{double}} + 4.0_{\text{double}} \rightarrow 7.0$

Explicit (casting): $7_{\text{int}} / 2_{\text{int}} \rightarrow 3_{\text{int}}$ (truncated division)

$(\underbrace{\text{double}}_{7.0})\ 7 / 2.0 \rightarrow 3.5$

* coercion affects expression evaluation but doesn't change underlying variables.

Poll Everywhere

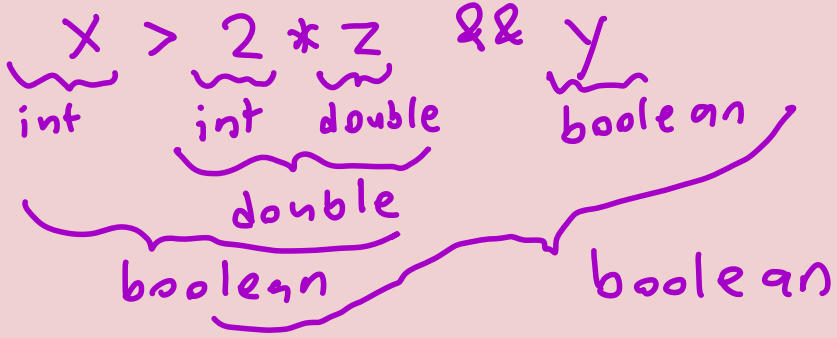
PollEv.com/2110fa25 text 2110fa25 to 22333

What is the static type of the highlighted expression?

* bonus poll, cut for time

```
int x = 6;
boolean y = true;
double z = 4.3;

System.out.print(x > 2 * z && y)
```

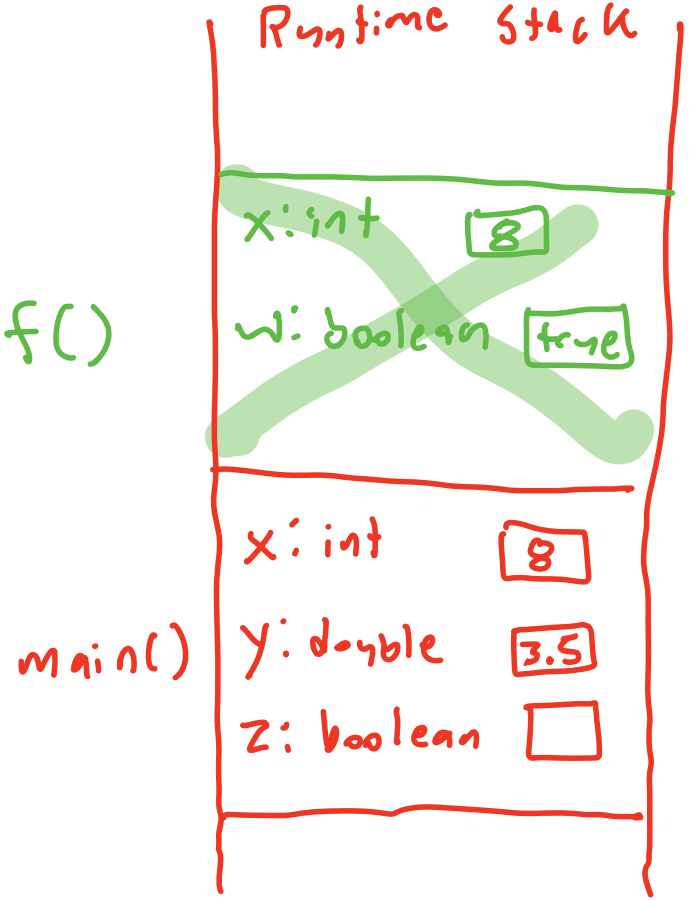


- (A) int
- (B) boolean
- (C) double
- (D) String

Program Execution

Starts with main() method

- whenever a method is invoked, a call frame is created on runtime stack
  - has space for all variables and parameters of that method
  - assignment statements update these values
- inner method calls "stack" up new call frames



Diagramming our Code

* See the animation in the lecture notes on the website to step through this example.

```
public class Convert {
  static int daysToMinutes(int days) {
    int hours = 24 * days;
    int minutes = 60 * hours;
    return minutes;
  }

  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in));
    int days = sc.nextInt();
    int mins = daysToMinutes(days);
    System.out.print(...);
  }
}
```