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 go to coffee/tea shop order? Enter your answer in the slido!

Music: CSE 122 Summer 2024

Instructors: Ido Avnon

TAs: Abby Williams
Chloë Mi Cartier
Connor Sun
Cynthia Pan
Katharine Zhang
Marcus Sanches
Rohini Arangam

Lecture Outline

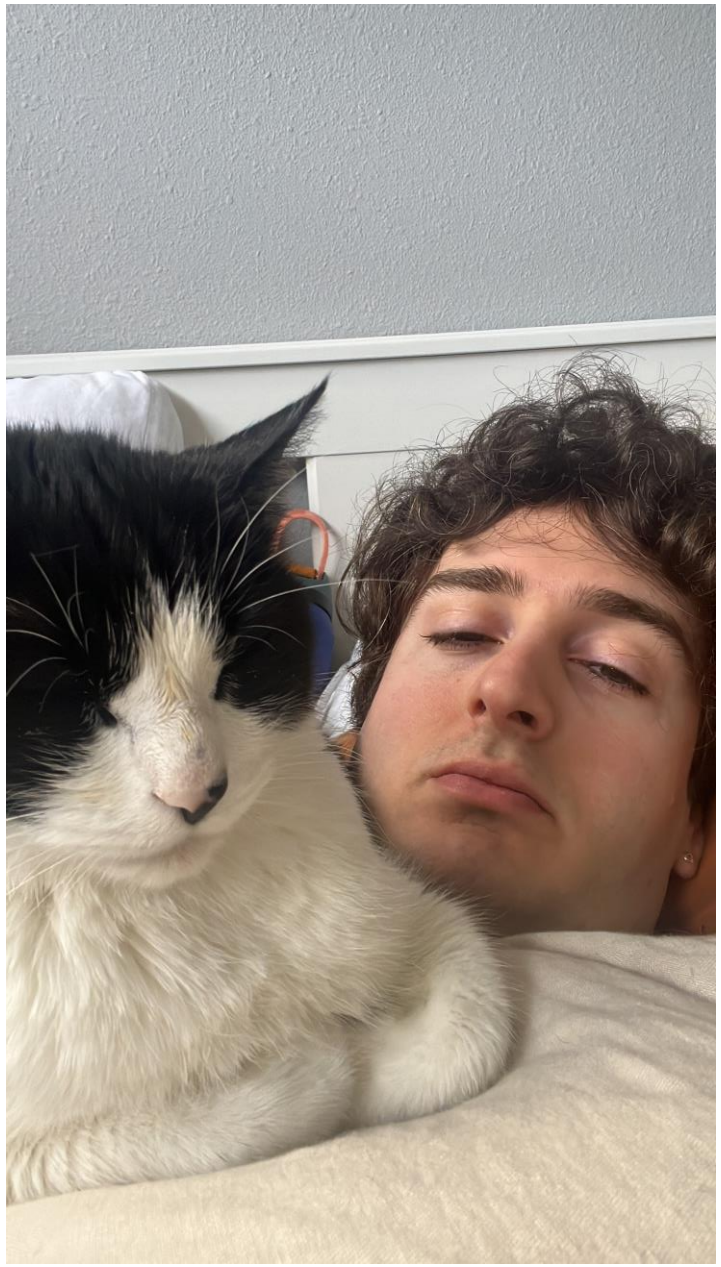
- Introductions/Announcements
- About this Course
  - Course Components & Tools
  - Policies
  - Making the Most of this Class
- Intro/Review Java
- Functional Decomposition

Course Staff

- Instructors: Miya Natsuhara and Kasey Champion

- Teaching Assistants: 7 Amazing TAs
  - 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 ☺



Announcements

- Hope you had fun in your first quiz section yesterday!
- Culminating Project 0 (C0) released, due next Thursday June 27th
  - Proposal for your project (no code)
- IPL and OH will also begin on Monday!
  - OH: Tuesday and Thursday, 2-3 or schedule a 1-on-1
- Introductory Survey
- Any questions from yester

Lecture Outline

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

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 Java Review Lesson useful!
  - If you want to know if this class is the right fit for you, take the Allen School Self-Placement Test

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
- Arrays & 2D arrays

CSE 122 – Computer Programming II

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

Course Components

Meetings

LECTURES

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

SECTIONS

- Held in person
- More practice, reviews, applications
- TA advice, how to be an effective student
- Preparation for quizzes / exams
- Not Recorded!

Assessments

PROGRAMMING ASSIGNMENTS (x4)

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

CULMINATING PROJECT (x4)

- Student designed
- Single project split up into four checkpoints

QUIZZES (x3)

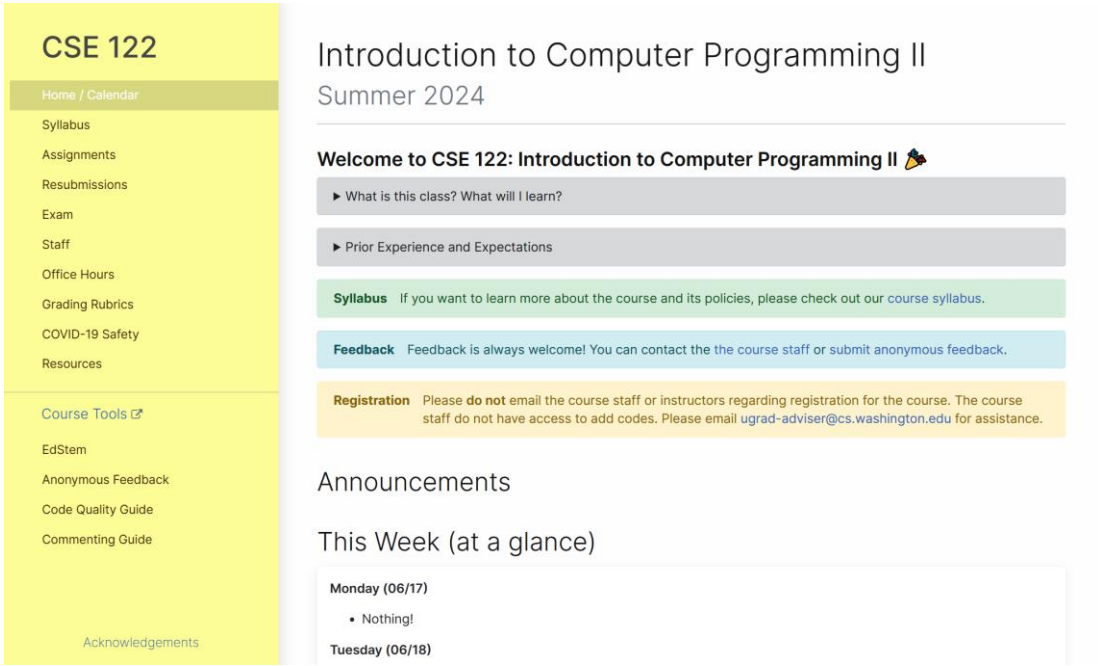
- Taken in quiz section
- 50 minutes on computer

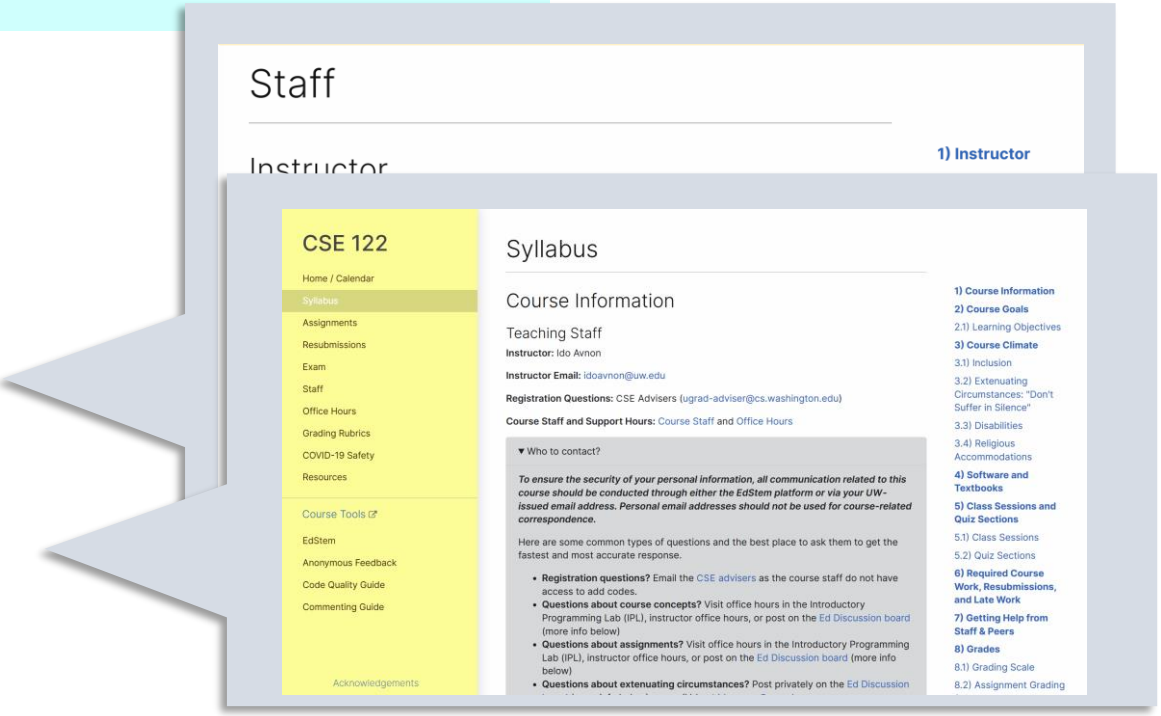
EXAM (x1)

- Culminating exam
- **Friday, August 16, 10:50-11:50**

Course Website

[cs.uw.edu/122](cs.uw.edu/122)





Please familiarize yourself with the course syllabus this week!

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

- Queueing in office hours

VSCode (Optional)

- Develop offline
- Visual debugger

Canvas

- Lecture recordings

Sli.do

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

Lecture Outline

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

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
- Generative AI Policy
- **Read full policy in Syllabus**

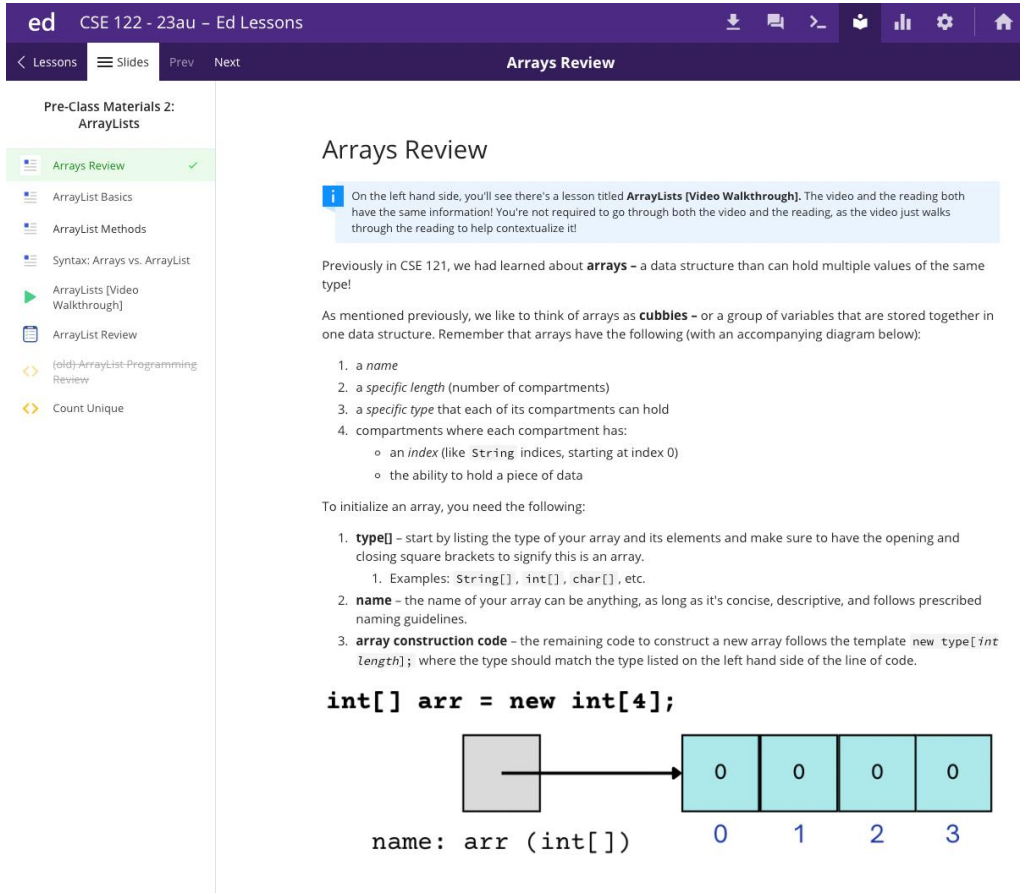
Textbook

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.



Lecture Outline

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

How do you learn best?

Questions during Class?

Raise hand or send here

sli.do #cse122

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). 310 of you >>> 31 of us!
  - For serious personal circumstances, you can email Miya and Kasey directly. It never hurts to email us, but if it's a common logistic question, we may politely ask you to post on the discussion board instead.

Lecture Outline

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

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`**

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

Practice: Think

sli.do #cse122

What is the output of this Java program?

```
public class Demo {
   public static void main(String[] args) {
      int[] nums = {1, 4, 4, 8, 13};

      int totalDiff = 0;
      for (int i = 1; i <= nums.length; i++) {
         totalDiff += (nums[i] - nums[i - 1]);
      }
      System.out.println("Total Diff = " + totalDiff);
   }
}
```

A) Total Diff = 12
B) Total Diff = 10
C) Total Diff = 9
D) Exception!

Practice:

sli.do #cse122

What is the output of this Java program?

```
public class Demo {
   public static void main(String[] args) {
      int[] nums = {1, 4, 4, 8, 13};

      int totalDiff = 0;
      for (int i = 1; i <= nums.length; i++) {
         totalDiff += (nums[i] - nums[i - 1]);
      }
      System.out.println("Total Diff = " + totalDiff);
   }
}
```

A) Total Diff = 12
B) Total Diff = 10
C) Total Diff = 9
D) Exception!

Class Example: Horoscope Guesser

A Short Program that:

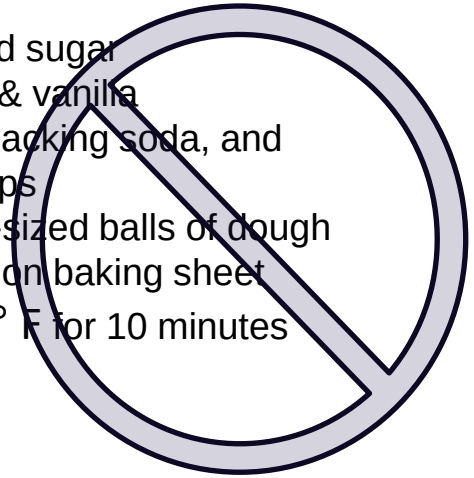
- Introduces itself to the reader
- Asks the user for input
- Uses that input to display a result

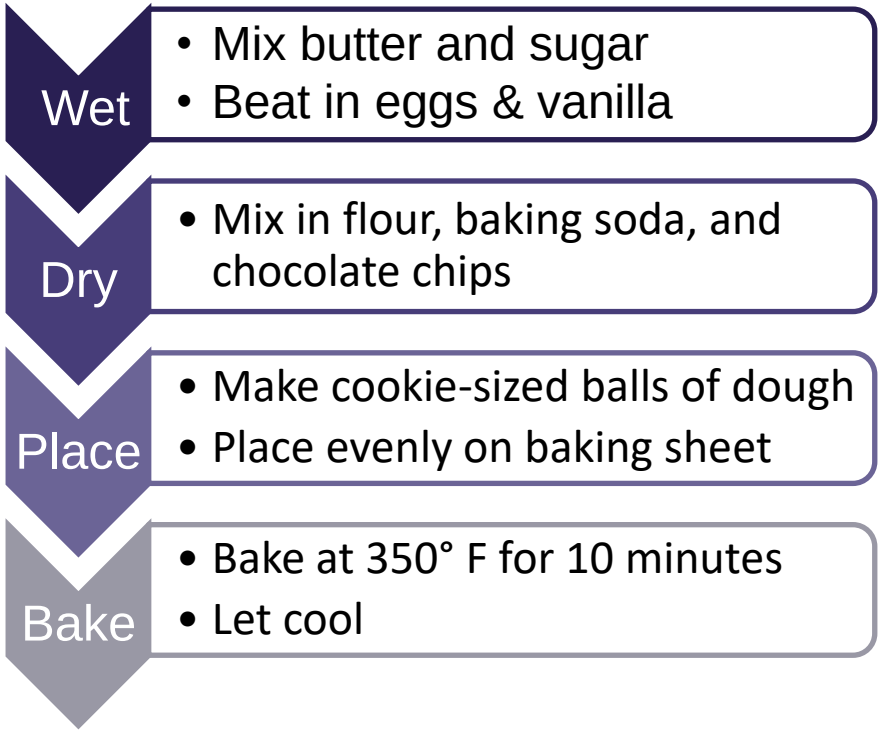
- Running on Ed
  - **Run** runs your program
  - **Mark** submits and runs autograder
    - Submit as many times as you like
  - **Solution** shows solution (if applicable)

Functional Decomposition

Functional decomposition is the process of breaking down a complex problem or system into parts that are easier to *conceive, understand, program,* and *maintain.*

"Bake the cookies"





TODOs for this Weekend 😊

- Think about your proposal for C0
  - This project spans the whole quarter, so think about something you would like to work on!
- PCM before lecture on Wednesday (website and Ed)!
- Attend section on Tuesday ☺