

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

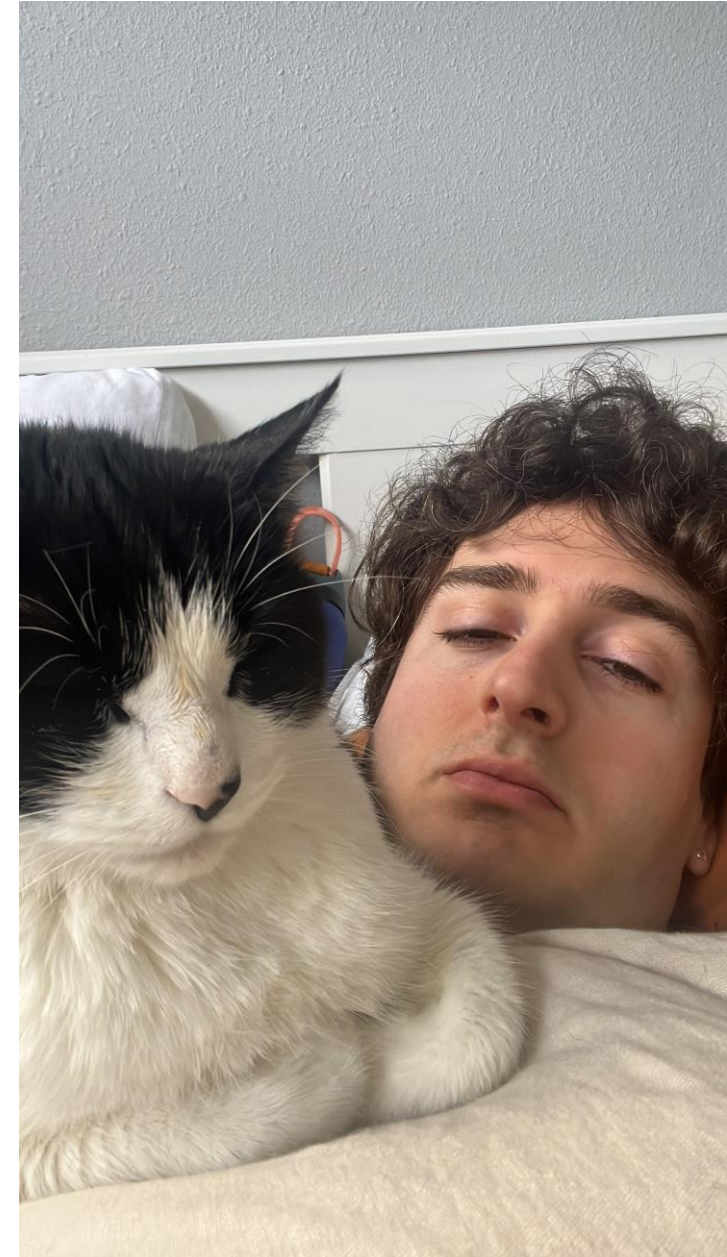
Lecture Outline

- **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

CSE 122

- Home / Calendar
- Syllabus
- Assignments
- Resubmissions
- Exam
- Staff
- Office Hours
- Grading Rubrics
- COVID-19 Safety
- Resources

Course Tools

- EdStem
- Anonymous Feedback
- Code Quality Guide
- Commenting Guide

Acknowledgements

Introduction to Computer Programming II

Summer 2024

Welcome to CSE 122: Introduction to Computer Programming II

- ▶ What is this class? What will I learn?
- ▶ Prior Experience and Expectations

Syllabus If you want to learn more about the course and its policies, please check out our course syllabus.

Feedback Feedback is always welcome! You can contact the the course staff or submit anonymous feedback.

Registration Please do not email the course staff or instructors regarding registration for the course. The course staff do not have access to add codes. Please email ugrad-adviser@cs.washington.edu for assistance.

Announcements

This Week (at a glance)

Monday (06/17)

- Nothing!

Tuesday (06/18)

Staff

Instructor

1) Instructor

- CSE 122**
- Home / Calendar
- Syllabus
- Assignments
- Resubmissions
- Exam
- Staff
- Office Hours
- Grading Rubrics
- COVID-19 Safety
- Resources

Course Tools

- EdStem
- Anonymous Feedback
- Code Quality Guide
- Commenting Guide

Acknowledgements

Syllabus

Course Information

Teaching Staff

Instructor: Ido Avnon

Instructor Email: idoavnon@uw.edu

Registration Questions: CSE Advisers (ugrad-adviser@cs.washington.edu)

Course Staff and Support Hours: Course Staff and Office Hours

Who to contact?

To ensure the security of your personal information, all communication related to this course should be conducted through either the EdStem platform or via your UW-issued email address. Personal email addresses should not be used for course-related correspondence.

Here are some common types of questions and the best place to ask them to get the fastest and most accurate response.

- **Registration questions?** Email the CSE advisers as the course staff do not have access to add codes.
- **Questions about course concepts?** Visit office hours in the Introductory Programming Lab (IPL), instructor office hours, or post on the Ed Discussion board (more info below)
- **Questions about assignments?** Visit office hours in the Introductory Programming Lab (IPL), instructor office hours, or post on the Ed Discussion board (more info below)
- **Questions about extenuating circumstances?** Post privately on the Ed Discussion board (more info below)

1) Course Information

- 2) Course Goals
- 2.1) Learning Objectives
- 3) Course Climate
 - 3.1) Inclusion
 - 3.2) Extenuating Circumstances: "Don't Suffer in Silence"
 - 3.3) Disabilities
 - 3.4) Religious Accommodations
- 4) Software and Textbooks
- 5) Class Sessions and Quiz Sections
 - 5.1) Class Sessions
 - 5.2) Quiz Sections
- 6) Required Course Work, Resubmissions, and Late Work
- 7) Getting Help from Staff & Peers
- 8) Grades
 - 8.1) Grading Scale
 - 8.2) Assignment Grading

Contains most course info – check frequently!
 Announcements, Calendar, Lecture Slides, Office Hours schedule,
 Staff Bios, Important Links

Please familiarize yourself with the course syllabus this week!

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

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.

The screenshot shows the Ed platform interface for 'Arrays Review'. The top navigation bar includes 'Lessons', 'Slides', 'Prev', and 'Next'. The main content area is titled 'Arrays Review' and contains an information icon with a note: 'On the left hand side, you'll see there's a lesson titled **ArrayLists [Video Walkthrough]**. The video and the reading both have the same information! You're not required to go through both the video and the reading, as the video just walks through the reading to help contextualize it!'. Below this, there is a section 'Previously in CSE 121, we had learned about **arrays** – a data structure than can hold multiple values of the same type!'. This is followed by a paragraph: 'As mentioned previously, we like to think of arrays as **cubbies** – or a group of variables that are stored together in one data structure. Remember that arrays have the following (with an accompanying diagram below):'. A list of four characteristics follows: 1. a *name*, 2. a *specific length* (number of compartments), 3. a *specific type* that each of its compartments can hold, and 4. compartments where each compartment has: an *index* (like `String` indices, starting at index 0) and the ability to hold a piece of data. Below the list, it says 'To initialize an array, you need the following:'. A list of three steps follows: 1. **type[]** – start by listing the type of your array and its elements and make sure to have the opening and 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 prescribed naming guidelines. 3. **array construction code** – the remaining code to construct a new array follows the template `new type[int length]`; where the type should match the type listed on the left hand side of the line of code. Below the text, the code `int[] arr = new int[4];` is shown. To the right of the code is a diagram of an array. A grey box labeled 'name: arr (int[])' has an arrow pointing to a row of four light blue boxes, each containing the number '0'. Below these boxes are the indices '0', '1', '2', and '3'.

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)

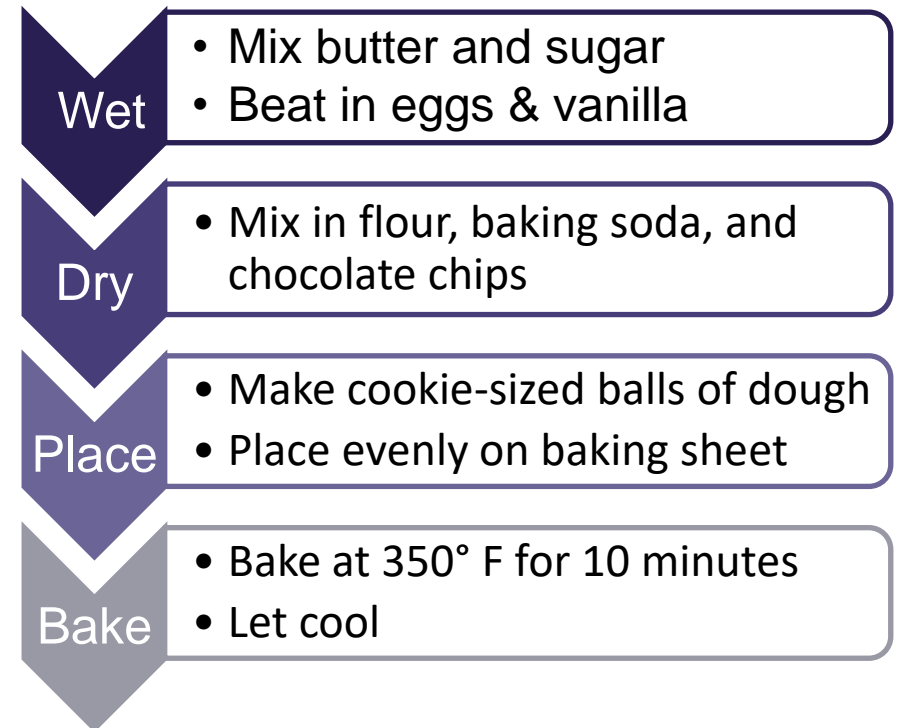
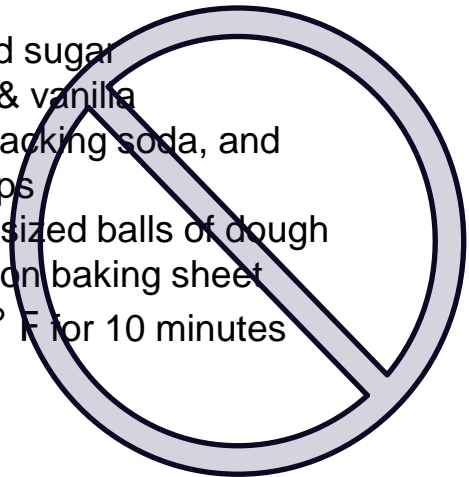


Functional Decomposition

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”

- Mix butter and sugar
- Beat in eggs & vanilla
- Mix in flour, baking soda, and chocolate chips
- Make cookie-sized balls of dough
- Place evenly on baking sheet
- Bake at 350° F for 10 minutes
- Let cool



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 😊