

LEC 01

CSE 122

Java Review; Functional Decomposition

Questions during Class?
Raise hand or send here

sli.do #cse122



BEFORE WE START

Talk to your neighbors:

What is your favorite handheld food?

Respond on sli.do!

Instructor: Brett Wortzman

TAs:	Archit	Daniel	Naomi	Sreshta
	Blake P.	David	Neha	Suyash
	Blake R.	Hanna	Nicole	Trisha
	Cady	Isis	Rio	Wesley
	Carson	Jun	Rohan	Yash
	Cole	Landon	Shreyank	
	Connor	Michelle	Somya	

Music: 🎸 [CSE 122 26au Lecture Tunes](#) 🎵

Lecture Outline

- **Announcements/Reminders** 
- Assignments/Grading
- Java Review
- Functional Decomposition

Announcements/Reminders

- Hope you had fun in your first quiz section yesterday!
- Review the [syllabus](#)
- Fill out the [Introductory Survey](#)

- Java Review Session on Monday, October 5th
 - 12:30 – 1:20pm in ARC 147; 2:30 – 3:20pm in SAV 260
 - Will be recorded!
- IPL will open on Monday, October 5th

Ed



- Our online learning platform
- Lessons, materials
- Discussion board, announcements
- Also, where we'll code and submit work!

The screenshot shows the Ed discussion board for CSE 122 - 26au. The interface includes a top navigation bar with the University of Washington logo and course title. On the left, there are sidebar menus for 'COURSES' and 'CATEGORIES'. The main content area displays a list of threads, with the top thread titled 'Day 0 Complete!' by Brett Wortzman. This thread has 406 views and 9 replies. The thread content includes a welcome message, a reminder to read the syllabus, and a list of important items for the week, such as meeting the TA and classmates, and completing pre-class materials. The thread also includes a link to the course website and a reminder to double-check the location of the quiz section.

ed W UNIVERSITY of WASHINGTON CSE 122 - 26au - Ed Discussion

New Thread Search Filter

COURSES

- Course Redesign Clinic s...
- CSE 121 - 26au 3
- CSE 122 - 26au
- CSE 123 - 26au 3
- STAFF CSE 12x/14x C... 543
- 8 more

CATEGORIES

- General
- Pre-Class Materials
- Lectures
- Sections
- Projects
- Computer Quizzes
- Paper Quizzes
- Grading Clarifications
- Logistics/Policy
- Social
- Slido Megathreads
- Announcements

Day 0 Complete! 0 0 0 #4

Brett Wortzman INSTRUCTOR/LEAD TA 17 hours ago in Announcements 406 UNPIN STAR WATCHING VIEWS

Hi all!

9 It was so nice getting to meet and see you all earlier today! As I mentioned in class, I am excited to work with you this quarter. 🥳

I wanted to send a reminder about some of the things we talked about today and some important items for the rest of the week:

- First and foremost, a reminder that **our website is at <http://cs.uw.edu/122>**. As I mentioned in class and in #1, the website will be your one-stop shop for accessing all course materials and resources.
 - Whenever you access the website, make sure you're on the Autumn 2026 version-- that *may not* be the top result from search engines.
- Please take some time and **read through the syllabus!** As I said in class today, it's long but full of useful information and ensures that the course staff and students have a common agreement about the structure and policies of the course.
- **Go meet your TA and classmates in Thursday's quiz section tomorrow!** You can find the room and on your MyUW registration.
 - Make sure to **double-check the location of your quiz section on MyUW**; sometimes there are last-minute room changes.
- We've put together an **introductory survey** to get a better sense of who is in the class! This isn't strictly required, but seeing the responses is super helpful for us. We've been giving a version of this survey every quarter for many years, and we're able to look at the data to better understand the students we're serving, and what adjustments we should make to serve them better. Please take a few minutes to fill this out this week!
 - Getting a permissions error for the form? **Make sure you are logged in to your UW NetID Gsuite account!**
- Before attending class on Friday, **complete the Pre-class Materials**. Find the link on the calendar on the course website. We will start class on Friday where the Pre-class Materials left off.

Have a great time meeting your TAs and classmates at your quiz sections tomorrow!

-Brett

Comment Edit Delete ...

Other Course Tools



sli.do

- Ask questions in class
- Live activities (ungraded)
- No account needed



Canvas

- Panopto lecture recordings (also linked from website)
- Some grades*

My Digital Hand

My Digital Hand

- Queueing in IPL



Gradescope

- Paper quiz grading



Visual Studio Code (optional)

- Offline IDE

Help Us Improve!

- This is a relatively new course, and we've made lots of changes this quarter!
- We are always looking for feedback on how to improve the class for you and for future students! Thank you in advance for your patience and understanding as we develop everything. 😊
 - 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!
- Post on the discussion board (can be public/private).
 - Note: Anonymous here is anonymous to other students, not to the staff.
- Submit feedback via the **Anonymous Feedback Tool** (linked under "Course Tools" on the website)

Lecture Outline

- Announcements/Reminders
- **Assignments/Grading** ◀
- Java Review
- Functional Decomposition

Course Components

PRE-CLASS MATERIALS

x18ish

- Introduce new concepts/ideas
- Readings, sample problems
- Due **before** lecture

LECTURES

x20



- Expand on new concepts
- Apply and practice
- In person, recorded

SECTIONS

x16

- Review concepts
- More apply and practice
- Get TA advice
- In person, **not** recorded

Lessons

PROGRAMMING PROJECTS

x4

- Structured assignments
- Applying & implementing course concepts
- AI and collaboration allowed

COMPUTER QUIZZES

x4

- 45 minutes on computer
- Expand on and connect to projects
- Scheduled outside of class

PAPER QUIZZES

x4

- 45 minutes on paper
- 3 in quiz section, 1 on exam day

PARTICIPATION

x1

- Complete PCMs, participate in section

CODE QUALITY

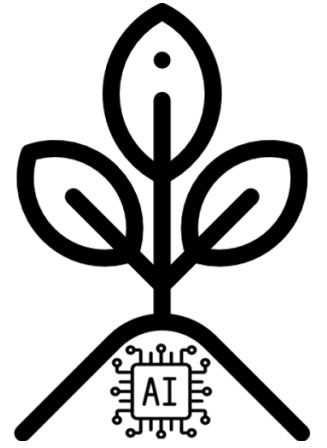
x3*

- Write code well
- Assessed throughout quarter

Graded Components

Collaboration and AI Policy

- You may collaborate with others and/or use AI freely for **general questions, explanations, examples, and ungraded work**
- On **Programming Projects**, you may collaborate with others and/or use AI freely
 - All collaboration and AI usage must be documented
- One **Computer Quizzes** and **Paper Quizzes**, you *may not* collaborate with others or use AI, and you will be restricted to limited resources



AI for Support and Growth

Assessment

Why assessment?

- Our goal: you gain **proficiency of the concepts and skills** we teach
- We *assess* your proficiency by asking you to apply these concepts and skills on tasks
- By necessity, we are assessing your work as a proxy for proficiency

Grading Scheme

Grades should reflect proficiency in course objectives.

All assignments, quizzes, and exams are graded with an “E/S/N” grade:

- **E (Excellent)**
- **S (Satisfactory)**
- **N (Not Yet)**
- **[U (Unassessable)]**

Graded Course Components

Category	ESN Grades Per	Grading Notes
PROGRAMMING PROJECTS	1	Autograded based on requirements in assignment
COMPUTER QUIZZES	4	4 problems, 1 grade per problem - Manually graded (with possible automated testing)
PAPER QUIZZES	4	4 problems, 1 grade per problem - Manually graded
CODE QUALITY	3	3 dimensions, 1 grade per dimension - Assessed on computer quizzes - Aggregated throughout quarter
PARTICIPATION	1	- Complete PCMs, participate in section - At least 26/32 → E - At least 22/32 → S

See [syllabus](#) for more details

Assignment Grades

Category	Count	ESN Grades Per	Total ESN Grades
PROGRAMMING PROJECTS	4	1	4
COMPUTER QUIZZES	4	4	16
PAPER QUIZZES	4	4	16
CODE QUALITY	1	3	3
PARTICIPATION	1	1	1
<i>Total</i>			<i>40</i>

See [syllabus](#) for more details

Course Grades

We provide a “*minimum grade guarantee*” for translating to a numeric grade, but not a direct formula.

Minimum Grade	Requirements
<i>Total ESN available</i>	<i>40</i>
3.5	32 Es AND 3 additional S+
3.0	24 Es AND 8 additional S+
2.5	16 Es AND 13 additional S+
2.0	8 Es AND 18 additional S+
1.5	18 S+
0.7	10 S+

S+ = S or E

See [syllabus](#) for more details

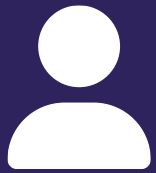
Lecture Outline

- Announcements/Reminders
- Assignments/Grading
- **Java Review** ◀
- Functional Decomposition

Reminders: Review Java Syntax

[Java Tutorial](#) reviews all the relevant programming features you should familiar with (even if you don't know them in Java).

- Printing and comments
- Variables, types, expressions
- Conditionals (if/else if/ else)
- Loops (for and while)
- Strings
- Methods
- Arrays
- 2D arrays



Practice : Think

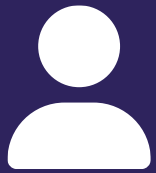


sli.do

#cse122

In-Class Activities

- **Goal:** Get you actively participating in your learning
- Typical Activity
 - Question is posed
 - **Think** (1 min): Think about the question on your own
 - **Pair** (2 min): Talk with your neighbor to discuss question
 - If you arrive at different conclusions, discuss your logic and figure out why you differ!
 - If you arrived at the same conclusion, discuss why the other answers might be wrong!
 - **Share** (1 min): We discuss the conclusions as a class
- During each of the **Think** and **Pair** stages, you will respond to the question via a sli.do poll
 - Not worth any points, just here to help you learn! 😊



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 : Pair

sli.do [#cse122](https://twitter.com/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!

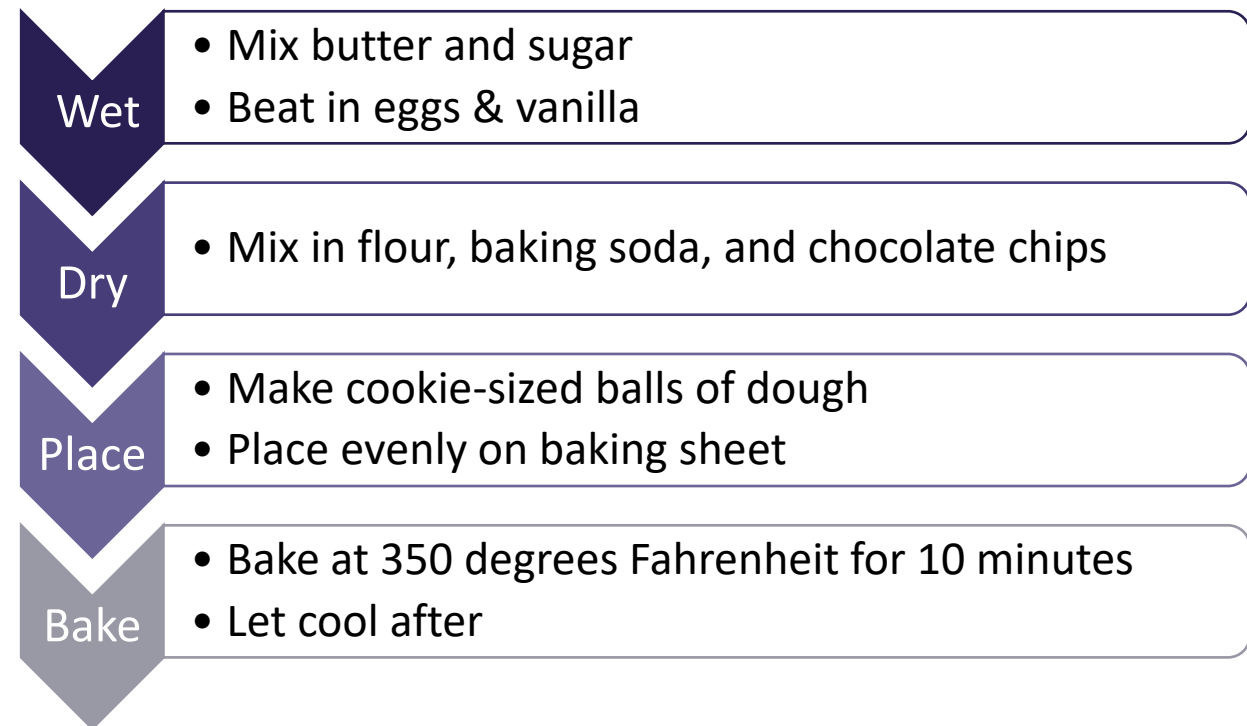
Lecture Outline

- Announcements/Reminders
- Assignments/Grading
- Java Review
- **Functional Decomposition** ◀

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” →



Functional Decomposition

- We should use methods to:
 - Reduce redundancy
 - Increase reusability
 - Generalize functionality
 - Improve readability
 - Abstract away details
 - ...
- *We should not* use methods to:
 - Give new names to existing methods
 - “Shorten” code

Functional Decomposition

In our code, functional decomposition often means breaking a task into smaller methods (also called functions).

Example: Bingo

- Set up populated game card
- Roll and call a new number
- Mark your card if the number called is present
- Check card for bingo

Avoid Trivial Methods

Introduce methods to decompose a complex problem, not just for the sake of adding a method.

Bad example:

```
public static void printMessage(String message) {  
    System.out.println(message);  
    System.out.println();  
}
```

Good Example:

```
public static String prompt(Scanner input, String prompt) {  
    System.out.print(prompt);  
    String response = input.nextLine();  
    return response;  
}
```

Rule of thumb: A method should do at least two steps

- Ask yourself: Does adding this method make my code easier to understand?