LEC 00

CSE 123

Welcome & Syllabus

Questions during Class?
Raise hand or send here

sli.do #cse123

BEFORE WE START

Talk to your neighbors:
Introduce yourself to your neighbor!

What is your name? Major? What have you been up to the past week?

Instructor: Brett Wortzman

TAs:

- Arohan
- Ashar
- Sean
- Chloe
- Jenny
- Nate
- Jonah
- Brice
- Chris
- Elden
- Ishita
- Kuhu
- Kavya
- Misha
- Kieran
- Sahana
- Anirudh
- Crystal
- Eeshani
- Aidan
- Cora
- Dixon
- Nhan
- Trien
- Evan
- Rena
- Katharine
- Anya

Now playing: 🎶 CSE 123 26wi Lecture Tunes 🎶

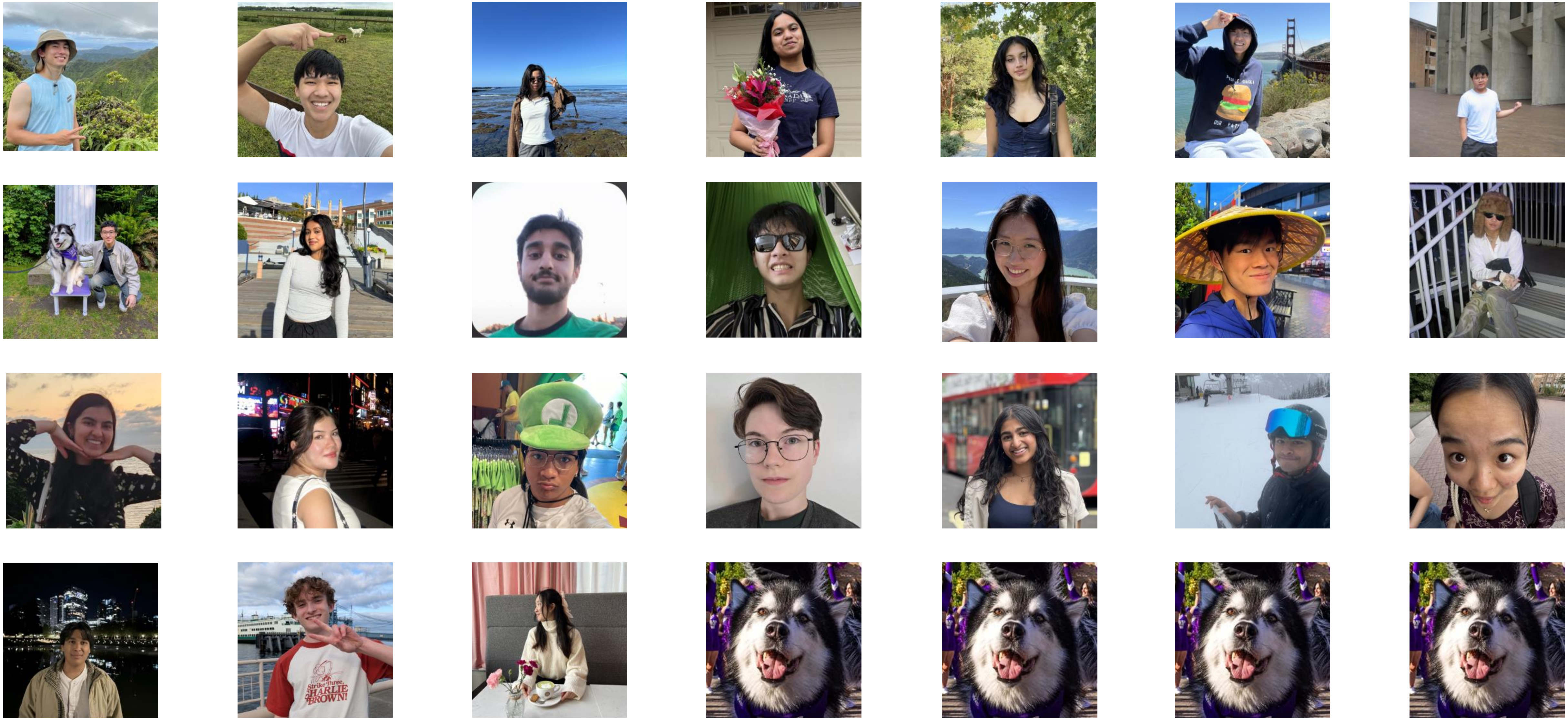
Lecture Outline

- Introductions
- About this Course
  - Course Components & Tools
  - Making the Most of this Class
- Assignments and Grading

Hi, I’m Brett! (he/him)

- Associate Teaching Professor
- Frequent intro CS instructor
  - Lead designer/developer of new 12X curriculum
- Also interested in CS education/pedagogy
- Previously:
  - trained CS teachers (and still!)
  - developed CS curriculum
  - taught high school CS
  - worked as a software engineering
- Non-CS hobbies: board games/RPGs, officiating football, announcing robotics competitions

Meet your 28 fabulous TAs!



What is this Class?

CSE 121 – Computer Programming I

- Data types (int, String, boolean)
- Methods / Functions
  - Parameters, Returns
- Control structures
  - Loops, Conditionals
- Arrays & 2D arrays
- **Computational Thinking** (language agnostic)

CSE 122 – Computer Programming II

- Functional Decomposition
- File I/O
- Using data structures
  - List, Stacks / Queues, Sets, Maps
- Object Oriented Programming
  - Interfaces

CSE 123 – Computer Programming III

- Advanced Object Oriented Programming
  - Comparable, Inheritance/Polymorphism, Abstract Classes
- Implementing data structures
  - ArrayLists, LinkedLists, Trees
- Recursion
- Critical analysis of design

Why 123?

1. To solve more complex problems by leveraging more complex programming structures / patterns

2. To better rationalize specific design decisions
   - How to "best" structure programs
   - Which data structures are "most" appropriate to use

Be a better programmer

3. To understand and critically analyze intersections between Computer Science and society
   - Search engines, algorithmic art, machine learning, etc.
   - Developing informed opinions on current issues

Be a better person

Prerequisite Knowledge

- Comfort with control structures
  - loops, conditionals, methods/functions
- Experience with using basic data structures
  - arrays, lists, sets, maps
- Experience with console and file input/output
- Exposure to simple object-oriented programming
  - classes, interfaces
- Programming experience *in Java*
  - Or willingness to pick up on your own

Course Expectations

What are you hoping to get out of taking CSE 123?

sli.do #cse123

Programming Skill

- Fluency with the relevant language and tools
    - consists of *many* conceptually simple facts
        - *e.g.* syntax (of a for loop, method, ...) , how to run a program
    - eventually becomes second nature

- Mental model of how programs execute
    - consists of a few deep concepts
        - *e.g.* what *is* a variable? a loop? a method? a reference?
    - develop the skill of ***tracing*** the current state of a program

Key course takeaway!

Lecture Outline

- Introductions
- **About this Course**
  - **Course Components & Tools**
  - Making the Most of this Class
- Assignments and Grading

Course Components

LECTURES x20

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

SECTIONS x16

- Held in person
- More practice, reviews, applications
- TA advice, how to be an effective student
- Preparation for quizzes / exams
- Exit ticket done at end of section

Meetings

PROGRAMMING ASSIGNMENTS x4

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

CREATIVE PROJECTS x4

- More open-ended assignments
- Explore new ideas and applications

QUIZZES x3

- Taken in quiz section
- 45 minutes on paper

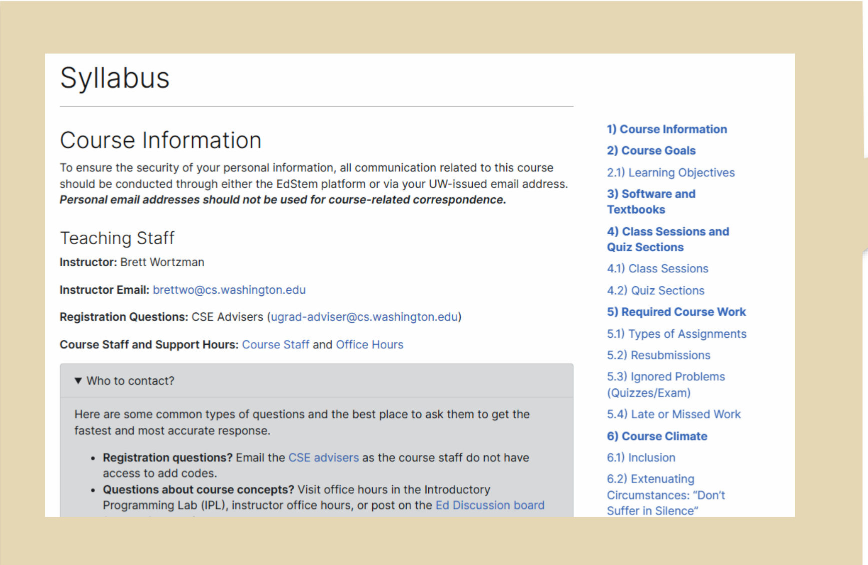
EXAM x1

- Culminating exam
- **Tue, March 17th**
- 12:30 – 2:20 PM

Graded Assignments

Course Website

[cs.uw.edu/123](cs.uw.edu/123)



Syllabus

Course Information

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.***

Teaching Staff

Instructor: Brett Wortzman

Instructor Email: brettwo@cs.washington.edu

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

Course Staff and Support Hours: Course Staff and Office Hours

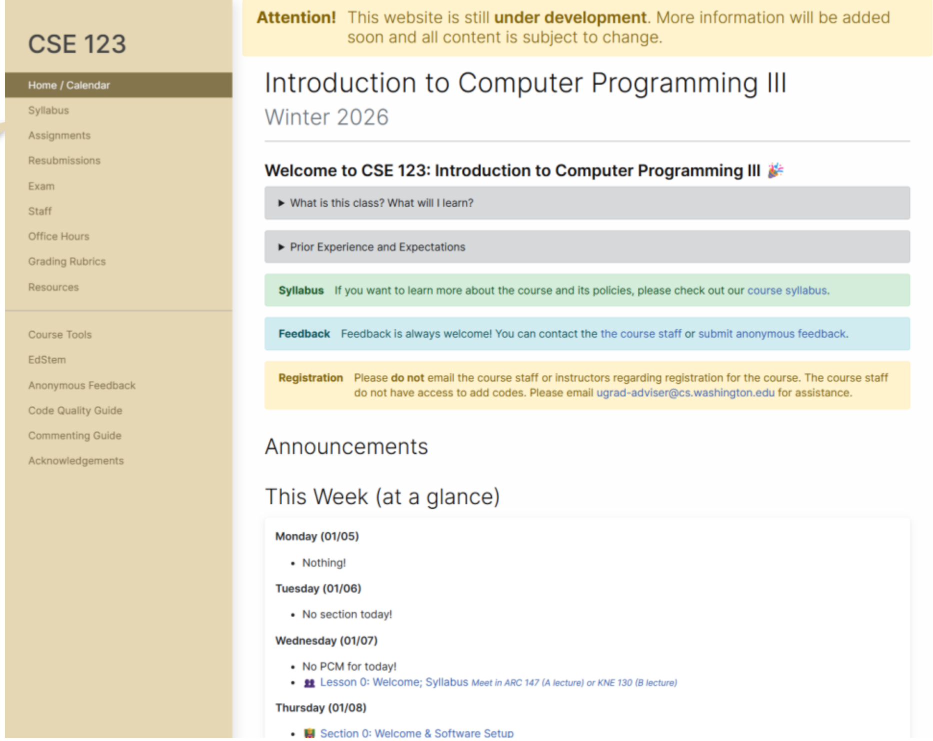
▼ Who to contact?

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

1) Course Information
2) Course Goals
2.1) Learning Objectives
3) Software and Textbooks
4) Class Sessions and Quiz Sections
4.1) Class Sessions
4.2) Quiz Sections
5) Required Course Work
5.1) Types of Assignments
5.2) Resubmissions
5.3) Ignored Problems (Quizzes/Exam)
5.4) Late or Missed Work
6) Course Climate
6.1) Inclusion
6.2) Extenuating Circumstances: "Don't Suffer in Silence"

Review the syllabus!



Attention! This website is still **under development**. More information will be added soon and all content is subject to change.

Introduction to Computer Programming III

Winter 2026

Welcome to CSE 123: Introduction to Computer Programming III 🎉

▶ 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 (01/05)

- Nothing!

Tuesday (01/06)

- No section today!

Wednesday (01/07)

- No PCM for today!
- 👥 Lesson 0: Welcome; Syllabus *Meet in ARC 147 (A lecture) or KNE 130 (B lecture)*

Thursday (01/08)

- Section 0: Welcome & Software Setup

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)
  - Announcements
- Pre-Class Materials / Section Handouts
- Assignments
  - Online IDE
  - Submit assignments
  - View Feedback

My Digital Hand

- Queueing in office hours

VSCode

- Develop offline
- Visual debugger

Gradescope

- Quiz and final exam grades

Sli.do

- In-class activities (ungraded)

Lecture Outline

- Introductions
- **About this Course**
  - Course Components & Tools
  - **Making the Most of this Class**
- Assignments and Grading

FLYING PIGEON

Digression: My Pandemic Hobby

Amigurumi: Japanese art of creating crocheted or knitted stuffed toys

How Learning Works

- Learning requires **active participation** in the process. It’s not as simple as sitting and listening to someone talk at you.
  - Requires **deliberate practice** in **learning by doing**
  - Benefits from **collaborative learning**

- Hybrid classroom model
  - Asks you to do some preparation before class in the form of readings and practice problems.
    - Should take ~30 minutes outside of class per lesson
  - Class will start with brief recap, then pick up where the reading and practice problems leave off.
  - Attendance isn’t graded, but showing up and trying is the first step in succeeding in the class!

- Pre-class materials are ungraded, but…
  - It’s okay if you find them challenging! That means you are learning!

Learning Pattern

• Pre-class Work	
- Your first introduction to a topic	Instruction
- Interactive components	Practice
• In-class lessons	
- Recap of pre-class work	
- Further instruction from where it left off	Instruction
- Interactive components	Practice
• Section	
- Review, restate, rephrase content	Instruction
- Very interactive	Practice
• Homework	
- Learn by applying	Practice/Assessment
- Help when you need it!	Instruction
• Quizzes/Final Exam	
- Show us what you've learned	Assessment

Learning in CSE 123: Live Support Systems

Programming is hard! We **want** to give you collaborative support!

Introductory Programming Lab (TA Office Hours) – starting Week 2

- > 40 hours/week (and highly rated in the class!)
- Face-to-face help from TAs on **any** course questions

Instructor Office Hours

- Great for things from lecture, personal questions, or just saying hi

Learning in CSE 123: Async Support Systems

Ed Board

- Best for content and logistics questions – 445 of you >> 29 of us!!
- Encourage public posts, except for things about **your** graded work
- Answer other students' questions – great way to learn!

Email

- Best for personal circumstances and/or private questions
- If unsure, always feel free to email Brett
  - May politely ask you to post on Ed instead!
- For emails, **please use your UW email** (protecting student privacy!)

Practice: Pair

sli.do #cse123

Think-Pair-Share: Inclusive Environments

CSE 123 will have many think-pair-share activities. Let's practice! Today's think:

What was an experience you had that made you feel welcome or included in a learning environment?

1. **Think** on your own, in silence for about ~ 30 seconds
2. **Pair** with your neighbor about it (and introduce yourself!!)
3. **Share** in sli.do and in class (I'll take a few volunteers from both)

Practice: Pair

sli.do #cse123

Think-Pair-Share: *Exclusive* Environments

CSE 123 will have many think-pair-share activities. Let's practice! Today's think:

What was an experience you had that made you feel unwelcome or excluded in a learning environment?

1. **Think** on your own, in silence for about ~ 30 seconds
2. **Pair** with your neighbor about it (and introduce yourself!!)
3. **Share** in sli.do and in class (I'll take a few volunteers from both)

Lecture Outline

- Introductions
- About this Course
  - Course Components & Tools
  - Making the Most of this Class
- **Assignments and Grading** ◀

Assignments and Grading

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

Assignments

- Your learning in this course will be assessed in four ways:
  - Programming Assignments (~biweekly, 4 total)
    - Structured programming assignments to assess your proficiency of programming concepts
  - Creative Projects (~biweekly, 4 total)
    - Smaller, more open-ended assignments to give you space to explore
  - Quizzes (3 total, in section)
    - Series of problems covering all material up to that point
  - Final Exam (Tuesday, March 17, 12:30-2:20)
    - Final, culminating assessment of all your skills and knowledge

Resubmission and Ignored Quiz Problems

Learning takes time, and doesn't always happen on the first try

- One previous Programming Assignment or Creative Project can be **resubmitted** each week
  - Must be accompanied by a write-up describing changes (via Google Form)
  - Grade on resubmission will replace original grade
  - An assignment can be resubmitted in the 3 cycles after feedback has been published
  - *Tip: Resubmit as early as possible!*
- We will ignore your **two lowest quiz/exam problem grades**
  - No special action required– we'll do this automatically

- See the syllabus for more details

Grading

Grades should reflect your proficiency in the course objectives

- All assignments will be graded **E (Excellent), S (Satisfactory), or N (Not yet)**
  - Under certain circumstances, a grade of U (Unassessable) may be assigned
- Final grades will be assigned based on the **amount of work at each level**

- See the syllabus for more details

Collaboration Policy

- When we assess your work in this class, we need to know that it's *yours*.
- Unless otherwise specified, **all graded work must be completed individually and without touching AI tools.**

Some specific rules to highlight:

  - do not share your own solution code or view solution code from any source – including but not limited to other students, tutors, or the internet
  - do not use AI tools (e.g. ChatGPT) to create graded work

See the syllabus for more details (this is *very* important to understand).

AI and CSE 123: Our Philosophy

Computing applications enabled by **artificial intelligence (AI)** are increasingly common and more widely used for a variety of tasks.

It is becoming more difficult to teach an introductory computing course without acknowledging the **existence of AI tools**.

But as relatively new programmers, **you still need to learn and practice effectively using core programming ‘building blocks’**.

CSE 123 AI Policy

No part of any graded work may touch an AI tool.

You may not copy and paste any work generated by AI into any graded submission, nor may you copy and paste any work from or for a graded assignment into an AI tool. All other uses of AI on graded work must be cited.

cs.uw.edu/123/syllabus/#using-ai-in-cse-123

CSE 123 AI Policy

No part of any graded work may touch an AI tool.

You may not copy and paste any work generated by AI into any graded submission, nor may you copy and paste any work from or for a graded assignment into an AI tool. All other uses of AI on graded work must be cited.

ALLOWED

- Asking AI to **explain an error message**
- Asking AI to **explain the functionality of non-graded code** snippets
- Asking AI to **suggest additional information** or resources

PROHIBITED

- **Generating code, comments, reflections**
- Using AI to **'solve' an assignment**
- Using AI to **write, modify, or extend** reflections, code, comments, etc.

cs.uw.edu/123/syllabus/#using-ai-in-cse-123

Coming up…

- 🧑‍🏫 Go to your first quiz section and meet your TAs tomorrow!
  - Make sure to double check MyUW for the location.
- 💻 Complete Pre-Class Work 1 before class on Friday!
  - It will be posted and linked from the course calendar later today.
- ❓ Complete the Introductory Survey
  - This helps us gather data about the students taking our classes and their backgrounds, to inform future offerings.