Welcome to CSE 121!

Miya Natsuhara

Spring 2023

Music: 121 23sp Lecture Vibes

TAs:				
	Jasmine	*Atharva*	*Mia*	*Justin*
	Shananda	*Julia*	*Archit*	*Aishah*
	Vidhi	*Anju*	*Grace*	*Claire*
	Larry	*Lydia*	*Kailye*	*Lydia*
	Jacqueline	*Jonus*	*Joshua*	*Kai*
	Afifah	*Hugh*	*James*	

sli.do #cse121

Agenda

- About me
- About this course
  - Learning objectives
  - Other similar courses
  - Course components
- Our learning model
- Tools and resources
  - Course Website
  - Ed
- Assessment and grading
- Collaboration

Agenda

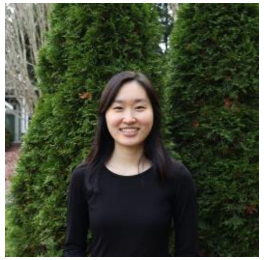
- About me ⟵
- About this course
  - Learning objectives
  - Other similar courses
  - Course components
- Our learning model
- Tools and resources
  - Course Website
  - Ed
- Assessment and grading
- Collaboration

Hi, I'm Miya! (she/her)

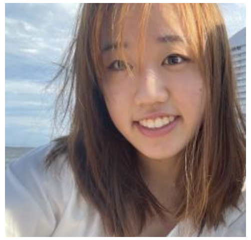
- Lecturer in the Allen School
  - (soon to be Assistant Teaching Professor)
- UW CSE alum
  - BS in CSE
  - BA in Math
  - Minor in ASL
  - MS in CSE
- Former Software Engineer at Microsoft
- Dog mom

Meet your 23 TAs!

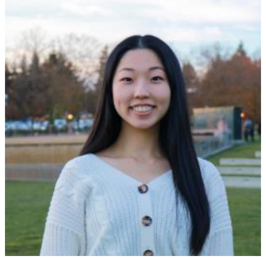
















Agenda

- About me
- About this course
  - Learning objectives
  - Other similar courses
  - Course components
- Our learning model
- Tools and resources
  - Course Website
  - Ed
- Assessment and grading
- Collaboration

Learning Objectives

or, "What will I learn in this class?"

- **Functionality/Behavior:** Write functionally correct Java programs that meet a provided specification and/or solve a specified problem
- **Functional Decomposition:** Break down problems into subproblems that are modular and reusable, and define methods to represent those subproblems
- **Control Structures:** Select and apply control structures (e.g. methods, loops, conditionals) to manage the flow of control and information in programs
- **Data Abstraction:** Select and apply basic data abstractions (e.g. variables, parameters, arrays, classes) to manage and manipulate data in programs
- **Code Quality:** Define programs that are well-written, readable, maintainable, and conform to established standards

Other Similar Courses

Course	Good choice if…
CSE 121	• You've never programmed before AND • You are, or want to be, in a major such as CS, CE, ECE, Info, etc. that requires Java programming
CSE 122	• You've done some programming (roughly one course worth) in any programming language AND • You are, or want to be, in a major such as CS, CE, ECE, Info, etc. that requires Java programming
CSE 123	• You've taken CSE 122 AND • You are, or want to be, in a major such as CS, CE, ECE, Info, etc. that requires Java programming
CSE 160	• You've never programmed before AND • You're interested in data science and analysis OR • You'd rather learn Python than Java* OR • You are, or want to be, in a major such as Physics, Bio, Stat, etc. where analyzing data through programming is useful

Other courses of interest: CSE 154, CSE 163

See Guided Self-Placement and Introductory Courses for more info

Course Components

Meetings

LECTURES (x20)

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

SECTIONS (x18)

- Held in person
- More practice, reviews, applications
- TA advice, how to be an effective student
- Preparation for quizzes / exams

Assessments

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 computer
- One retake per quiz

EXAM (x1)

- Culminating exam
- **Thursday, June 8 2:30pm-4:20pm *tentatively***

Agenda

- About me
- About this course
  - Learning objectives
  - Other similar courses
  - Course components
- Our learning model ⟵
- Tools and resources
  - Course Website
  - Ed
- Assessment and grading
- Collaboration

How Learning Works

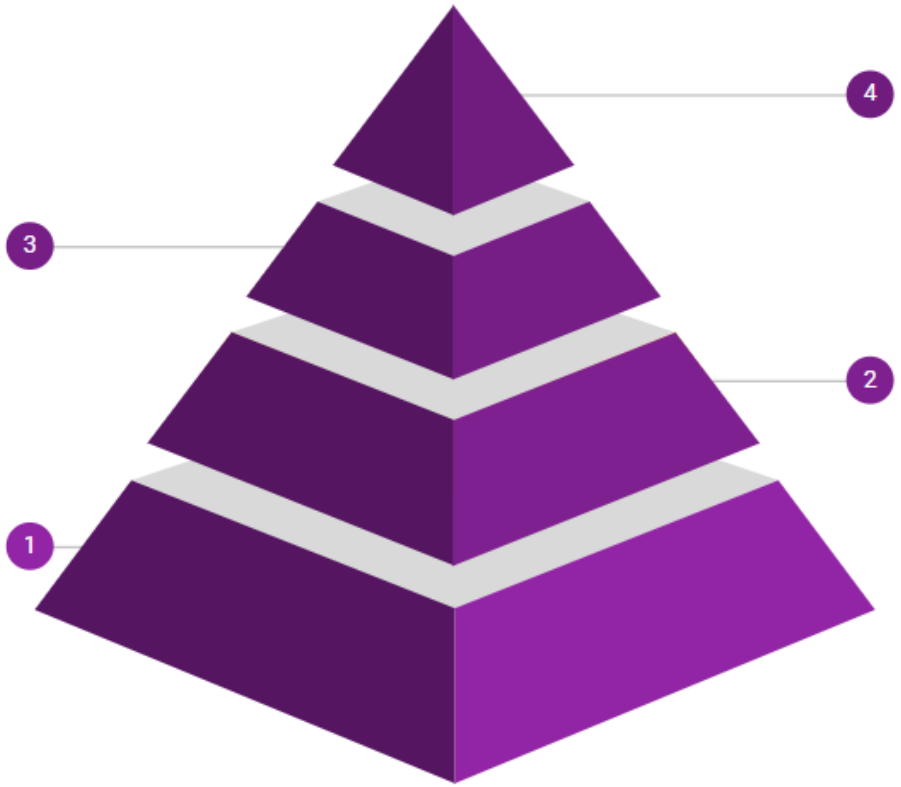
- Learning requires **active participation** in the process. It's not as simple as sitting and listening to someone talk at you.
  - Requires **deliberate practice** in **learning by doing**
  - Benefits from **collaborative learning**
- Hybrid classroom model
  - Asks you to do some preparation before class in the form of readings and practice problems.
    - Should take ~30 minutes a day
  - 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!

Metacognition

- **Metacognition**: asking questions about your solution process.

- Examples:
  - **While debugging**: explain to yourself why you're making this change to your program.
  - **Before running your program**: make an explicit prediction of what you expect to see.
  - **When coding**: be aware when you're not making progress, so you can take a break or try a different strategy.
  - **When designing**:
    - Explain the tradeoffs with using a different data structure or algorithm.
    - If one or more requirements change, how would the solution change as a result?
    - Reflect on how you ruled out alternative ideas along the way to a solution.
  - **When studying**: what is the relationship of this topic to other ideas in the course?

Learning in CSE 121 (or anywhere)



Assessment
Take-home assessments

Build on the scaffolding from 1-3.
Still learning by doing: you're not done!
Demonstrate your mastery
(even if it's still developing).

Guided Practice
Lesson activities, sections, labs

Practice with support from course staff.
Learn by doing: make mistakes and learn from them.
Start to develop mastery.

Independent/Group Practice
Checkpoints, section problems, additional practice

Practice on your own or with classmates.
Continue to **learn by doing.**
Get close to mastery.

Exposure
Lessons, videos, textbook

Encounter concepts for the first time.
See examples and ask questions.
Nowhere near mastery!

Course Culture and Support

- Currently 400 students enrolled!
  - Wide range of backgrounds, interests, and goals
  - **Everyone** is new to programming
- Support and help each other!
  - Form study groups
  - If you have a question, others almost certainly do too

Course Culture and Support: 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
- Email
  - We prefer that all content and logistic questions go on the Ed discussion board (even if you make them private). 400 of you >>> 24 of us!
  - For serious personal circumstances, you can email Miya directly. It never hurts to email me, but if it's a common logistic question, I may politely ask you to post on the discussion board.

Course Culture and Support

- Policies designed with flexibility in mind
  - Resubmissions/Retakes, lecture recordings

- But life and the world still happen…

- ***Please reach out ASAP*** *if you're struggling or have circumstances that require extra support*

The World Around CSE 121

- Our goal is to give you a great CSE 121 experience
  - But CSE 121 does not exist in a vacuum – there's a lot going on in the world right now that can impact your education
- We've designed course policies for maximum flexibility: ability to resubmit assignments and retake quizzes
  - But we cannot cover every individual situation

- **Please reach out** if you need accommodations of any kind to deal with these unfamiliar situations

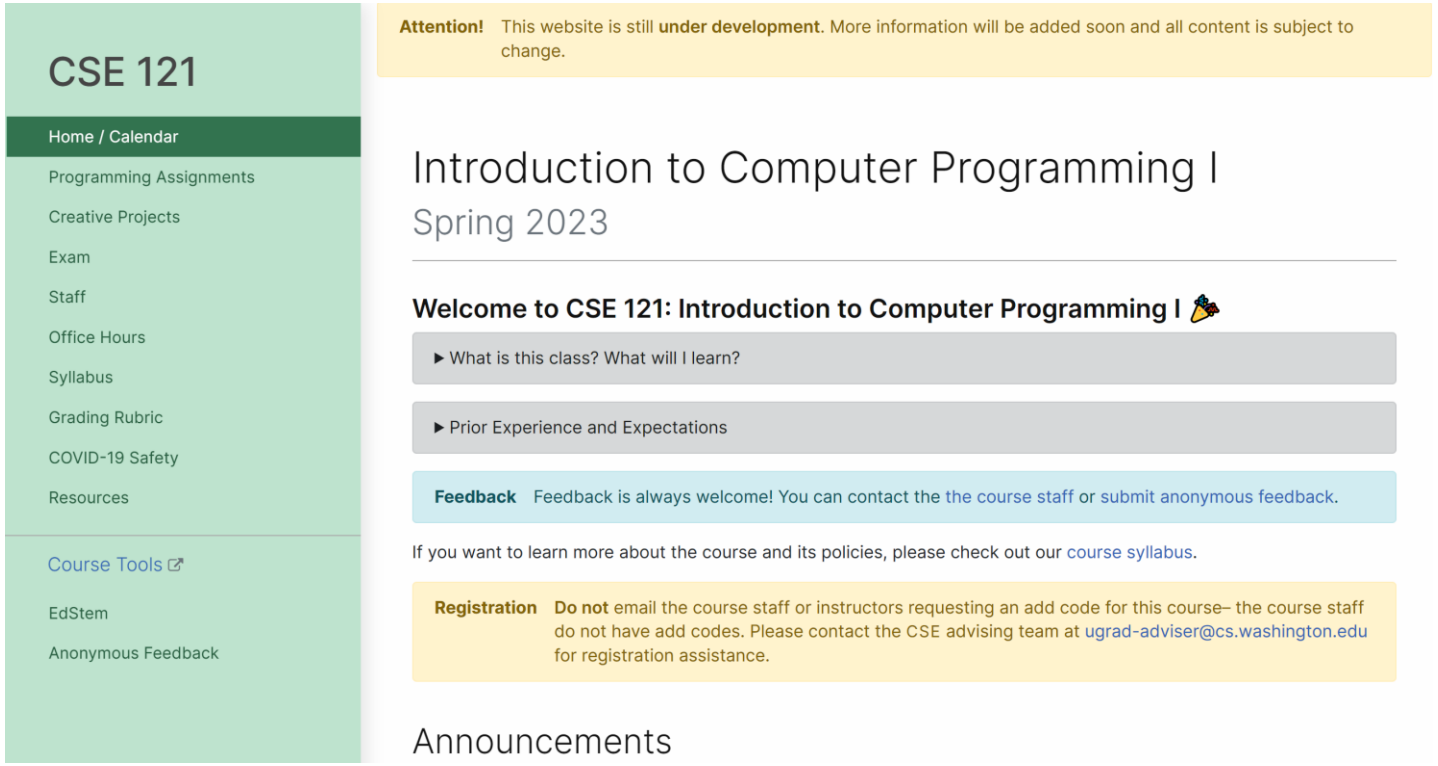
Agenda

- About me
- About this course
  - Learning objectives
  - Other similar courses
  - Course components
- Our learning model
- Tools and resources ⟵
  - Course Website
  - Ed
- Assessment and grading
- Collaboration

Course Website

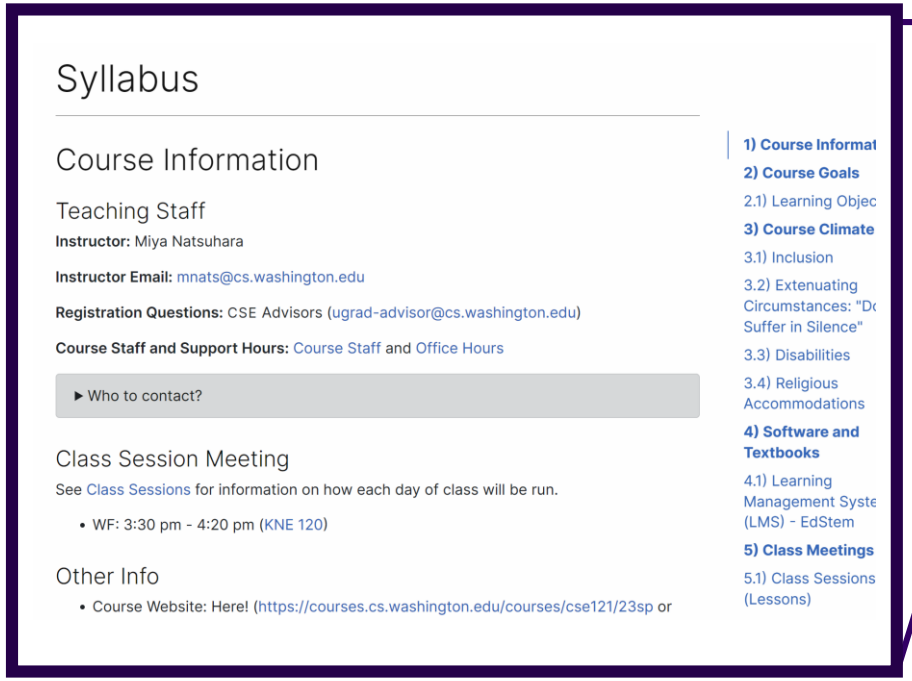
cs.uw.edu/121

- Primary source of course information (*not* Canvas)
- Calendar will contain links to (almost) all resources



Course Website

Please review the syllabus ASAP.



Syllabus

Course Information

Teaching Staff

Instructor: Miya Natsuhara

Instructor Email: mnats@cs.washington.edu

Registration Questions: CSE Advisors (ugrad-advisor@cs.washington.edu)

Course Staff and Support Hours: Course Staff and Office Hours

▸ Who to contact?

Class Session Meeting

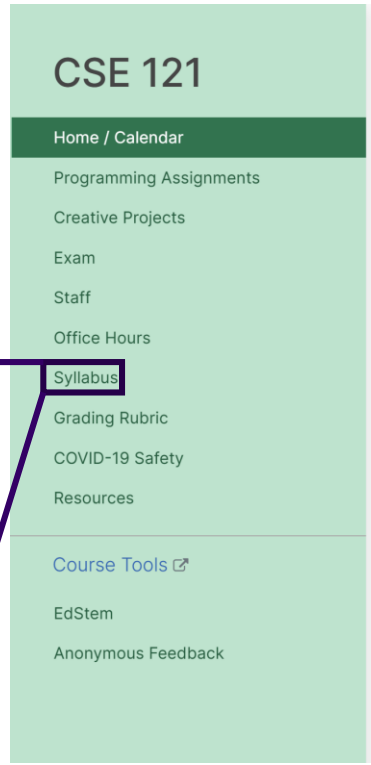
See Class Sessions for information on how each day of class will be run.

- WF: 3:30 pm - 4:20 pm (KNE 120)

Other Info

- Course Website: Here! (https://courses.cs.washington.edu/courses/cse121/23sp or

1) Course Informat
2) Course Goals
2.1) Learning Objec
3) Course Climate
3.1) Inclusion
3.2) Extenuating Circumstances: "D Suffer in Silence"
3.3) Disabilities
3.4) Religious Accommodations
4) Software and Textbooks
4.1) Learning Management Syste (LMS) - EdStem
5) Class Meetings
5.1) Class Sessions (Lessons)

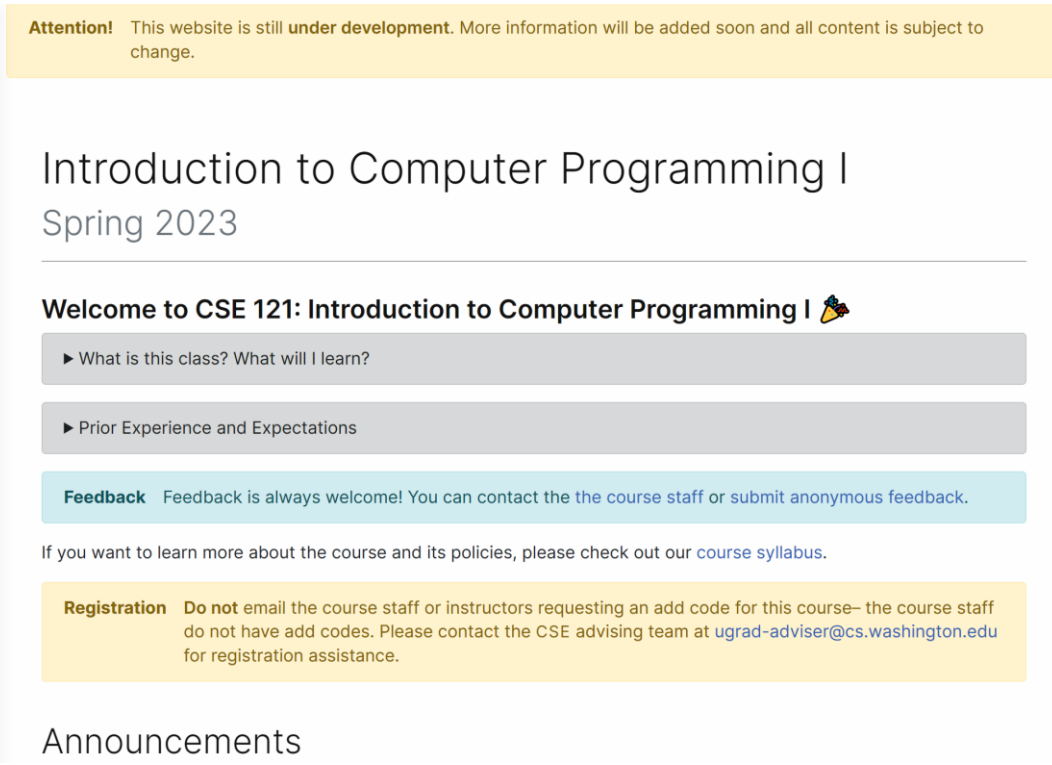


CSE 121

Home / Calendar
Programming Assignments
Creative Projects
Exam
Staff
Office Hours
Syllabus
Grading Rubric
COVID-19 Safety
Resources

Course Tools

EdStem
Anonymous Feedback



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

Introduction to Computer Programming I

Spring 2023

Welcome to CSE 121: Introduction to Computer Programming I

▸ What is this class? What will I learn?

▸ Prior Experience and Expectations

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

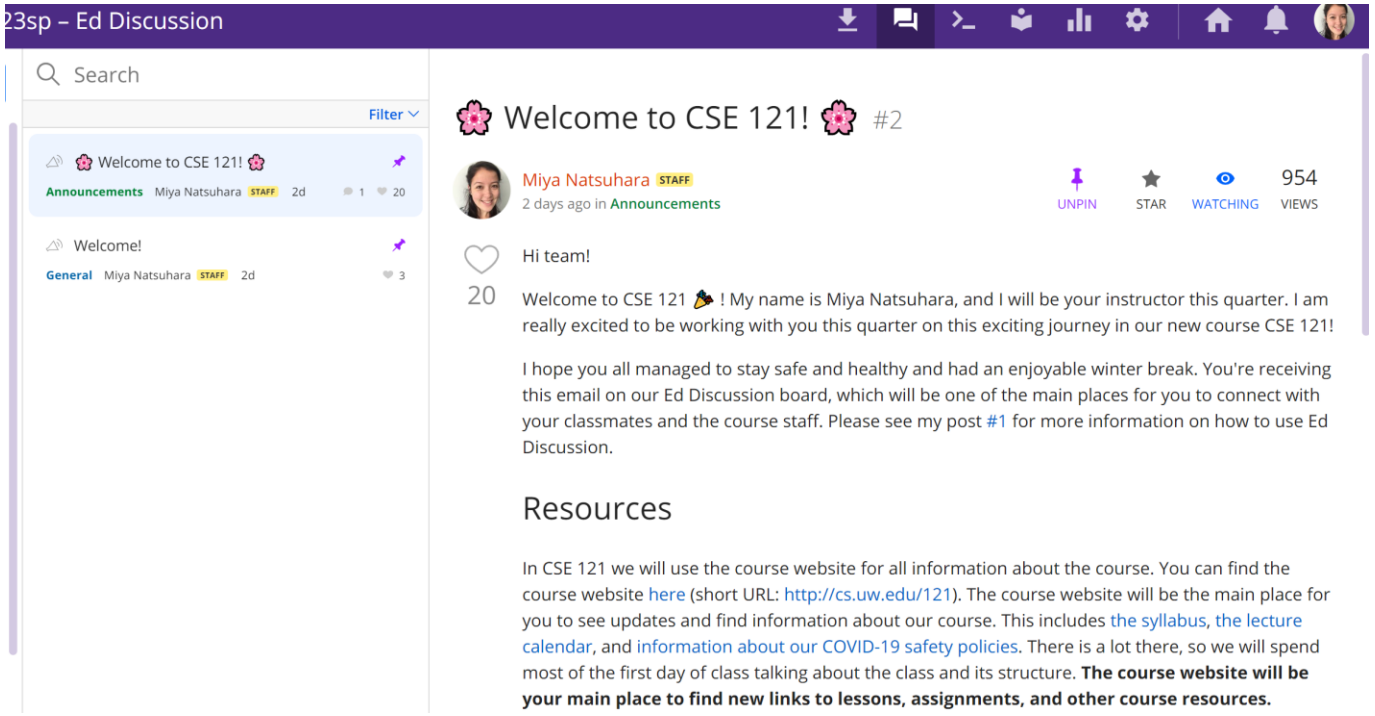
If you want to learn more about the course and its policies, please check out our course syllabus.

Registration **Do not** email the course staff or instructors requesting an add code for this course– the course staff do not have add codes. Please contact the CSE advising team at ugrad-adviser@cs.washington.edu for registration assistance.

Announcements

Ed

- Our online learning platform
- Lessons, sections, quizzes all here
- Intro and walkthrough in Section 0



Other Course Tools

My Digital Hand
- Queueing in office hours

Canvas
- Gradebook
- Lecture recordings

IntelliJ
- Develop offline
- Visual debugger

Sli.do
- In-class activities (ungraded)
- No account needed

Agenda

- About me
- About this course
  - Learning objectives
  - Other similar courses
  - Course components
- Our learning model
- Tools and resources
  - Course Website
  - Ed
- Assessment and grading ⟵
- Collaboration

Assessment and Grading

- Our goal in the course is for you to **gain proficiency of 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

Resubmission/Retakes

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

- Each week, one previous Programming Assignment or Creative Project can be resubmitted
  - Must be accompanied by write up explaining changes.
  - Grade on resubmission replaces original grade.
  - An assignment is only eligible for resubmission within 3 weeks of its original due date.
- Each quiz can be retaken at most once
  - A quiz is only eligible for retake within 3 weeks of its original date.

See 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

Agenda

- About me
- About this course
  - Learning objectives
  - Other similar courses
  - Course components
- Our learning model
- Tools and resources
  - Course Website
  - Ed
- Assessment and grading
- Collaboration ⟵

Collaboration Policy

Learning is hard, but it's easier when you learn from each other

- You are encouraged to form study groups, work together on practice and review, and discuss your ideas and approaches **at a high level**
  - In general, share ideas and work together, but don't copy work. Never send someone else your code or solution write up.
- If you discuss your ideas with others, you must **cite them**
- All work you submit for grading **must be *predominantly* and *substantially* your own**
- Withdrawal policy

- See the syllabus for more details

Help Us Improve!

- CSE 121 is ***super new!***
- We worked hard to build a course we think will be effective and supportive and help you succeed
- We probably didn't get it all right

- We appreciate your patience and understanding if we need to make adjustments during the quarter
- Please give us lots of feedback!
  - Post on Ed and/or use the Anonymous Feedback Tool

“Homework” for Next Time

- First assignment will be released Friday, but there are some things to do in the mean time.

- TODO this week
  - Fill out the introductory survey
  - Post an introduction video on your section’s Ed thread! 😊
  - Go meet your TA and classmates in Thursday’s quiz section
  - ☆ Complete the pre-class material for Friday (see calendar)
  - Check over syllabus details