Welcome to CSE 142B!

Miya Natsuhara

Spring 2022

We’re so excited you’re here!

Agenda

- About us
- What is Computer Science?
- About this course
  - Learning objectives
  - Other similar courses
  - Course components
- Our learning model
- Tools and resources
  - Course Website
  - Ed
  - PollEverywhere
  - Discord

(Wed) →

- Assessment and grading
- Collaboration

Agenda

- About us ←
- What is Computer Science?
- About this course
  - Learning objectives
  - Other similar courses
  - Course components
- Our learning model

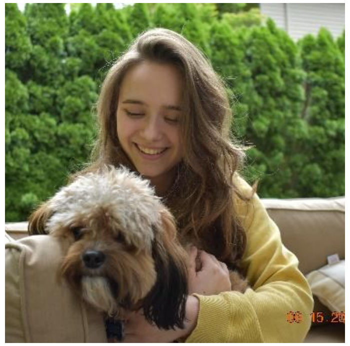
(Wed) →

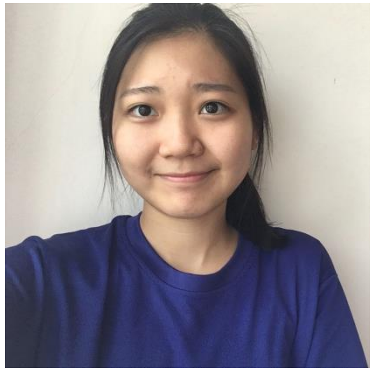
- Tools and resources
  - Course Website
  - Ed
  - PollEverywhere
  - Discord
- Assessment and grading
- Collaboration

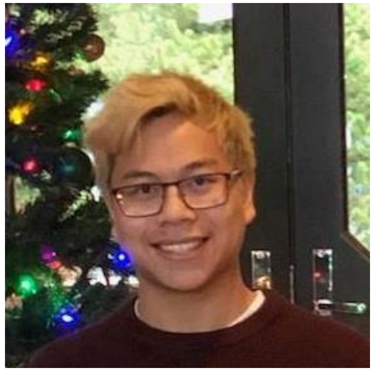
Hi, I’m Miya! (she/her)

- Instructional Lecturer
- Former Software Engineer at Microsoft
- B.A. in Mathematics and B.S. in Computer Science, Masters degree in Computer Science from UW
- Previously…
  - Frequent TA and Head TA for CSE 142
  - Lecturer for CSE 142

Meet your TAs







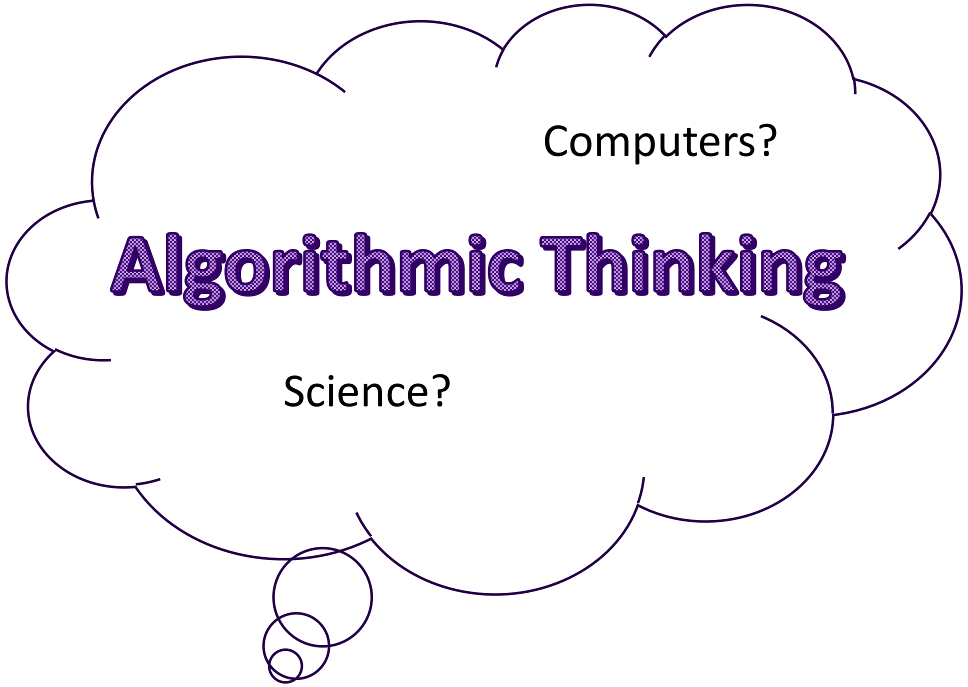
Agenda

- About us
- What is Computer Science?
- About this course
  - Learning objectives
  - Other similar courses
  - Course components
- Our learning model

(Wed)

- Tools and resources
  - Course Website
  - Ed
  - PollEverywhere
  - Discord
- Assessment and grading
- Collaboration

What is Computer Science?



Computers?

Algorithmic Thinking

Science?

Algorithm:

a step-by-step procedure for solving a problem or accomplishing some end *especially by a computer*

Programming is like a building block

Agenda

- About us
- What is Computer Science?
- About this course ←
  - Learning objectives
  - Other similar courses
  - Course components
- Our learning model

- Tools and resources
  - Course Website
  - Ed
  - PollEverywhere
  - Discord

(Wed) →

- Assessment and grading
- Collaboration

CSE 142A vs. CSE 142B

- Distinct courses with different course structures, instructors, and assessments
    - CSE 142A with Stuart Reges at 12:30pm
    - CSE142B with Miya Natsuhara (me!) at 8:30am
- **Not** interchangeable – make sure you are paying attention to the course you are registered for!

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 142	• You've never programmed before OR • You've done a little programming but feel rusty or not confident AND • You are, or want to be, in a major such as CS, CE, EE, Info, etc. that requires Java programming
CSE 143	• You've programming in Java before OR • You took AP CS A or IB CS in high school
CSE 143X	• You've programmed a lot before *in a language other than Java* OR • You are confident you can pick up new concepts very quickly OR • You *really, really* need to get through two courses in one quarter
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

Course Components

Lessons (aka Lectures)

- MWF, 8:30am PDT
- Held live in person; recordings released after
- First introductions to course concepts
- Mix of presentation of content and practice activities/problems
- Some required pre-work

Sections

- Th, various times
- Led by TAs
- Held live in person; ***not*** recorded
  - Short videos will be released on occasion when important material is covered
- Additional review, discussion, and practice
- Mostly practice problems

Attendance is not taken, but you are responsible for all material (including announcements).

Course Components

Labs (optional)

- T, various times
- Problems released online, support from TAs in person
- CSE 190 sections ***I through M and P through Q*** *(pending confirmation)*
- 1 credit course
- Credit/No Credit grading

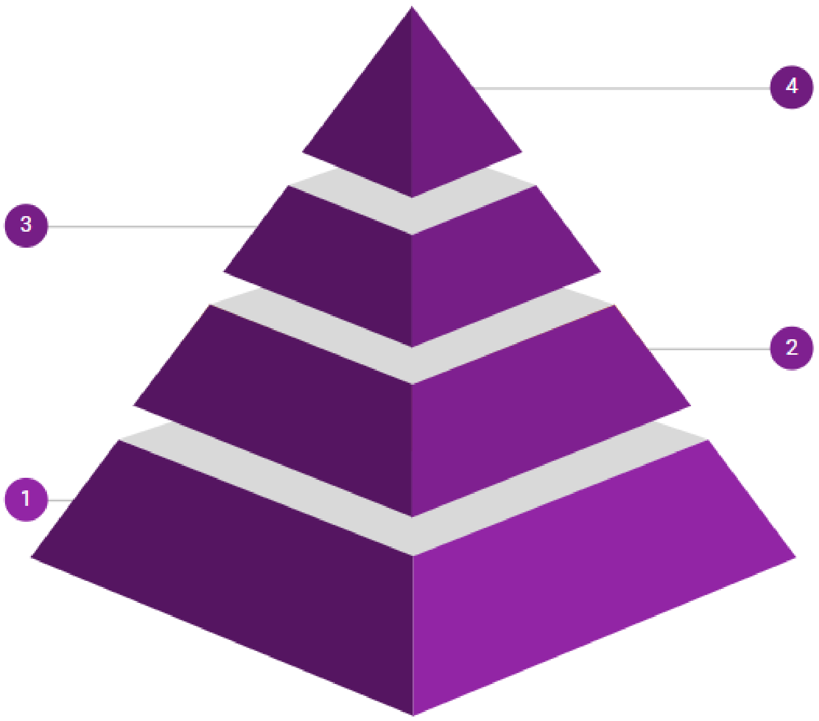
Agenda

- About us
- What is Computer Science?
- About this course
  - Learning objectives
  - Other similar courses
  - Course components
- Our learning model ←
- Tools and resources
  - Course Website
  - Ed
  - PollEverywhere
  - Discord
- Assessment and grading
- Collaboration

← (Wed)

FLYING PIGEON

Learning in CSE 142 (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).

Independent/Group Practice
Checkpoints, section problems, additional practice

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

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.

Exposure
Lessons, videos, textbook

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

Agenda

- About us
- What is Computer Science?
- About this course
  - Learning objectives
  - Other similar courses
  - Course components
- Our learning model
- Tools and resources
  - Course Website
  - Ed
  - PollEverywhere
  - Discord

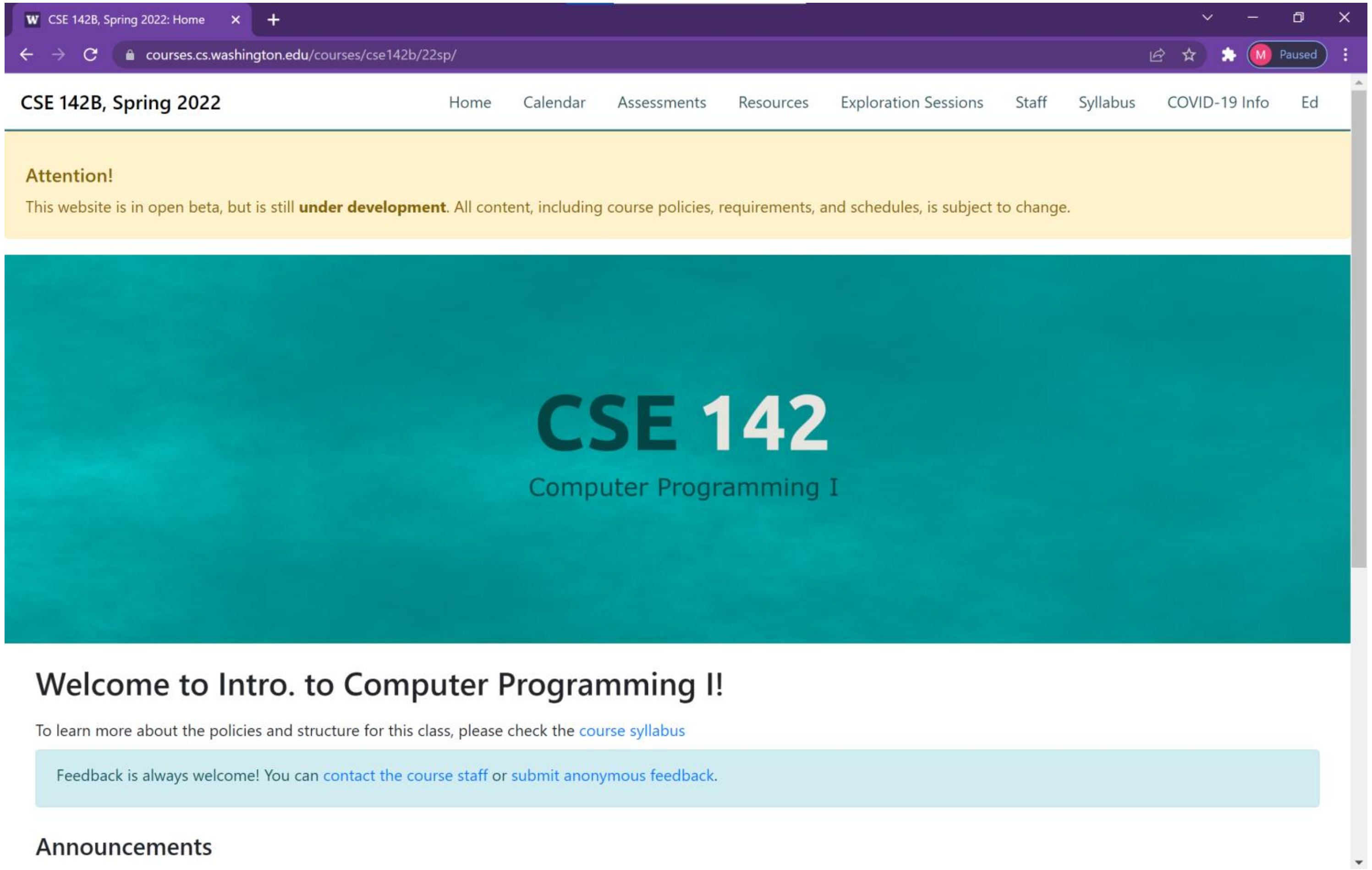
(Wed)

- Assessment and grading
- Collaboration

Course Website

cs.uw.edu/142b

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



Attention!

This website is in open beta, but is still **under development**. All content, including course policies, requirements, and schedules, is subject to change.

Welcome to Intro. to Computer Programming I!

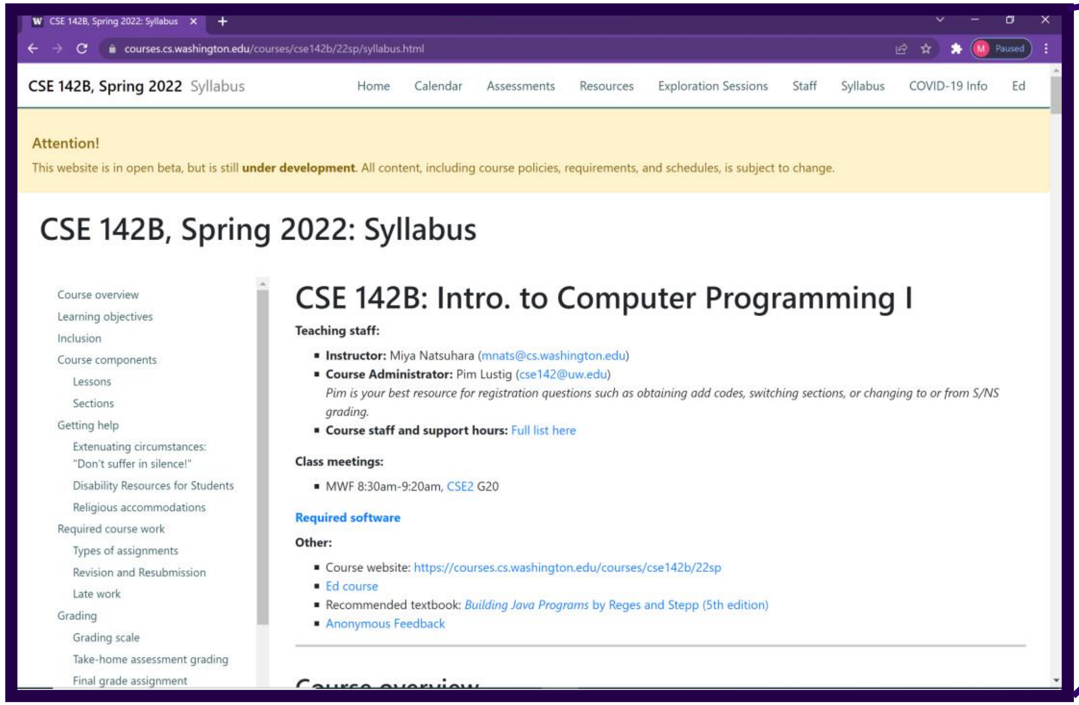
To learn more about the policies and structure for this class, please check the course syllabus

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

Announcements

Course Website

Please review the syllabus ASAP.


CSE 142B, Spring 2022: Syllabus
courses.cs.washington.edu/courses/cse142b/22sp/syllabus.html
Paused
CSE 142B, Spring 2022 Syllabus
Home
Calendar
Assessments
Resources
Exploration Sessions
Staff
Syllabus
COVID-19 Info
Ed

Attention!

This website is in open beta, but is still **under development**. All content, including course policies, requirements, and schedules, is subject to change.

CSE 142B, Spring 2022: Syllabus

CSE 142B: Intro. to Computer Programming I

Teaching staff:

- **Instructor:** Miya Natsuhara (mnats@cs.washington.edu)
- **Course Administrator:** Pim Lustig (cse142@uw.edu)
  Pim is your best resource for registration questions such as obtaining add codes, switching sections, or changing to or from S/NS grading.
- **Course staff and support hours:** Full list here

Class meetings:

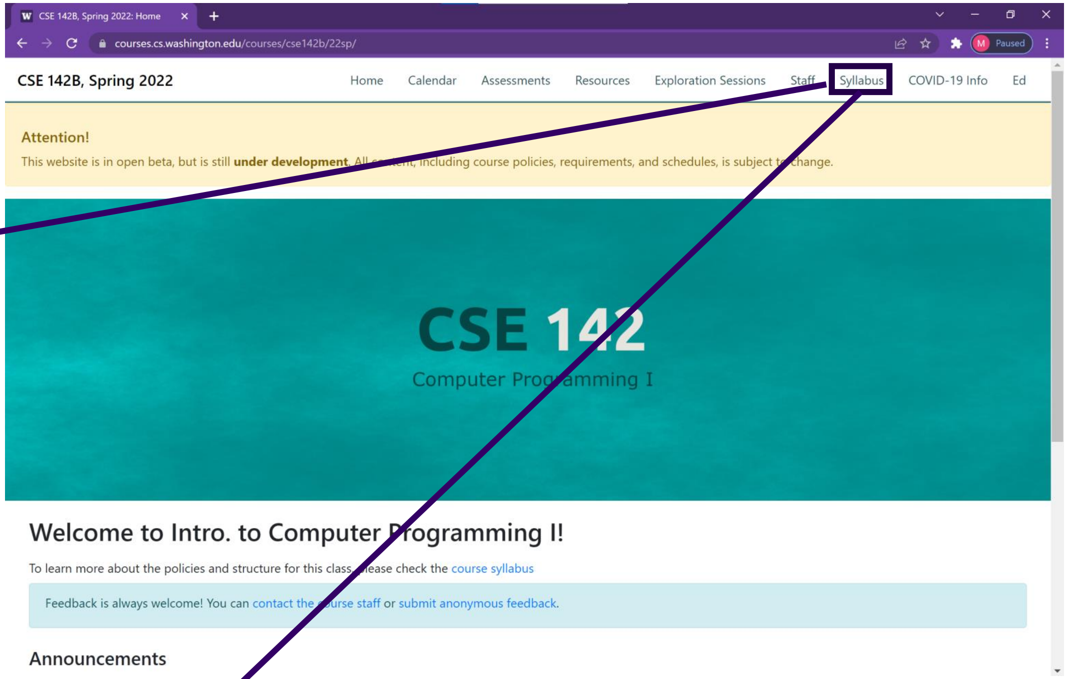
- MWF 8:30am-9:20am, CSE2 G20

Required software

Other:

- Course website: https://courses.cs.washington.edu/courses/cse142b/22sp
- Ed course
- Recommended textbook: *Building Java Programs* by Reges and Stepp (5th edition)
- Anonymous Feedback

Course overview



Attention!

This website is in open beta, but is still **under development**. All content, including course policies, requirements, and schedules, is subject to change.

Welcome to Intro. to Computer Programming I!

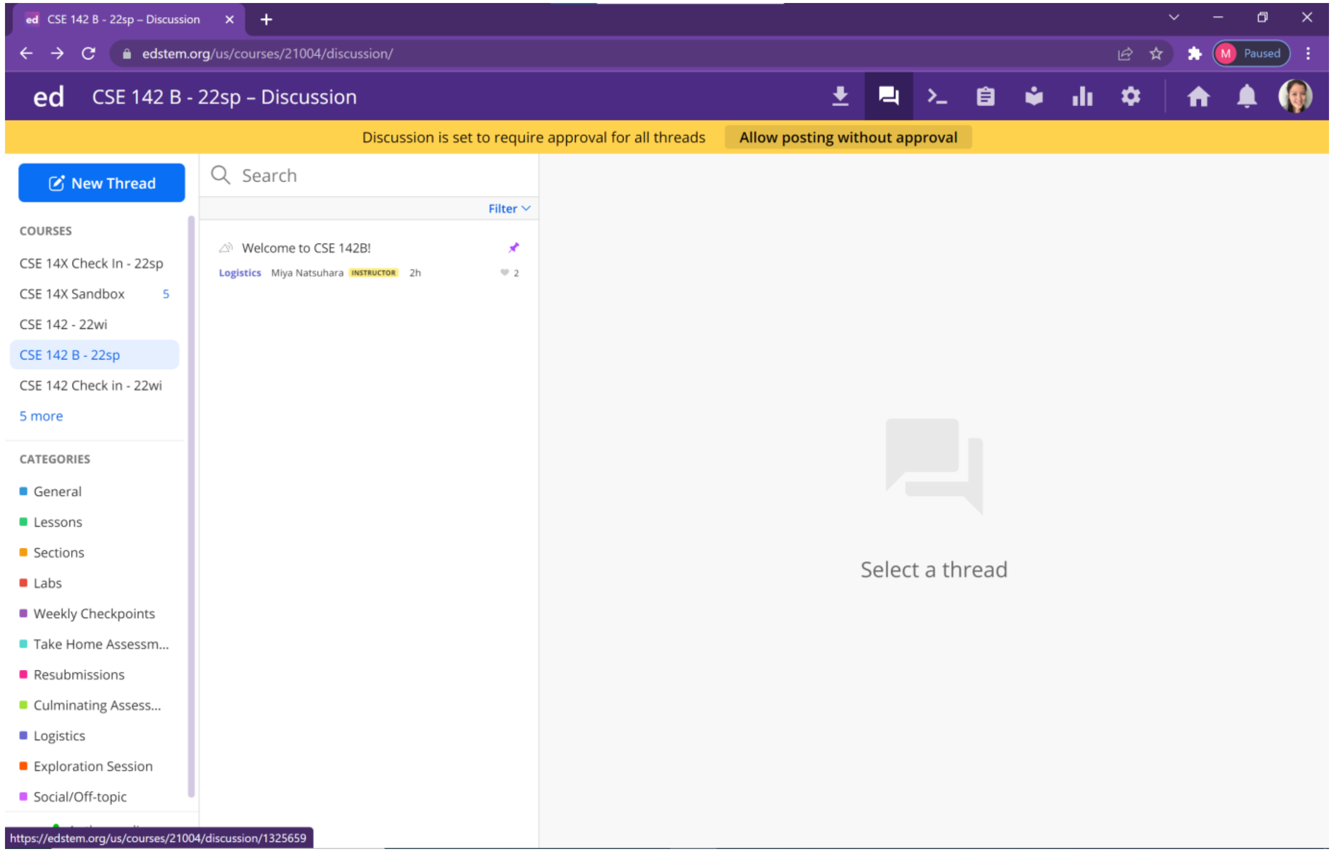
To learn more about the policies and structure for this class, please check the course syllabus

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

Announcements

Ed

- Our online learning platform
- Lessons, sections, labs, assessments all here
- Lecture Megathreads for each day that will be monitored by TAs during lecture
- Intro and walkthrough video forthcoming



edstem.org/us/courses/21004/discussion/

CSE 142 B - 22sp – Discussion

Discussion is set to require approval for all threads

Welcome to CSE 142B!

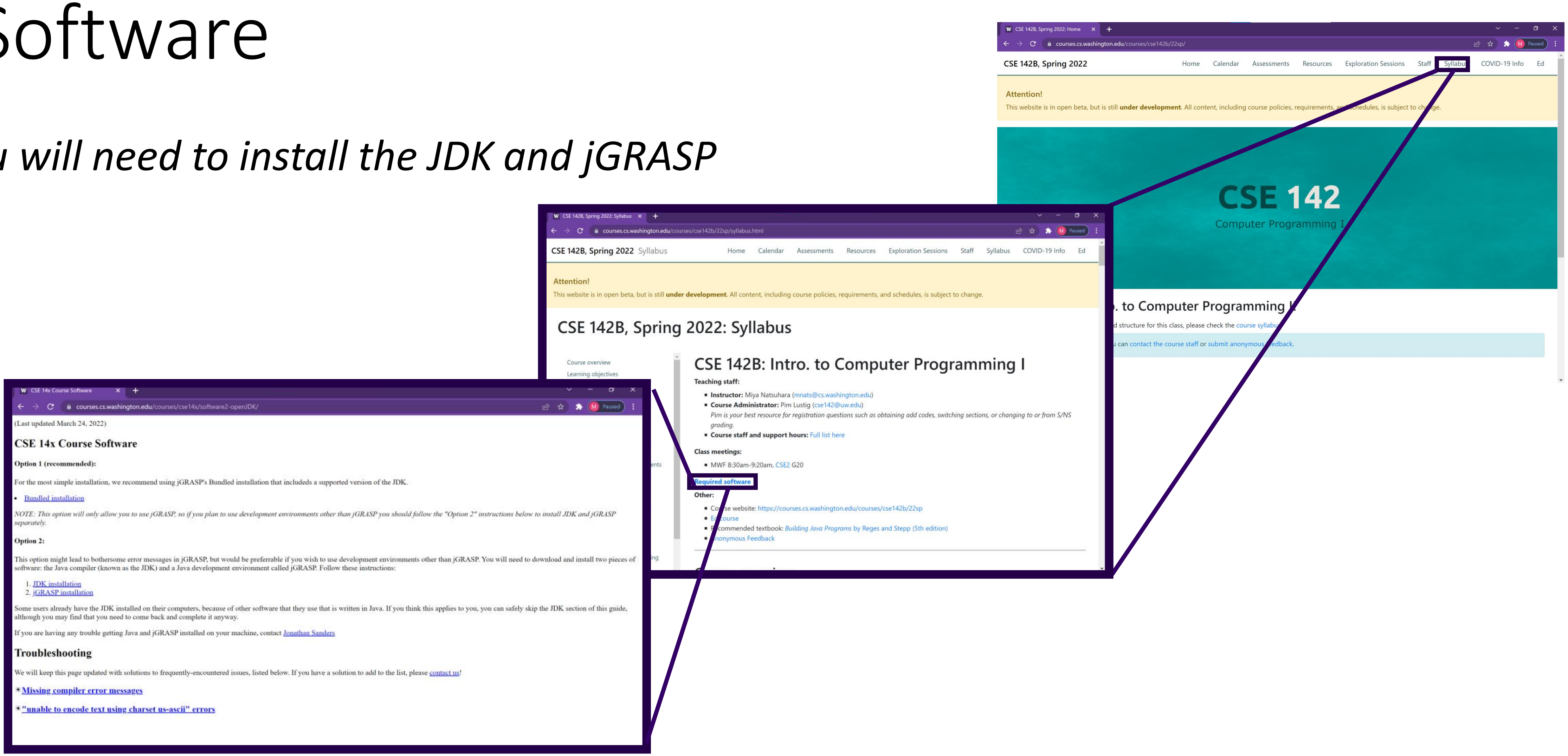
Logistics Miya Natsuhara INSTRUCTOR 2h 2

Select a thread

https://edstem.org/us/courses/21004/discussion/1325659

Software

You will need to install the JDK and jGRASP



Attention!

This website is in open beta, but is still **under development**. All content, including course policies, requirements, an schedules, is subject to change.

. to Computer Programming I

d structure for this class, please check the course syllabu

u can contact the course staff or submit anonymous edback.

CSE 142B, Spring 2022: Syllabus
courses.cs.washington.edu/courses/cse142b/22sp/syllabus.html
CSE 142B, Spring 2022 Syllabus
Home
Calendar
Assessments
Resources
Exploration Sessions
Staff
Syllabus
COVID-19 Info
Ed
Course overview
Learning objectives

Attention!

This website is in open beta, but is still **under development**. All content, including course policies, requirements, and schedules, is subject to change.

CSE 142B, Spring 2022: Syllabus

CSE 142B: Intro. to Computer Programming I

Teaching staff:

- **Instructor:** Miya Natsuhara (mnats@cs.washington.edu)
- **Course Administrator:** Pim Lustig (cse142@uw.edu)
  Pim is your best resource for registration questions such as obtaining add codes, switching sections, or changing to or from S/NS grading.
- **Course staff and support hours:** Full list here

Class meetings:

- MWF 8:30am-9:20am, CSE2 G20

Required software

Other:

- Co se website: https://courses.cs.washington.edu/courses/cse142b/22sp
- Ed course
- R commended textbook: *Building Java Programs* by Reges and Stepp (5th edition)
- nonymous Feedback

(Last updated March 24, 2022)

CSE 14x Course Software

Option 1 (recommended):

For the most simple installation, we recommend using jGRASP's Bundled installation that includeds a supported version of the JDK.

- Bundled installation

NOTE: This option will only allow you to use jGRASP, so if you plan to use development environments other than jGRASP you should follow the "Option 2" instructions below to install JDK and jGRASP separately.

Option 2:

This option might lead to bothersome error messages in jGRASP, but would be preferrable if you wish to use development environments other than jGRASP. You will need to download and install two pieces of software: the Java compiler (known as the JDK) and a Java development environment called jGRASP. Follow these instructions:

1. JDK installation
2. jGRASP installation

Some users already have the JDK installed on their computers, because of other software that they use that is written in Java. If you think this applies to you, you can safely skip the JDK section of this guide, although you may find that you need to come back and complete it anyway.

If you are having any trouble getting Java and jGRASP installed on your machine, contact Jonathan Sanders

Troubleshooting

We will keep this page updated with solutions to frequently-encountered issues, listed below. If you have a solution to add to the list, please contact us!

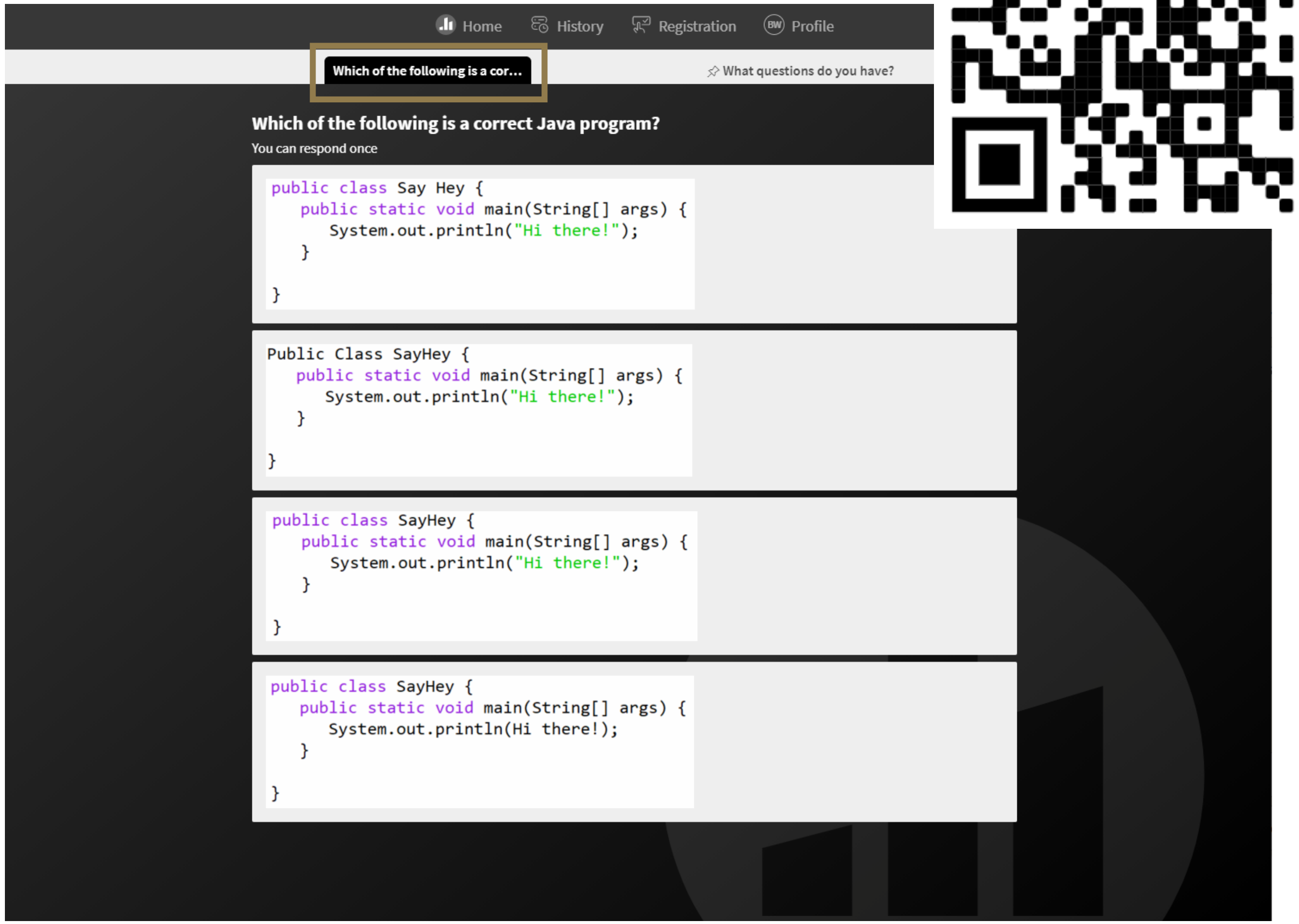
- **Missing compiler error messages**
- **"unable to encode text using charset us-ascii" errors**

PollEverywhere

pollev.com/mnats

Two purposes (at least):

- In-class activities
  - Short questions, problems, etc.
  - Usually multiple choice
  - *Not* graded
    - Not even on participation



Which of the following is a correct Java program?

You can respond once

```
public class Say Hey {
   public static void main(String[] args) {
      System.out.println("Hi there!");
   }

}
```

```
Public Class SayHey {
   public static void main(String[] args) {
      System.out.println("Hi there!");
   }

}
```

```
public class SayHey {
   public static void main(String[] args) {
      System.out.println("Hi there!");
   }

}
```

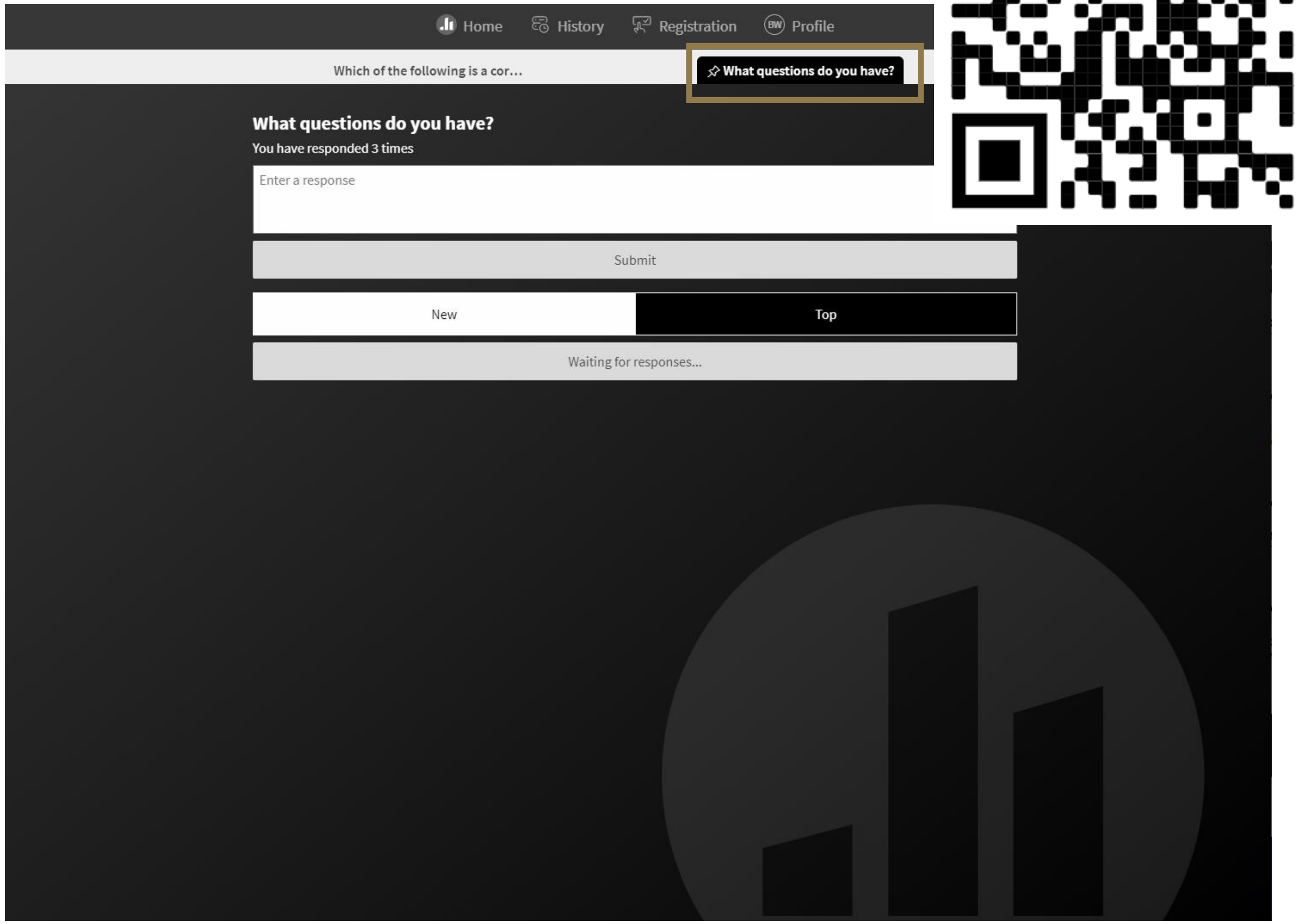
```
public class SayHey {
   public static void main(String[] args) {
      System.out.println(Hi there!);
   }

}
```

PollEverywhere

pollev.com/mnats

Two purposes (at least):

- Questions backchannel
  - Ask questions at any time
  - I'll check periodically and respond
  - Some may be deferred
    - Answers will be posted on Ed after class



What questions do you have?

You have responded 3 times

Agenda

- About us
- What is Computer Science?
- About this course
  - Learning objectives
  - Other similar courses
  - Course components
- Our learning model
- Tools and resources
  - Course Website
  - Ed
  - PollEverywhere
  - Discord
- Assessment and grading ←
- Collaboration

Assessment and Grading

- Our goal in the course is for you to **master the concepts and skills** we teach
- We assess your mastery 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 *mastery*
- Your final grade should reflect **the extent to which you have demonstrated mastery of the course objectives**

Assessment

- Your learning in this course will be assessed in four ways:
  - Checkpoints (~weekly)
    - Short problems to help you practice and make sure you've got the basics for the week
  - **Take-home assessments (~weekly)**
    - **Large programming assignments to assess your full mastery of that week's concepts (plus some previous material)**
  - Culminating assessments (2/quarter)
    - Series of problems covering all material up to that point
  - Reflections (w/other assignments)
    - Written assignments to help you think critically about your learning and progress

Resubmission

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

- One previous take-home assessment can be **resubmitted** each week
  - Must be accompanied by a write-up describing changes
  - Grade on resubmission will replace original grade

- See the syllabus for more details

Grading

Grades should reflect your mastery of the course objectives

- Checkpoints, culminating assessments, and reflections are graded **S (Satisfactory) or U (Unassessable)**
  - If you submit on time and meet all requirements, you'll get an S
- Take-home assessments will be grade **E (Exemplary), S (Satisfactory), N (Not yet), or U (Unassessable)** on four dimensions:
  - Behavior
  - Functional decomposition
  - Use of Language Features
  - Code Quality
- Final grades will be assigned based on the **amount of work at each level**

- See the syllabus for more details

Agenda

- About us
- What is Computer Science?
- About this course
  - Learning objectives
  - Other similar courses
  - Course components
- Our learning model
- Tools and resources
  - Course Website
  - Ed
  - PollEverywhere
  - Discord
- 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**
- If you discuss your ideas with others, you must **cite them**
- All work you submit for grading **must be your own**
- Any work found to not be your own will receive a grade of **U and may not be resubmitted**
  - If it's not your work, we can't assess your mastery from it

- See the syllabus for more details

Amnesty

Sometimes, we make bad choices that we regret

- “If you submit work that is in violation of the academic conduct policy, you bring the action to Miya’s attention within 72 hours of submission and request amnesty. If you do so, you will receive a grade of U for the initial submission, but you **will be allowed to resubmit your work under the normal resubmission process**.”
- See the syllabus for more details