

Welcome to CSE 160: Data Programming

Summer 2026



W PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING



Lecture Overview

- Introductions!
- Who is in this class?
- About this Course
 - Course Components & Tools
 - Grading
 - Policies
 - Making the Most of this Class
- Introduction to Python

Course Instructor: Adrian Salguero

- Born and raised in Los Angeles, CA
- Bachelors in Computer Science from UC Santa Cruz
- PhD in Computer Science from UC San Diego
 - Research focus in Computer Science Education
- Office Location: [CSE 442](#)
- Some things I enjoy doing: weightlifting, running, video games, reading comics, and baseball.



Your Amazing Course TAs! 🎉



Varun Atray



Daniella Maor



Kellen Rodriguez



Guy Zur

How many of you...



...have taken a CS
programming course
anywhere?



...have written code in
Java, C, C++, Python, or
any other similar
language?



...have written code on
your own to create some
awesome projects?



What is this Class?

- Python fundamentals
 - Variables & Expressions
 - Loops
 - Lists
 - Conditionals (If-statements)
 - File I/O
 - Dictionaries
 - Nested Structures
- Data Visualization
- Python Classes and Objects
- A tiny bit of "Data Science" tools (e.g. Jupyter Notebook and Pandas)

*~60% of the course is
going to focus on
Python fundamentals*

Prerequisite Knowledge

- There are no prerequisites for this course!
- We do assume that you are familiar with high school math and basic computer usage (e.g. running apps, editing files, etc.)
- We assume everyone has ***zero** prior programming experience*
- **If you have already taken CSE 143 or CSE 123, you cannot receive credit for this course!**
 - I encourage you to enroll in [CSE 163](#) instead.
- **If you've taken CSE 122**, you can still receive credit but in my opinion you are too advanced for this course.
- If you already have extensive knowledge in Python or another programming language (such as Java or C), you should take [CSE 163!](#)

Course Website

cs.uw.edu/160

We do NOT use Canvas in this class

Course Tools



Ed

- Announcements
- Q&A



JupyterHub

- Coding environment
- No need to install anything!



Anaconda/VS Code

- [Optional] Coding environment



Gradescope

- Assignment submissions, grades, and feedback

Grade Breakdown

Category	Weight
Homework	30%
Midterm Exam	30%
Final Exam	30%
Participation (Written Check-ins + Coding Practice)	10%

Exam Grading Policy

Sometimes when we take an exam, we don't give it our best shot or we wish we could have done better. Therefore in this class has the following policy:

If your final exam score is higher than your midterm score

→ Your final exam score **replaces** your midterm score.

If your final exam score is lower or the same as your midterm score

→ Your midterm score **remains unchanged**.

Course Grades

The following are minimum GPA guarantees - grade can always (and are usually) higher than minimum guarantee 😊.

Example: You earn a 95% in the course, that doesn't mean you can only get a 3.5. This only means *you can't get anything lower than a 3.5*

Percent Earned	Course Grade
95 - 100%	≥ 3.5
85 - 95%	≥ 3.0
75 - 85%	≥ 2.5
65 - 75%	≥ 2.0
50 - 65%	≥ 0.7

Course Components

Lectures

- Introduce concepts, practice ideas, discuss applications
- Recorded 🎥

Written Check-Ins

- Started in section, due the day after section!
- Additional conceptual practice questions

Sections

- Held in person
- More practice, reviews, and **written check-ins**
- TA-led
- 🚨 **Not Recorded!** 🚨

Homework

- Programming in Python
- Structured Assignments
- Applying & Implementing course concepts

Coding Practice

- Meant to give you more practice with Python fundamentals
- Smaller programming questions
- Creative components

Exams

- One **midterm exam (July 20)**
- One **final exam (August 21)**



ONLY Homework assignments are eligible for resubmissions!

Resubmissions

Learning is a challenging process that takes time, it doesn't always happen on your first try!

At a certain point in the course, each week you will have the opportunity to resubmit one of your previous programming assignments.

- Grade on resubmission replaces original grade
- You cannot resubmit an assignment until you receive feedback (~ 1 week)
- Accompanied by a short write up of changes made (helps your TA 😊)
- An assignment found to involve academic misconduct **may not be resubmitted**

Collaboration

- Programming concepts can be challenging – we strongly encourage discussion and 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 solutions!**
 - Be aware of and avoid use of [Prohibited Actions](#) in submitted work.
 - ***Using any AI-assisted tools*** to write code
 - Looking at another person's submission or solution
 - Copying or using code copied from another resource (even with edits)
- Read full policy in the Syllabus

CSE 160 AI Policy

- No use of AI is allowed in CSE 160
 - Any work we suspect of using AI (for code or documentation) will be flagged as academic misconduct and result in a 0 grade.
 - Multiple misconduct flags will result in being reported to CSSC. More than 1 report will result in a 0 in the course.
 - ***Any assignment flagged for misconduct cannot be resubmitted.***

“You can choose to teach your kid to play the stereo, or play the piano”

– [Mitch Resnick](#), professor at MIT and founder of Scratch

CSE 160 Out-of-Scope Policy

- Any out-of-scope work on your assignments will result in lost points.
 - All your work on any assignment (e.g. homework, exams, programming practice, written check-ins) ***must only use material covered in class, unless otherwise specified.***
 - Assignments using out-of-scope content **can be resubmitted** for full credit.
 - Example: We do not cover *case* and *match* in this course, using either on any of your work counts as out-of-scope content use and will result in losing points.
- If you are relying heavily on out-of-scope content on your assignments, this course is likely too easy for you and I suggest enrolling in CSE 163.

How Learning Works

- Learning requires ***active participation*** in the process. It's not as simple as sitting and listening to someone talk about it.
 - Requires deliberate practice and learning by doing
 - Benefits from collaborative learning
- Active learning during lecture
 - Attendance isn't graded but showing up will help you stay on track
 - Active discussions and exercises in lecture
- Lecture attendance is not required but you will be responsible for staying on track with the material and announcements made in lecture.

Getting Help

- ***Discussion Board (Ed)***
 - Make a public or private post
 - We encourage others to answer others' questions!
- ***Section***
 - Smaller groups and face-to-face interaction with your TAs
- ***Office Hours***
 - Instructor or TA office hours! [Check out the calendar on the website.](#)
- ***Email***
 - Prefer any content and logistics questions go to Ed
 - For serious or personal circumstances, please email Adrian directly.
 - Please ask any grading related questions to your TAs or post on Ed! Adrian gets too much email already!