

CSS 1 – Introductory Programming for Computational Social Sciences

Spring 2025

Instructor: Catarina Roman

Office Hours: Thursdays 3:00 PM to 5:00 PM

Office Hours Sign-Up: <https://calendly.com/acroman/office-hours>

Office Hours Zoom Link:

<https://ucsd.zoom.us/j/95904398430?pwd=BbZaDSbZg0Lo7deHfbKjabVct657zw.1>

TA: Luna Bellitto

Email: lbellitto@ucsd.edu

Office Hours: TBD

Office Hours Sign-Up:

TA: Joyce Hu

Email: joh011@ucsd.edu

Office Hours: TBD

Office Hours Sign-Up: <https://calendly.com/joh011-ucsd/css1-officehours>

Objective

This course develops computational thinking practices and skills critical for defining, describing, and analyzing social science problems using a computational approach. Students will learn to program in Python in the context of computational social science problems.

Learning Outcomes

This class will teach you:

1. Design and implement Python programs.
2. Read, interpret, and debug Python code.
3. Execute Python programs in Jupyter Notebooks.
4. Teach you valuable skills to perform computational social sciences tasks and data analytics.

Location, hours, and credits

Some of our sessions in this class will occasionally take place on zoom. See below for the schedule – may change along the quarter, stay up to date on Canvas notifications!!

Zoom sessions will be recorded and posted to the Media Gallery on Canvas.

In-person

Location: Central 216
Hours: MWF 9:00-9:50 AM

On zoom

Zoom link: <https://ucsd.zoom.us/j/99803146755?pwd=yaRumCAdsbRH1nmTkZvMaSivZmWITH.1>

Sessions on Zoom:

Week 2: Monday, April 14th
Tentative: Week 5: Friday, May 2nd
Week 8: Friday, May 23rd (Memorial Day Weekend)

TA Lab Sessions:

Session A01:

Location: HSS 2154
Hours: M 10:00-10:50 AM

Session A02:

Location: HSS 2154
Hours: M 11:00-11:50 AM

Note: No TA sessions on Week 01. Labs start April 7th.

Credits: 4

Prerequisites

1. [Canvas](#) access
2. [DataHub](#) access
3. Curiosity and hard work

Books and Readings

I have one book suggestion:

1. [PP] Lutz, Mark. *Programming Python*. [O'Reilly Media](#), Inc., 2021 (or latest edition).

This book is available online for free using your UCSD account. I will not follow every detail in the book, but I highly recommend it.

Grading and Evaluations

Assignment	Weight	Due Date
Participation x 17	5 %	CANVAS quizzes

Coding Labs x 8	40 %	<p>Due on Mondays:</p> <p>Lab 1 - April 7th – Week 2 Lab 2 - April 14th – Week 3 Lab 3 - April 21st – Week 4 Lab 4 - April 28th – Week 5 Lab 5 - May 5th – Week 6 Lab 6 - May 12th – Week 7 Lab 7 - May 19th – Week 8 No Lab or class May 26th, Memorial Day Lab 8 - June 2nd – Week 10</p>
Problem Sets x 4	40 %	<p>Due on Fridays:</p> <p>PS 1 - April 25th – Week 4 PS 2 - May 9th – Week 6 PS 3 - May 23rd – Week 8 PS 4 - June 6th – Week 10</p> <p>All PSs will be live two weeks before their due date.</p>
Final Project	15 %	June 10th
Extra Credit	5 %	5% bonus to all students, if 90% of the students fill up the evaluation forms.
	105%	

Participation

Active participation is essential. The main form of participation is showing up. By showing up, you will ensure you are in the right place at the right time. This is mostly enough to guarantee that you will ask great questions and participate further.

Participation is computed using Canvas quizzes. Every class, I will put on the board a pin for the participation quiz of the day. This will start counting starting on April 7th.

You can miss four classes and still get full participation marks. And because of that, we will not manually input your participation.

Coding Lab

Labs are intended to give you hands-on programming experience. You will have one lab each week, designed for completion during the TA session.

Labs are collaborative.

You can (and should!) work with other classmates. In general, programming is often a team sport. You will find that accurate in software engineering and data science. Thus, labs are an excellent opportunity to practice collaboration with others.

Labs are graded for effort and correctness.

My lab grading method is:

$$70\% \times \text{Finished} + 30\% \times \text{Accuracy}$$

You will see only accuracy on nbgrader (the 30% portion of it). However, when the TAs transfer the grades to Canvas, they will add the 70% effort bonus.

Important: No TA sessions during the first week!

Problem-Sets

Problem sets are **bi-weekly** (as in one roughly every two weeks) programming exercises intended to give you more hands-on experience applying course concepts.

Problem sets are completed independently.

Unlike labs, problem sets should be completed **independently**. You can ask the TAs (or me) for help, but each answer should reflect your work.

Problem sets are graded for correctness.

Unlike labs, your grade on a problem set will reflect how many questions you answered **correctly**.

Note on auto-grading

Note that grading will be done using an **auto-grader**. This software will check your solutions to each problem and determine whether they are correct.

For the auto-grader to work correctly, it's imperative **not to edit or delete any notebook cells containing an assert statement.**

What does having a problem set, exam, or lab auto-graded mean? Auto-graders are pieces of code that help us determine if you got it right. They consist of public tests (you can see those) and private tests (you cannot see those). Your grade is computed by having the auto-grader score the correctness in the public and private tests. You can only see your final grade once we run the auto-grader. Your code may be working fine,

passing the public tests, but your grade might be lowered (even zero-ed sometimes) if you do not do what we ask. To make sure you get a good grade:

1. If the code asks you to create a variable, create the variable asked.
2. If the code asks you to assign a particular value or result to a variable, assign the result or value asked for to the variable.
3. If the code asks you to create a function, create the function asked.
4. If the code asks you that your function should have a given parameter, don't forget to have that parameter in it.
5. If unsure, test the code: read the problem, understand it, and think of tests that should be true if your code is good.

Pro-tip: After completing your assignment, restart the Vocareum kernel and rerun all the cells (all this will be clear later). This will tell if the code passes all the public tests, which is an excellent start to getting all the private tests correct.

Final Project

As part of the course, you must complete a **final project**.

You can think of the final project as a more extensive, more coherent problem set. Like the problem sets, you will be graded for correctness (and you should complete it independently). But unlike the problem sets, the questions in this final project will *build* towards a more significant conclusion—you will produce a small piece of working software and data analysis.

Goals

Like the labs and problem sets, the goal of this final project is to give you hands-on experience programming in Python.

Unlike the other assessments, the final project is intended to be a more **coherent** exercise.

The final project is completed independently.

Again, you can think of this as a more extensive problem set. It'll be graded for correctness, and you should work independently.

Late Policy

The final project must be submitted before the deadline. **No late submissions will be accepted.**

Grading Criteria

Read the question and make sure that you understand what is being asked. If you do not understand it, please email the TA, and they will gladly clarify. Once you know what must be done, ensure you do it most straightforwardly. You gain no extra points for making something simple into something convoluted and weird.

Requests for checking grades are welcome before the next assignment of the same type. For example, the TA may revise your problem set grade before your next problem set. After that, the grade is final. If you ask us to regrade your assignment, remember that your grade may go up, stay the same, or go down.

Late Policy

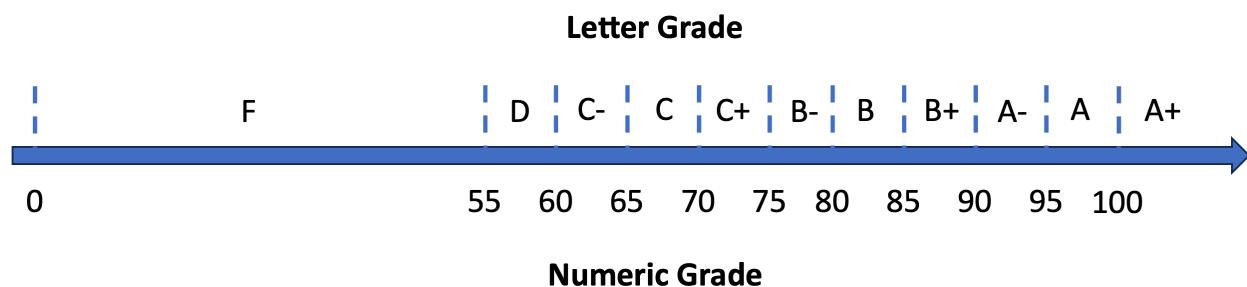
We understand that there may be circumstances where you may need more time to complete assignments, so the late penalty is designed to accommodate for these situations.

Labs and Problems Sets will incur a **10% late penalty** for each day that the assignment is submitted late.

No late submissions will be accepted for the final project, so make sure that you submit your project before the deadline.

Grading Scale

Your final letter grade is assigned using the following scale:



Note that the category above contains the endpoint. It means that 100 = A+, and not A; 95 = A, not A-; 90 = A- and not B+, 85 = B+ and not B, and so on.

There is no rounding. For example, a final grade of 89.99 is a B+, not an A-.

Academic Integrity

Here is the statement on Academic Integrity from the UCSD Academic Integrity Office:

"Academic Integrity is expected of everyone at UC San Diego. This means that you must be honest, fair, responsible, respectful, and trustworthy in all of your actions. Lying, cheating, or any other forms of dishonesty will not be tolerated because they undermine learning and the University's ability to certify students' knowledge and abilities. Thus, any attempt to get, or help another get, a grade by cheating, lying or dishonesty will be reported to the Academic Integrity Office and will result in sanctions. Sanctions can include an F in this class and suspension or dismissal from the

University. So, think carefully before you act by asking yourself: a) is what I'm about to do or submit for credit an honest, fair, respectful, responsible & trustworthy representation of my knowledge and abilities at this time and, b) would my instructor approve of my action? You are ultimately the only person responsible for your behavior. So, if you are unsure, don't ask a friend—ask your instructor, instructional assistant, or the Academic Integrity Office. You can learn more about academic integrity at academicintegrity.ucsd.edu” (Source: Academic Integrity Office, 2018)

To clarify a few things, collaborations are welcome and encouraged. Doing problem sets with your peers is fun and saves time. You can ask questions to your friends and learn with them or answer your friends' questions and reinforce your acquired knowledge. You can save time searching things on the internet on websites such as stackoverflow.com and others. However, these are not substitutes for hard work. This means that your answers should be yours only.

Please read UC San Diego's [Policy on Integrity of Scholarship](#) and take the [integrity pledge!](#)

AI policy

These tools are fantastic, and you should use them. But like a calculator, you must learn to compute before benefiting from it. I encourage you to try hard by yourself before using AI.

For coding, you can check [GitHub copilot](#). I am old-school, so I use [StackOverflow](#) to find help. [ChatGPT](#) is great, especially to ask to explain code.

About your instructor



Catarina is a 4th-year PhD candidate in the Political Science Department. Her research centers around electoral participation, with a regional focus on Latin America. She has a Bachelor's Degree in Social Sciences from the University of São Paulo, in Brazil, where she is from. Catarina enjoys exercising, painting, and eating good food (ask me for recommendations in SD!). She currently splits her time between San Diego and Portland, OR. [Learn more about Catarina here.](#)

Communication Expectations

1. Most of the communication regarding the class should be done with your TA. They will let me know if you have questions they cannot solve.

2. TAs have been instructed not to answer questions that are answered by the syllabus. **If you have a question about course logistics, the first thing you should do is check the syllabus for the answer.**
3. **I will only respond to emails if forwarded by the TA.** If you have any questions, please come by during office hours.
4. If you have any questions or want to chat about CSS, PoliSci, my research, sports, etc., please sign up for my office hours using Calendly.
5. My Calendly webpage: <https://calendly.com/acroman/office-hours>

Piazza Forum

Programming involves solving little pieces that build up to a larger body of knowledge. Since we may get stuck in little pieces of it, it is good to have a place to post our questions. This is where Piazza enters. We will be glad to answer any questions about the class there.

Help and support

I want to say that we are here for you! The TAs and I are committed to your learning experience, and if you have any questions or concerns, please let us know!

Syllabus Changing Policy

The syllabus is a plan, not a contract. It is subject to change throughout the semester. However, I will inform you about any changes or adjustments in any part of the syllabus.

Acknowledgment

This class is based on Professor Umberto Mignozzetti's CSS 1 class. He has been my mentor for many years, and I am excited to have this class passed on to me as an instructor! His course design is based on [Sean Trott's](#) CSS 1 class.

Resources for Support and Learning

You have many resources at your disposal at UCSD. Please make sure that you check them up.

Learning and Academic Support	
Ask a Librarian: Library Support <i>Chat or make an appointment with a librarian to focus on your research needs</i>	Writing Hub Services in the Teaching + Learning Commons <i>One-on-one online writing tutoring and workshops on key writing topics</i>
Course Reserves, Connecting from Off-Campus and Research Support <i>Find supplemental course materials</i>	Supplemental Instruction <i>Peer-assisted study sessions through the Academic Achievement Hub to improve success in historically challenging courses</i>

<p><u>First Gen Student Success Coaching Program</u> <i>Peer mentor program that provides students with information, resources, and support in meeting their goals</i></p> <p><u>Office of Academic Support & Instructional Services (OASIS)</u> <i>Intellectual and personal development support</i></p>	<p><u>Tutoring – Content</u> <i>Drop-in and online tutoring through the Academic Achievement Hub</i></p> <p><u>Tutoring – Learning Strategies</u> <i>Address learning challenges with a metacognitive approach</i></p>
<p>Support for Well-being and Inclusion</p>	
<p><u>Basic Needs at UCSD</u> <i>Any student who has difficulty accessing sufficient food to eat every day, or who lacks a safe and stable place to live is encouraged to contact: foodpantry@ucsd.edu basicneeds@ucsd.edu (858) 246-2632</i></p> <p><u>Counseling and Psychological Services</u> <i>Confidential counseling and consultations for psychiatric service and mental health programming</i></p> <p><u>Triton Concern Line</u> <i>Report students of concern: (858) 246-1111</i></p> <p><u>Office for Students with Disabilities (OSD)</u> <i>Supports students with disabilities and accessibility across campus</i></p>	<p><u>Community and Resource Centers</u> <u>Office of Equity, Diversity, and Inclusion</u> <i>As part of the <u>Office of Equity, Diversity, and Inclusion</u> the campus community centers provide programs and resources for students and contribute toward the evolution of a socially just campus (858).822-.3542 diversity@ucsd.edu</i></p> <p><u>Get Involved</u> <i>Student organizations, clubs, service opportunities, and many other ways to connect with others on campus</i></p> <p><u>Undocumented Student Services</u> <i>Programs and services are designed to help students overcome obstacles that arise from their immigration status and support them through personal and academic excellence</i></p>

Course Schedule (subject to adjustments based on class pace)

Week	Content
01	Introduction, tooling, and basic variables
02	Variables and syntax
03	Conditionals and Loops
04	Working with Strings

05	Lists and Dictionaries
06	Functions and Classes
07	Reading files
08	Numpy
09	Pandas
10	Data analysis in Python