# CSS 1 – Introductory Programming for Computational Social Sciences

**Instructor**: Dr. Umberto Mignozzetti
Office Hours: Tuesdays and Fridays, 6:00 PM to 8:00 PM (Zoom)
Office Hours Sign-Up: https://calendly.com/umbertomig/office-hours

**TA**: Lilly Amirjavadi
Email: lamirjav@ucsd.edu
Office Hours: Mondays, from 4:00 to 5:30 PM
Office Hours Sign-Up: https://calendly.com/lillyamirjavadi/css-1-office-hours

## 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 computation social sciences tasks and data analytics.

## Location, hours, and credits

Credits: 4

Class:
Location: Pepper Canyon Hall – Room 122
Hours: MWF 8:00-8:50 AM

TA Lab Sessions:
Session A01:
Location: Cognitive Sciences Building – Room 005
Hours: M 10:00-10:50 AM
Session A02:

Location: Cognitive Sciences Building – Room 005
Hours: F 3:00-3:50 PM

## 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 16 | 5 % | CANVAS quizzes |
| Coding Labs x 8 | 40 % | Due on Tuesdays:<br><br>Lab 01 – Jan 23<br>Lab 02 – Jan 30<br>Lab 03 – Feb 06<br>Lab 04 – Feb 13<br>Lab 05 – Feb 20<br>Lab 06 – Feb 27<br>Lab 07 – Mar 05<br>Lab 08 – Mar 12 |
| Problem Sets x 4 | 40 % | PS 01 – Jan 26<br>PS 02 – Feb 09<br>PS 03 – Feb 23<br>PS 04 – Mar 08<br><br>All PSs will be live two weeks before their due date. |

| | | |
|---|---|---|
| Final Project | 15 % | Due Mar 20 |
| 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 week, I will put on the board a pin for the participation quiz of the day. This will start counting since the first day.

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% x Finished + 30% x 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. It comprises 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 DataHub 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. Ask Lilly how to do this.

### *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.

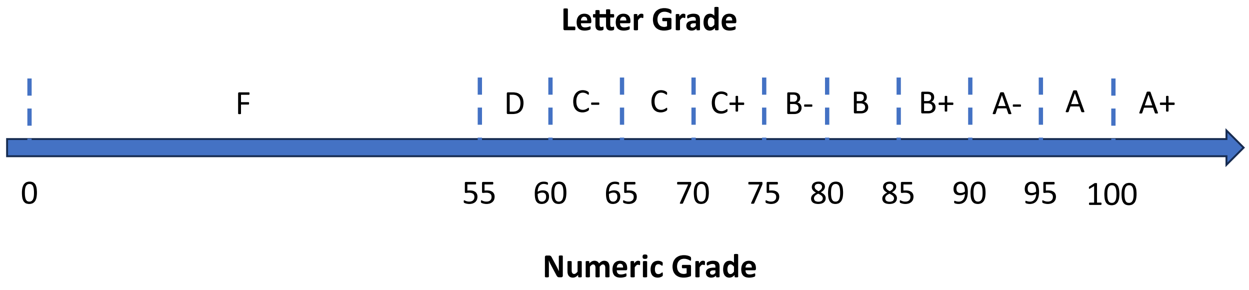
### *Grading Criteria*

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.

### *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.

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

## Generative AI Helper

This quarter, we will use a Generative AI tool to help with the course materials. You will notice that we will provide help in Canvas to start with:

1. *Prerequisites*: Students are encouraged to self-assess if they understand the prerequisite concepts needed for the lecture to come. If you have the required knowledge, these multiple-choice questions should be easy to answer. If you get the multiple-choice question wrong, you will be given a detailed explanation of the correct answer and a pointer to where to study.

2. *Preview*: We encourage you to attempt the preview before attending the lecture. The preview tests you on the concepts to be taught in the lecture so that you are primed to understand the material when it is presented in the lecture.

3. *Review*: The review tests you on the material just taught in the lecture. If you do not understand a concept, you can go back and try and understand it or ask for help.

Later in the course, we may introduce a generative AI tutor to engage with you and answer your questions directly. A subject matter expert will check all answers the AI tutor gave later to ensure correctness.

## 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.

## Instructors

*Dr. Umberto Mignozzetti*

You can find information about my work and research interests at http://www.umbertomig.com.

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. **I will only respond to emails if forwarded by the TA**. If you have any questions, please come by during office hours.
3. If you have any questions or want to chat about CSS, PoliSci, my research, sports, etc., please sign into my office hours using Calendly.
4. My Calendly webpage: https://calendly.com/umbertomig/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!

*Lilly Amirjavadi*

Email: lamirjav@ucsd.edu  
Calendly link: https://calendly.com/lillyamirjavadi/css-1-office-hours  
Office hours: Mondays, from 4:00 to 5:30 PM

## 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 Sean Trott's CSS 1 class. His class design was so great that I am honored in following his footsteps here.

## 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<br>*Chat or make an appointment with a librarian to focus on your research needs*<br>Course Reserves, Connecting from Off-Campus and Research Support<br>*Find supplemental course materials*<br>First Gen Student Success Coaching Program<br>*Peer mentor program that provides students with information, resources, and support in meeting their goals*<br>Office of Academic Support & Instructional Services (OASIS)<br>*Intellectual and personal development support* | Writing Hub Services in the Teaching + Learning Commons<br>*One-on-one online writing tutoring and workshops on key writing topics*<br>Supplemental Instruction<br>*Peer-assisted study sessions through the Academic Achievement Hub to improve success in historically challenging courses*<br>Tutoring – Content<br>*Drop-in and online tutoring through the Academic Achievement Hub*<br>Tutoring – Learning Strategies<br>*Address learning challenges with a metacognitive approach* |
| Support for Well-being and Inclusion | |
| Basic Needs at UCSD<br>*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<br>Counseling and Psychological Services<br>*Confidential counseling and consultations for psychiatric service and mental health programming* | Community and Resource Centers<br>Office of Equity, Diversity, and Inclusion<br>*As part of the Office of Equity, Diversity, and Inclusion the campus community centers provide programs and resources for students and contribute toward the evolution of a socially just campus*<br>(858).822-.3542 \| diversity@ucsd.edu<br>Get Involved |

| | |
|---|---|
| Triton Concern Line<br>*Report students of concern:* (858) 246-1111<br><br>Office for Students with Disabilities (OSD)<br>*Supports students with disabilities and accessibility across campus* | *Student organizations, clubs, service opportunities, and many other ways to connect with others on campus*<br><br>Undocumented Student Services<br>*Programs and services are designed to help students overcome obstacles that arise from their immigration status and support them through personal and academic excellence* |

## Course Schedule (subject to adjustments based on class level and 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 |

I like to adjust to my students so that we may go faster or slower. However, the content to be covered will be consistent with our CSS 1 curriculum planning (stated above).