

Syllabus

Course overview and goals

This course will introduce new programmers to Python and the Jupyter Notebook environment in the context of Computational Social Science problems. No programming experience is necessary as we'll start with the basics.

Main Course Webpage for Materials and Resources

[Main Page](#)

Goals

At the end of the course you should have a working understanding of the Python language and the Jupyter environment. This means that you will be able to:

- Create notebooks with markdown and Python code
- Write small computer programs in Python
- Read basic Python code and understand what it does
- Debug common errors in Python programs

Strategy

This course is structured to help you get there. The basic premise we start from is that programming is a skill, and to acquire a skill you need lots and lots of practice (see [expectations](#)). In lecture, we will introduce new basic elements, and will try them together as a class. In labs, we will work in groups to combine these elements into small programs with instructor help. In problem sets, you will apply these skills on your own to solve specific problems. As much as possible, we will use interesting Computational Social Science problems as motivating problems, but we are constrained first and foremost by finding problems appropriate for your developing Python skills.

Course Information

Winter, 2022

Instructors

Role	Name	email	Office hours
Instructor	John Serences	jserences@ucsd.edu	Weds 10:00-11:00AM
TA	Sourabh Raja-Murali	srajamurali@ucsd.edu	
TA	Panayu Keelawat	pkeelawa@ucsd.edu	Monday 5:00-6:00pm
TA	Pulkit Agrawal	p3agrawal@ucsd.edu	
TA	Purva Kothari	pukothar@ucsd.edu	Tuesday 5:00-6:00PM

All the Zoom links for class, labs (office hour links above)

- Main `{{ url_website }}`: contains all the materials and links
- Zoom (Note: lectures & labs use different zoom rooms):
 - `{{ url_lecture }}`: M/W 9am - 9:50am PST - conducted live, recordings posted on canvas
 - `{{ url_lm10 }}`: Monday 10am - 12pm PST - conducted live
 - `{{ url_lm12 }}`: Monday 12pm - 2pm PST - conducted live
 - `{{ url_lt9 }}`: Tuesday 9am - 11am PST - conducted live
 - `{{ url_lf9 }}`: Friday 9am - 11am PST - conducted live
- `{{ url_canvas }}`: used to post grades and recordings of lectures/labs
- `{{ url_datahub }}`: used to submit completed labs and problem sets

Recordings of class lectures (M/W) will be made available on `{{ url_canvas }}`

Materials

- All materials will be provided via this website and the links above.
- No textbook is required atop the lectures notes here, but we provide recommendations for some paid and free [extracurricular resources](#)
- No local software is required (as we will use remotely hosted jupyter notebooks). If you want to install a local copy, we recommend the bundled [anaconda distribution](#) of Python 3

Class/topic schedule, tentative and subject to change (as of 2021-12-25):

Date	Topics	Assignment due
M-01-03-2022	Basic python	
W-01-05-2022	Basic python	
M-01-10-2022	Control flow	
W-01-12-2022	Control flow	
M-01-17-2022	MLK - no class	
T-01-18-2022		pset 1 due (one day extension due to holiday)
W-01-19-2022	Stings	
M-01-24-2022	Lists	
W-01-26-2022	Lists	
M-01-31-2022	Dictionaries	
M-01-31-2022		pset 2 due
W-02-02-2022	Dictionaries	
M-02-07-		

2022	Functions	
W-02-09-2022	Functions	
M-02-14-2022	Classes	
M-02-14-2022		pset 3 due
W-02-16-2022	Classes	
M-02-21-2022	Presidents' Day - no class	
W-02-23-2022	Pandas Series	
M-02-28-2022	Pandas Dataframes	
M-02-14-2022		pset 4 due
W-03-02-2022	Plotting (Matplotlib)	
M-03-07-2022	Advanced	
W-03-09-2022	Advanced	
F-03-11-2022		final pset due

Grading

Basis

You are evaluated based on:

- 45% 4 bi-weekly problem sets
- 40% 9 labs (1 per week, starting Week 2) - 15% final (due Friday March 11, 2022, end of week 10)

Labs: Labs are short exercises designed to be completed during the scheduled lab time, with interactive help from the TA and other students. Labs are completed by turning them in on datahub. Labs are due by the end of the day following lab (e.g., a Tuesday lab is due by end of day Wednesday). This window is wide so that people who cannot attend lab, or otherwise do not complete the work during lab, can submit on their own schedule. That said, *it is strongly advised that you attend lab to complete the activities and get interactive help!*

Problem Sets: Are longer, bi-weekly assignments. They are due by the end of the day (11:59PM) every other Monday (see above table for exact due dates). You are to complete each problem set **on your own** (i.e., these are not group projects, and you are not allowed to seek help from classmates or anyone else). You are advised to *start early* so that you can do the easier problems to get credit and so that you can identify and set aside extra time for the problems that are hardest for you.

Final: The final is a more involved, more integrated problem set and will be cumulative (as will all the other problem sets given the cumulative nature of programming). The final problem set will be due at the end of week 10 (11:59PM on Friday, March 11th, 2022).

Pro-Social Behavior: While not a graded component of the class, all students are encouraged to help other students and the instructors. Most importantly, **everyone is expected to contribute to a supportive and positive class environment**. Programming can be very intimidating, and my expectation is that all questions and communications will be treated with respect, patience, and honest attempts to answer. This includes things like: showing up and participating during lectures and labs, participating in slack discussion (asking well thought-out questions, answering others' questions), demonstrating an interest in learning, not just maximizing your grade, etc.

Letter grades

Letter grades will be based on the percentage of total points earned across the items above. Standard grading scale with 90-100 an A, 80-89.9999 a B and so forth. Each letter grade will be further broken into +'s and -'s based on which third of the interval you are in (e.g., a 92% would be an A-, and 85 would be a B, and a 82% would be a B-). More formally, if we encode the percentage of total points that you earn in the variable `percent`, you can obtain your grade using the following python code:

```
if percent >= 90:
    letter = 'A'
    remainder = percent - 90

if 90 > percent >= 80:
    letter = 'B'
    remainder = percent - 80

if 80 > percent >= 70:
    letter = 'C'
    remainder = percent - 70

if 70 > percent >= 60:
    letter = 'D'
    remainder = percent - 60

if 60 > percent:
    letter = 'F'
    remainder = 5

if remainder >= 7:
    modifier = '+'
elif remainder < 3:
    modifier = '-'
else:
    modifier = ''

grade = letter + modifier
```

Assignment scores

Assignment scores will be posted on Canvas.

Manual Regrades

Problem sets and labs are scored using [nbgrader](#) on [UCSD datahub](#). Some parts are graded automatically by computer, and some parts are graded manually by a human.

We will work hard to grade everyone fairly and return assignments quickly. And, we know you also work hard and want you to receive the grade you've earned. Occasionally, grading mistakes do happen, and it's important to us to correct them. If you think there is a mistake in your grade on an assignment, post privately using a direct message on Slack to me and your lab TA within 72 hours. This post should include evidence of why you think your answer was correct and

should point to the specific part of the assignment in question.

Note that points will not be rewarded if you fail to follow instructions. For example, if the instructions say to name the variable `orange` and you name it `ornage` (misspelled), you will not be rewarded credit upon regrade. This is because (1) following instructions and being detail-oriented in general, (2) referring to things by their correct names, and getting other minor technicalities right is *essential* to programming.

Questions, feedback, and communication

The instructors can be reached in the following ways:

- Drop in during scheduled **office hours** (see [syllabus](#) for links and schedule).
- Public message on `{{ url_slack }}`.
- Private direct message to specific lab TA or to the course instructor on `{{ url_slack }}`

Outside of office hours, all communication should happen over `{{ url_slack }}`. Email is reserved for the unanticipated circumstances when Slack is down, or you cannot access it. In that case, email your lab TA [M 10-12](#), [M 12-2](#), [T 9-11](#), [F 9-11](#) about inability to access slack.

Specific types of questions / comments

- **if you have questions about course logistics:** First, check the syllabus and the detailed how-to pages on the `{{ urlwebsite }}`. *If you can't find the answer there, first ask a classmate if you are able. If still unsure or you are unable to reach a classmate, post on `{{ urlslack }}`.*
- **if you have questions about course content:** these are awesome! We want everyone to see them, be able to answer them, and have their questions answered too, so post these to `{{ url_slack }}`.
- **if your code produces an error that you cannot fix:** first follow the [debugging instructions](#) to find a minimal reproducible example and fill out the debugging question checklist, then post on Slack if you are still stuck after following that guidance.
- **a technical assignment question:** Come to office hours (or post to Slack). Answering technical questions is often best accomplished 'in person' where we can discuss the question and talk through ideas. However, if that is not possible, post your question to `{{ url_slack }}`. Be as specific as you can in the question you ask. And, for those answering, help your classmates as much as you can without just giving the answer. Help guide them,

point them in a direction, and you can even provide pseudo code (e.g., guidance about the steps that one might take to solve the problem), **but do not provide code that directly answers assignment questions.**

- **been stuck on something for a while (>30min) and aren't even really sure where to start** - Programming can be frustrating and it may not always be obvious what is going wrong or why something isn't working. That's OK - we've all been there! IF you are stuck, you can and should reach out for help, even if you aren't exactly sure what your specific question is. To determine when to reach out, consider the 2-hour rule. This rule states that if you are stuck, work on that problem for an hour. Then, take a 30 minute break and do something else. When you come back after your break, try for another 30 minutes or so to solve your problem while working through our [debugging](#) checklist. If you are still completely stuck, stop and contact us (office hours, post on Slack). If you don't have a specific question, include the information you have (what you're stuck on, the [debugging checklist](#)).
- **questions about a grade** - Post on Slack in a direct message to your instructor and lab TA.
- **something super cool to share related to class or want to talk about a topic in further depth** - come to office hours, post on Slack, or send in a DM to the instructors!

Slack Rules

Slack is an incredible resource for technical classes. It gives you a place to post questions and an opportunity to answer others' questions. We do our very best as an instructional staff to make sure each and every question is answered in a timely manner. We also want to make sure this platform is being used to learn and not thwarting anyone's education. To make all of this possible, there are a few rules for this course's slack:

1. Before posting your question, look at questions that have already been posted to avoid duplicates.
2. If posting about an assignment, note title should have assignment number, question number, and 1-2 words about the question. (i.e. PS01 Q1 Variable Naming)
3. Never post an answer to or code for an assignment on a public post. Pseudocode is encouraged for public posts (e.g., plain english hints on the steps that one might take to solve a problem, but **do not** insert the actual code). If you must include code for an assignment in order to make your question clear, do so **only** via direct message to your lab TA or to the instructor.
4. Your post must include not only your question/where you're stuck, but also what you've already done to try to solve it so far and what resources (class notes, online URLs, etc.)

you used to try to answer the question up to this point. See how to ask [debugging questions](#).

Remote learning during weeks 1-2

The current campus plan is to hold all courses over zoom during weeks 1 and 2 while people slowly move back onto campus. While most of us have now gotten a bit of practice participating in class over zoom, there is still a lot going on in the world. While students have always been under the typical pressure and stress of navigating college courses, the struggles students may encounter this quarter (for a whole bunch of different reasons) may go beyond what is typical. We acknowledge this and we are here to help you succeed.

Please take care of yourselves and one another, and we'll work as hard to ensure a supportive learning environment for all students this quarter.

Remote technology

If you do not have consistent access to the technology needed to fully access remote instruction options, please use the form below to request a loaner laptop for the period during which you will be learning remotely due to the COVID-19 pandemic: <https://eforms.ucsd.edu/view.php?id=490887>. (For any issues that you may have, please email vcsa@ucsd.edu and they will work to assist you.)

Lectures and Labs

Due to uncertainty around campus closures, attendance will not be required for any part of the course this quarter and the first two weeks of class will be held remotely over zoom (barring any changes to current campus plans). That said, lectures and coding labs will take place during their scheduled times, and attending class and labs synchronously -- and in person when campus fully re-opens -- is highly encouraged to get the full benefits of the course.

Lectures will take place at their scheduled time (M/W 9-9:50am). The instructor will lecture and do live coding in class and students will be expected to write out the code along with the instructor, to complete small coding challenges during lecture on their own, and to interact with the instructor and classmates during lecture. Following along with the lectures in real time and writing out the code is highly recommended for success - as noted elsewhere, there is no substitute for practice when learning the *skill* of coding.

However, every lecture will be recorded and shared so that students who are not able to or choose not to participate during the scheduled class time are still able to receive and digest all

class materials. That said, this class is not designed to be taught in an entirely remote format (i.e, it is **NOT** an intentionally designed "R" course). Thus, during the first few weeks of class when campus is partially closed, we will hold lectures/labs on zoom and those zoom sessions will be recorded and posted. However, after campus re-opens, the lectures/labs will **not be simulcast in realtime on zoom**. Instead, they will be recorded using the podcast capabilities in the classrooms and no real-time remote interactions will be possible. As a result, those who participate entirely asynchronously will not be able to ask questions in real time during class/labs. *This makes it especially important to attend the lab sections in person* after campus re-opens, as the TAs will be moving about in the computer lab to interactively help students and we will not be able to capture those interactions for later viewing.

All lecture and lab recordings will be available on {{ url_canvas }} in the Media Gallery.

UCSD policies & resources

Academic Integrity

[From UCSD Academic Integrity office](#)

Integrity of scholarship is essential for an academic community. The University expects that both faculty and students will honor this principle and in so doing protect the validity of University intellectual work. For students, this means that all academic work will be done by the individual to whom it is assigned, without unauthorized aid of any kind.

[Please read the full UCSD policy](#)

Labs You are encouraged to work together and help one another on the *lab assignments*. That said, you are *personally responsible* for the work you submit, and it is your responsibility to ensure you understand everything you've submitted.

Problem sets and final. *You must work independently on the problem sets and the final.* You may ask and answer [debugging questions](#) on slack or in lab sections, but doing work for another student or providing assistance outside of public questions on slack on the problem sets or final project will be treated as a violation of academic integrity and you will be referred for disciplinary action. Similarly, emailing with or otherwise communicating with other students or anyone else for assistance on the problem sets or the final will be treated as a violation of academic integrity and will be referred for disciplinary action. Cheating and plagiarism have been and will be strongly penalized. Please review campus academic integrity policies [here](#).

You are responsible for ensuring that the correct file has been submitted and that the

submission is uncorrupted. If, for whatever reason, Canvas or DataHub is down or something else prohibits you from being able to turn in an assignment on time, immediately contact the instructor by emailing the assignment, otherwise the assignment will be graded as late.

Class Conduct

In all interactions in this class, you are expected to be respectful of your classmates, the TAs, and the instructor. This includes following the [UC San Diego principles of community](#).

The teaching team places top priority on making class a welcoming, inclusive, and harassment-free experience for everyone, regardless of gender, gender identity and expression, age, sexual orientation, disability, physical appearance, body size, race, ethnicity, religion (or lack thereof), political beliefs/leanings, or technology choices.

At all times, you should be considerate and respectful. Always refrain from demeaning, discriminatory, or harassing behavior and speech. Last of all, take care of each other.

If you have a concern, please speak with the Professor or your TAs. If you are uncomfortable doing so, the [OPHD](#) (Office for the Prevention of Sexual Harassment and Discrimination) and [CARE](#) (confidential advocacy and education office for sexual violence and gender-based violence) are wonderful resources on campus.

Disability Access

Students requesting accommodations due to a disability must provide a current Authorization for Accommodation (AFA) letter. These letters are issued by the Office for Students with Disabilities (OSD), which is located in University Center 202 behind Center Hall. Please make arrangements to contact Professor privately to arrange accommodations.

Contact information for the OSD:

858.534.4382 (phone)

osd@ucsd.edu (email)

<http://disabilities.ucsd.edu>

Important Resources for Students

- [UCSD's principles of community](#)
- [Counseling and Psychology Services \(CAPS\)](#). "CAPS provides FREE, confidential, psychological counseling and crisis services for registered UCSD students. CAPS also provides a variety of groups, workshops, and drop-in forums."

- [CARE at the Sexual Assault Resource Center](#) is the UC San Diego confidential advocacy and education office for sexual harassment, sexual violence and gender-based violence (dating violence, domestic violence, stalking).
- [Office for the Prevention of Harassment & Discrimination \(OPHD\)](#). OPHD "works to resolve complaints of discrimination and harassment through formal investigation or alternative resolution."