

CS229 Fall 2023-24 Course Logistics + FAQs

1. Prerequisites

Students are expected to have the following background:

2. Knowledge of basic computer science principles and skills, at a level sufficient to write a reasonably non-trivial computer program in Python/NumPy. (CS106A or CS106B, CS106X.)
 - Familiarity with probability theory. (CS 109, MATH151, or STATS 116)
 - Familiarity with multivariable calculus and linear algebra (relevant classes include, but not limited to MATH 51, MATH 104, MATH 113, CS 205, CME 100.) Stanford Math 51 course text can be found [here](#).

3. Honor Code

We strongly encourage students to form study groups. Students may discuss and work on homework problems in groups. However, each student must write down the solution independently, and without referring to written notes from the joint session. Each student must understand the solution well enough in order to reconstruct it by him/herself.

It is an honor code violation to copy, refer to, or look at written or code solutions from a previous year, including but not limited to: official solutions from a previous year, solutions posted online, solutions you or someone else may have written up in a previous year, and solutions for related problems. In particular, you may not compare your solutions and code to other students' or to resources you can find online. If you discuss a problem with others or use an online resource for help, you must cite this in your submission.

Furthermore, it is an honor code *violation to post your assignment solutions online*, such as on a public git repo. We run plagiarism-detection software on your code against past solutions as well as student submissions from previous years. Additionally, you may not screen share your Overleaf doc or code to other students (including during office hours). Please take the time to familiarize yourself with [the Stanford Honor Code](#) and [the Stanford Honor Code as it pertains to CS courses](#). If you're ever unsure about whether something is an honor code violation, please create an Ed post (or reach out to our Honor Code Lead, TBD).

4. Friday TA Lectures

To review material from the prerequisites or to supplement the lecture material, additional lectures led by TAs will be held every Friday from **TBD**. Attendance at these lectures is optional, but encouraged.

5. Optional Discussion Sections

There will also be optional weekly discussion sections led by TAs. These sessions are meant to be interactive and in a small, traditional classroom setting. They will largely involve working through problems that are similar to the homeworks. We will have one discussion section each week, for six weeks in the quarter, starting from Week 2. The timings, schedule and location will be updated soon.

6. Syllabus and Course Materials

There is no required text for this course. The syllabus is available [here](#). Notes will be posted periodically in the syllabus.

7. Course Calendar & Office Hours

A course calendar with times for the lectures, office hours, Friday TA lectures, and discussions can be found on Canvas. All assignment deadlines and the midterm date can be found in the syllabus [here](#).

Office hours will primarily be held in person with locations/times on the Canvas calendar. There will be a smaller # of remote OO targeted at SCPD/remote students (but open to all) that will also be reflected on Canvas with Zoom links.

8. Ed and Gradescope

Ed is the primary method of communication for this class with announcements and Q&A. Gradescope will be used for assignment submission.

Ed and Gradescope access will be granted after enrollment to the class as we periodically synchronize with the official course roster. ***If you enrolled in the class but do not have access to Ed or Gradescope, it should come within two days.*** If it has been more than that, send David Lim: dslim@stanford.edu an email.

9. Grading

There will be four assignments, one midterm, and a final project. The assignments will contain written questions and questions that require some

Python programming. The grading breakdown is as follows: assignments are collectively worth 40%, the midterm is worth 20%, and the final project is worth 40%. The assignments are weighted by their respective point values - for example, if {**p1, p2, p3, p4**} denotes the point values of each assignment, then HW1 is worth $p1 / (p1+p2+p3+p4) * 40\%$ of the total grade. This quarter's grading basis is letter grade or CR/NC. Please make sure on Axxess that you are enrolled with your desired grading basis.

*We highly encourage students to answer each others' questions on Ed. To incentivize this, we will be giving **2% extra credit** of the total course grade to the **ten** students with the most instructor-endorsed responses on Ed by the end of the quarter.*

10. Submitting Assignments

To limit access to the assignments to only enrolled students and avoid the solutions to show up online publicly, the assignments will only be posted on Ed (not the course website nor Canvas). Assignments will be submitted through [Gradescope](#). You will receive an invite to Gradescope for CS229 Machine Learning Fall 2023. If you have not received an invite email after the first few days of class, first log in to Gradescope with your @stanford.edu email and see whether you find the course listed; if not please post a private message on Ed for us to add you. All homework assignments must be submitted individually. The final project for the course can be submitted as a group of up to three members by adding group-members to a single submission on Gradescope.

11. Late Assignments

Each student will have a total of three free late (calendar) days to use for homeworks. Once these late days are exhausted, any assignments turned in late will be penalized 20% per late day. However, no assignment will be accepted more than three days after its due date. Each 24 hours or part thereof that a homework is late uses up one full late day. Please note that late days are applied individually. Please keep in mind that **late days cannot be applied to the final project report and poster deadlines** due to constraints in the grading deadline. OAE accommodations do not apply to group assignments.

12. Regrade Requests

Students will have **one week** to submit regrade requests on Gradescope for assignments starting from when the grades are released.

13. AI Tools Policy

Students are required to independently submit their solutions for CS229 homework assignments. Collaboration with generative AI tools such as Co-Pilot and ChatGPT is allowed, treating them as collaborators in the problem-solving process. However, the direct solicitation of answers or copying solutions, whether from peers or external sources, is strictly prohibited.

It is crucial to emphasize that employing AI tools to substantially complete assignments or exams will be considered a violation of the Honor Code. For additional details, please refer to the Generative AI Policy Guidance [here](#).

14. Lecture Video Policy

Lectures this quarter will be held **in-person** location **TBD 1:30 PM - 2:50 PM (PST) on Mondays and Wednesdays**. Some of Prof. Andrew Ng's lectures will be over zoom, while all of Profs. Moses Charikar and Carlos Guestrin's lectures will be in-person at Gates B1. All lectures this quarter will be recorded and posted on Canvas soon after the lecture is given. You can find the recordings on the course Canvas page. These recordings might be reused in other Stanford courses, viewed by other Stanford students, faculty, or staff, or used for other education and research purposes. Note that while the cameras are positioned with the intention of recording only the instructor, occasionally a part of your image or voice might be incidentally captured. If you have questions, please contact a member of the teaching team.

15. Midterm Policy

We will have an in-person 3-hour timed open-notes exam on **TBD**. For SCPD/NDO students: You will be taking the exam remotely and need to nominate an exam monitor. Please refer to the SCPD handbook for information about the process: <https://online.stanford.edu/non-degree-option-program#EXAMS-AND-HOMEWORK>. In order to better accommodate our SCPD students in different timezones, we allow a 24 hour time window starting from the date/time of the main on-campus exam time for SCPD students to choose the time stamped 3 hour exam. Please reach out to the SCPD exams office (scpd-exams@stanford.edu) if you have any questions on the exam process for SCPD students.

The teaching staff will provide more details on the exam once it is finalized.

16. Office Hours and QueueStatus

The office hour schedule will be posted on the course calendar on Canvas by the end of Week 1. We will hold many in-person office hours and some remote (via Zoom) office hours this quarter. We will be using QueueStatus to determine priority order. After putting your name in the queue, please watch for messages from the TAs on QueueStatus. More information on the logistics of Office Hours will be posted on Ed.

17. Incomplete Requests from Previous Quarters

If you have an Incomplete from a previous quarter and you wish to complete the course this quarter, please contact John Cho (johncho@stanford.edu) at the beginning of the quarter to notify us that you would like to complete CS229 this quarter.

18. Auditing

Please fill out this [form](#). All audit requests will be thoroughly reviewed, and if approved, you will be added as an observer to the course's Canvas page. It's important to note that as an auditor, access to the Ed forum for assignments will not be provided. Additionally, auditors do not have access to Gradescope for assignment submissions. This policy is uniform, and no exceptions will be made.

Our primary focus is to ensure that registered students have the full opportunity to participate and benefit from the course. In cases of limited space, we encourage you to engage with the course content through the Canvas platform.

Please email John Cho (johncho@stanford.edu) if you are waiting to receive access.

19. Students with Documented Disabilities

Students should also send their accommodation letter to the teaching staff by making a private post on Ed, as soon as possible.

We assume that all of us learn in different ways, and that the organization of the course must accommodate each student differently. We are committed to ensuring the full participation of all enrolled students in this class. If you need academic accommodation based on a disability, you should initiate the request with the [Office of Accessible Education \(OAE\)](#). The OAE will evaluate the request, recommend accommodations, and prepare a letter for faculty. Students should

contact the OAE as soon as possible and at any rate in advance of assignment deadlines since timely notice is needed to coordinate accommodations.

FAQ

1. Difference between 3 and 4 units?

- a. Students, both undergraduates and graduates, have the option to enroll in the class with either 3 or 4 units. The workload remains consistent between the two options. This flexibility is provided to accommodate individual preferences or unit constraints. While there is a general encouragement for students to register for 4 units, opting for 3 units is entirely acceptable without the need for prior approval. Feel free to choose the unit count that aligns best with your specific circumstances.

2. Is this the same class as the free machine learning class?

- a. No, that is a different class, which is not good for Stanford academic credit. You can learn more about it at www.ml-class.org.

3. When will solutions for homeworks be released?

- a. The release of solutions will coincide with the completion of homework grading and the publication of grades. Specifically, for HW0, solutions will be made available shortly after the submission deadline

4. Can I take courses that overlap with CS229?

- a. Typically, this is not permitted; however, an exception can be granted under the condition that the student taking overlapping courses acknowledges and commits to fulfilling all obligations despite potential schedule conflicts. We will not provide accommodations. If an instructor's signature is necessary, kindly contact John Cho at johncho@stanford.edu.

5. Why am I seeing an out-dated web page with information from previous quarters?

- a. We try our best to keep the website up-to-date starting from a few days before the quarter starts. You might want to force reload the page and override local cache.

On Mac, use Command + Shift + R .

On Windows/Linux, use Ctrl + Shift + R .

6. How should I ask for TAs to help me debug code?

- a. Please note that the teaching staff will not debug code longer than 2-3 lines via Ed. Learning to debug is a critical skill for software programmers, and remote requests for help with code usually end up with the teaching staff giving you the answer rather than you learning how to find the answer.
- b. Moreover, since programming at the level of CS106A/B is a prerequisite for this course and the course's focus is on machine learning techniques rather than coding, the TAs are **discouraged** from helping you look at and debug large blocks of your code during the office hours. The TAs are also generally discouraged to help debug compilation errors.
- c. The best way to use office hours and ask TAs for coding questions would be
 - i. You should come to office hours having done your own legwork and ruled out basic logical errors. Identify the place where the error is suspected to come from by doing ablation studies. (Please see below for some common debugging tips.)
 - ii. During the office hours, you should articulate what your goals are and what you have observed in your experiments, what you think might be the problem, and what advice you need to move forward.
 - iii. The TAs will mostly help you by **looking at and analyzing the outputs of your code** instead of looking at the original code. Typical advice that the TAs might give you would be to ask you to do more analytical or ablation studies about your code. For example, when you observe that your test error does not decrease as training for longer, the TAs might ask you to check if your training error decreases. If your training error does not decrease, then the TAs might ask to check if the gradient of your algorithm is implemented correctly.
- d. Here are some common debugging strategies that might be useful (courtesy of CS221)
 - i. Construct small test cases that you have worked through by hand and see if your code matches the manual solution.
 - ii. Spend some time understanding exactly what the test cases are doing and what outputs they are expecting from your code.

- iii. If possible, write your codes in small chunks and test that each part is doing exactly what you expect.
- iv. [PDB](#) is the default python debugger. It is very helpful and allows you to set breakpoints. You can set a breakpoint with the following lines: `import pdb; pdb.set_trace()` .
- v. Printing the state of your computation frequently can help you make sure that things are working as expected and can help you narrow down which portion of your code is causing the bug you are seeing, e.g. `print("var1 has current value: {}".format(var))` .
- e. Debugging tips for timeouts:
 - i. Set operations in general are pretty slow, so if you have any see if you can do them in some other way.
 - ii. Check if all loops / linear operations are necessary. For example, with searching through a list for a specific item, sometimes you can make that constant time by giving each item an ID (say 0, 1, 2, 3) and then using a dictionary as a cache (although sometimes you just have to live with the cost).
 - iii. If you have a specific helper function you are calling a lot, see if there is anything in there you can optimize!
- f. Other debugging tips
 - i. If you do not know what type a variable is, use `type(.)` .
 - ii. If you are running into issues where "None" pops up, a function may not be returning what you are expecting.
 - iii. For indexing into lists: `example_list[a: b]` is INCLUSIVE for a but EXCLUSIVE for b.
 - iv. If a function has optional arguments, make sure you are feeding in the proper arguments in the proper places (very easy to mess up).
 - v. Since python 3.6, you can use [f-strings for printing debug messages](#), rather than `format`.
 - vi. Because of broadcasting and other implicit operations, it's useful to assert shapes of np arrays (and tensors for deep learning) after each operation that can change the shape.