# CS229 Winter 2023-24 Course Logistics + FAQs

### Prerequisites

Students are expected to have the following background:

1. 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](https://web.stanford.edu/class/math51/stanford/math51book.pdf).

### 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](https://communitystandards.stanford.edu/policies-and-guidance/honor-code) and [the Stanford Honor Code as it pertains to CS courses](http://web.stanford.edu/class/archive/cs/cs106b/cs106b.1164/handouts/honor-code.pdf). 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).

### 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 **1:30pm - 2:50pm.** The location is [**HEWLETT102, William R. Hewlett Teaching Center**](https://25live.collegenet.com/pro/stanford#!/home/location/72/details)**.** Attendance at these lectures is optional, but encouraged.

### Optional Discussion Sections

In addition to the main course structure, optional weekly discussion sections will be facilitated by TAs. These sessions are designed to be interactive, fostering a small and traditional classroom setting. The primary focus of these discussions will involve collaborative problem-solving, particularly addressing problems akin to those found in the homework assignments.

The discussion sections are scheduled for six weeks during the quarter, commencing from Week 2. Details regarding timings, schedule, and location will be provided via ED.

### Syllabus and Course Materials

There is no required text for this course. The syllabus is available [here](https://docs.google.com/spreadsheets/d/1-H4TZo6xHLR0o9HR4-SSnR7F5KJJi9OT/edit#gid=348511794). Notes will be posted periodically in the syllabus.

### 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](https://docs.google.com/spreadsheets/d/1-H4TZo6xHLR0o9HR4-SSnR7F5KJJi9OT/edit#gid=348511794).

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.

### Ed and Gradescope

Communication within this class primarily takes place through Ed, serving as the primary platform for announcements, and Q&A. Gradescope will be the designated platform for submitting assignments.

Access to both Ed and Gradescope will be granted after enrollment in the class, with periodic synchronization occurring with the official course roster. **If you have enrolled in the class but still need access to Ed or Gradescope, please allow up to two days for synchronization.** If access is not granted within this timeframe, kindly email our Head TA, Jason Ding, at d1ng@stanford.edu for further assistance. Your patience and cooperation in this matter are appreciated.

### Grading

The evaluation for this course comprises four assignments, one midterm, and a final project. Assignments encompass both written and Python programming questions. The grading breakdown is as follows: assignments collectively contribute 40%, the midterm constitutes 20%, and the final project holds a weight of 40% in determining the overall grade. 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. The grading basis for this quarter is letter grade or Credit/No Credit (CR/NC). It is crucial to verify your enrolled grading basis on Axess and ensure alignment with your preference.

*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.*

### Submitting Assignments

To ensure restricted access to assignments and prevent public exposure of solutions, all assignment postings will exclusively be available on ED, excluding the course website and Canvas. For assignment submissions, Gradescope will be the designated platform.

You will receive an invitation to join Gradescope via email. If you are still waiting to receive the invite within the initial days of the class, please log in to Gradescope using your @stanford.edu email. If the course is not listed, kindly initiate a private message on Ed to request inclusion.

It is essential to note that each homework assignment must be submitted individually. However, submissions can be executed for the final project as a group comprising up to three members. Group members can be added to a single submission on Gradescope to streamline the submission process. Your cooperation with these guidelines ensures a secure and efficient learning environment for all enrolled students.

### Late Assignments

Each student is granted a total of three free late (calendar) days to use for homework assignments. Upon exhaustion of these late days, any late submissions will incur a penalty of 20% per late day. **It is crucial to be aware that assignments will not be accepted if submitted more than three days after the due date. There are no exceptions to this rule.** The calculation of late days is on a 24-hour basis, with each 24-hour period or part thereof constituting one full late day. It is important to note that late days are applied individually and cannot be accumulated or combined.

**Please be mindful that late days are not applicable to the final project report and poster deadlines due to constraints in the grading schedule. Furthermore, OAE assignment extensions do not extend to group assignments.**

**In the case of group assignments, the utilization of late days is contingent upon the availability of late days for every member within the group.** Please coordinate with your group members to confirm the availability of late days before applying them to any group assignment.

We appreciate your adherence to these guidelines to ensure a fair and efficient evaluation process.

### Regrade Requests

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

### AI Tools Policy

Each student is expected to submit their own solutions to the CS229 homeworks. You may use generative AI tools such as Co-Pilot and ChatGPT as you would use a human collaborator. This means that you may not directly ask generative AI tools for answers or copy solutions, and acknowledge generative AI tools as collaborators. The use of generative AI tools to substantially complete an assignment or exam (e.g. by directly copying) is prohibited and will result in honor code violations. We will be checking students' homework to enforce this policy. For more details: Generative AI Policy Guidance reference [here](https://communitystandards.stanford.edu/generative-ai-policy-guidance).

### Lecture Video Policy

Lectures this quarter will be held **in-person** in the **NVIDIA Auditorium from 1:30 PM - 2:50 PM (PST) on Mondays and Wednesdays**. 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.

### Midterm Policy

We will have an in-person 3-hour timed open-notes exam on **Friday, Feb 16, 2024, 6pm-9pm**. 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](mailto: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.

### 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.

### 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.

### Auditing

Please fill out this [form](https://forms.gle/oJyyXQh41oSKyqkC8) here. All audit requests will undergo a thorough review, and upon approval, auditors will be added as observers to the course's Canvas page. **It is essential to know that auditors do not get access to the Ed forum and assignments via GradeScope. There are no exceptions to this rule**. Please be advised that our primary focus is facilitating a conducive learning environment for officially registered students. In instances where space is constrained, auditors are encouraged to join course activities via Panopto to ensure an inclusive experience. We appreciate your understanding and cooperation in prioritizing registered students' access to course resources and support channels.Please email John Cho ([johncho@stanford.edu](mailto:johncho@stanford.edu)) if you are waiting to receive access.

### 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)](https://oae.stanford.edu/). 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

### Difference between 3 and 4 units?

* 1. This course can be taken with either 3 or 4 units, accommodating both undergraduate and graduate students. The workload remains consistent between the two options. This flexible unit structure has been implemented to give students academic planning adaptability. Students can choose either 3 or 4 units. While the general recommendation is to register for four units, opting for three units is acceptable for any reason, including unit constraints. No formal approval is required for selecting the preferred unit count.

### Is this the same class as the free machine learning class?

* 1. 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](http://www.ml-class.org/).

### When will solutions for homeworks be released?

* 1. The solutions for homework assignments will be made available subsequent to the completion of the grading process and will coincide with the publication of grades. Specifically, for HW0, solutions will be released shortly after the submission deadline.

### Can I take courses that overlap with CS229?

* 1. Under normal circumstances, concurrent enrollment in multiple courses is not allowed. However, exceptions may be considered on a case-by-case basis, subject to the condition that the student undertaking overlapping courses explicitly recognizes and commits to fulfilling all obligations, even in potential schedule conflicts. The student is responsible for adhering to all dates and deliverables outlined in both courses. No accommodations will be granted for overlapping schedules. Should you require an instructor's approval, please get in touch with John Cho at johncho@stanford.edu for further assistance.

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

* 1. 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 .

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

* 1. 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.
  2. 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.
  3. The best way to use office hours and ask TAs for coding questions would be
     1. 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.)
     2. 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.
     3. 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.
  4. Here are some common debugging strategies that might be useful (courtesy of CS221)
     1. Construct small test cases that you have worked through by hand and see if your code matches the manual solution.
     2. Spend some time understanding exactly what the test cases are doing and what outputs they are expecting from your code.
     3. If possible, write your codes in small chunks and test that each part is doing exactly what you expect.
     4. [PDB](https://docs.python.org/2/library/pdb.html) 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() .
     5. 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)) .
  5. Debugging tips for timeouts:
     1. Set operations in general are pretty slow, so if you have any see if you can do them in some other way.
     2. 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).
     3. If you have a specific helper function you are calling a lot, see if there is anything in there you can optimize!
  6. Other debugging tips
     1. If you do not know what type a variable is, use type(.) .
     2. If you are running into issues where “None” pops up, a function may not be returning what you are expecting.
     3. For indexing into lists: example\_list[a: b] is INCLUSIVE for a but EXCLUSIVE for b.
     4. If a function has optional arguments, make sure you are feeding in the proper arguments in the proper places (very easy to mess up).
     5. Since python 3.6, you can use [f-strings for printing debug messages](https://realpython.com/python-f-strings/), rather than format.
     6. 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.