

[CS136](#)[Home](#)[Syllabus](#)[Course Materials](#)[Resources](#)[Staff & Office Hours](#)

# Syllabus



This is a class about the digital economy, specifically the interplay between economic thinking and computational thinking as it relates to electronic commerce, incentives engineering, and networked systems. Topics covered vary each year, but include a subset of:

- game theory (including algorithmic game theory),
- auctions (including internet advertising and combinatorial auctions),
- incentive compatible mechanism design (including theoretical and algorithmic approaches),
- human computation, crowd sourcing, and peer prediction,
- matching algorithms,
- trust and reputation,
- cryptoeconomics,
- networks (network formation, cascades, games on networks),
- privacy (including differential privacy, privacy-protected advertising), and
- ethical considerations.

Emphasis is given to fundamental topics, and the class involves the discussion of theoretical and empirical content, as well as algorithm design. We hope to convince you that economic considerations matter in many computational settings, and that computation matters in many economic ones.

## Prerequisites

We will assume familiarity with probability theory (as taught by, e.g., Stat 110), single-variable calculus (as taught by, e.g. Math 1b), linear algebra (as taught by, e.g., Math 21b/22a/25a/AM 21b), and the basics of theoretical computer science (e.g., complexity theory, asymptotic runtime analysis, as taught by CS 120, 121, or 124). You should also be prepared to write multi-file programs in Python (and prep from a course like CS 51 can help). No background in economic theory is assumed, although students without some economics background would benefit from a course that provides exposure to the basic ideas of utility maximization and rational behavior (e.g., from an AI course). Generally, we suggest students have completed *one of* Computer Science 181, Computer Science 182, Economics 1011a, Economics 1052, Economics 1071, or Economics 2056.

## Book

The instructor is developing a textbook with Sven Seuken (U. Zurich), hopefully to be published by Cambridge University Press in early 2024, and we will follow a draft of the book in class. Chapters will be distributed during the course. [Networks, Crowds, and Markets: Reasoning About a Highly Connected World](#), by David Easley and Jon Kleinberg, also provides a good introduction to many topics (this is not required reading).

**beginning of class.** It is very important to keep on top of the reading, and familiarity with the reading will be assumed during lecture. **You should set aside two hours before each class for the reading.**

I do not expect you to fully understand everything before coming to class. Rather, the goal you should set for yourself is to familiarize yourself with new terminology and definitions, and be ready to ask about things that you find especially confusing. **It is strongly recommended to attend class** (lecture recordings will only be available 2-3 days after class).

## Support Resources

*Online Q+A.* We will be using Ed. **All announcements will be made on Ed**, and we encourage you to answer each other's questions (but without giving away answers to homework questions).

*Office hours.* Please make use of staff office hours! You are welcome to come with specific questions about the material, to discuss final project ideas, or just chat about things you find interesting.

## Philosophy

You can expect the course staff to work hard to make the course useful for you, be available throughout the semester and look forward to meeting you in person, promptly answer your questions, and return assignments and midterms to you in a timely manner. We expect that you will, in turn, be prepared to play an active role in your learning, and come to class prepared (having done the assigned reading), and participate in classroom discussions. In this way, we can use class time to work together to develop a deeper understanding of the material.

CS 136 is a challenging class because it covers interdisciplinary material. But we'll work together to try to get as much understanding as possible of the underlying principles and methodologies. If you feel confused about something and you have given yourself some time to try to understand on your own, you should seek help by discussing it with other students or with the staff, or asking a question in the classroom. Other students might well have the same confusion!

## Learning outcomes

After successful completion of this course, you will be able to...

### Apply principles of economics and computation to

- Understand, model and analyze the interplay between incentives and computation in the design of online platforms
- Solve mathematical and conceptual problems involving such systems, including problems you have not seen before

## Organize your learned knowledge in Econ/CS, by

- Articulating the big ideas from each lecture
- Making connections between different concepts
- Synthesizing concepts with ideas from other branches of computer science, including machine learning, complexity theory, and artificial intelligence
- Having an appreciation for the way in which computational systems influence, and are influenced by, the aggregate behavior of groups of people

## Collaborate and communicate clearly, in particular

- Justify your thinking by clearly articulating the principles that apply in a given setting
- Explain your solutions, either written or orally to peers and instructors

## Logistics

### Sections

There will be a recommended, but optional, 60 minute section each week. During section, you will work in groups to solve exercises that are designed to prepare you for the homework and midterms. The TF is there to help guide you and clarify confusions.

### Assignments

There are two types of assignments: theory and programming.

**Theory assignments** include qualitative discussion questions as well as formal, theoretical analysis. The qualitative questions will help you gain an understanding of the assumptions behind the models that we discuss, and the connections to real-world problems. The theoretical questions will teach you to formally analyze simplified economic and computational systems, and to reason precisely. Some problems will require you to write proofs. **You should work by yourself on the theory assignments.**

**Programming assignments** are mainly in Python, and will let you experiment with simplified models of real problems. These flow into contests (and very good performance can lift up a grade at a grade boundary), where we pit submissions against each other. The assignments are designed to give you the chance to be creative and experiment with different ideas. **You are strongly encouraged to work in pairs on the programming assignments.**

### Midterms

T. ⓘ will be **two in-class midterms**--- one is half-way through the semester, and the second in the last week of the semester (and non-cumulative).

There is a class project. The goals of the project and are to allow you to explore independent interests, learn more about a particular area, and practice teamwork and presentation skills. The project will result in a final paper that is due during reading week. **Projects should be done in groups of 2 or 3, and may be primarily theoretical or primarily programming-based.**

A project can also involve an **exposition of a topic** related to topics covered in class, but in this case should **be done by a single student.**

## Grading

Problem sets are **due at 5pm on Fridays (with [two exceptions](#))**. Students have a maximum of six late days during the semester. There is a two day maximum on any pset, with no credit after two late days. For pair assignments, only one student needs the late days, but they will be counted against both students (if you have remaining credit!).

Submissions are to Gradescope. We will only give extensions for emergencies, and will need either a note from a doctor or your Resident Dean. In the case of regrade requests, the work will be regraded fully, and your grade may go up or down. **Regrade requests are due 1 week after the grades are released.**

We will evaluate the programming assignments according to the following criteria:


- Correctness: does it work?
- Write-ups: what did you do? Why? How well did it work? Why did you get the results you did? Is the writing clear?
- In class contests: How does your approach compare to those of your classmates? Doing well in contests will be worth a large amount of glory (!) but will only factor into your grade at grade boundaries.

We will evaluate your solutions to theory assignments according to the following criteria:

- Correctness: are your answers correct? This is precisely defined for mathematical questions, but includes having valid arguments in qualitative responses.
- Clarity: is your explanation clear? Is the main argument muddled by a lot of irrelevant discussion?

Your final grade will be determined as follows:

- Assignments: 45%
- In-class midterms: 30%
- Final project: 15%
- Participation/completing reading comments: 10%

**E  theory assignment will be equally weighted. The bottom theory assignment will not be counted in the final tally. Every student must attempt every assignment!**

The course philosophy is that we want you to work together and feel comfortable talking about the problem sets. Discussion and the exchange of ideas are essential to doing academic work. However, at the same time, we want to be able to grade your own work, and not the work of your friends. Please reach out to David Parkes if anything is unclear or you have concerns.

**Comprehension Questions:** You can discuss the reading with others but your responses should be your own.

**Single-person Assignments (theory):** For the theory assignments, you are encouraged to discuss with your classmates as you work on problem sets. **However, you should not share answers**, and after discussions with peers, you should make sure that you can work through the problem yourself and ensure that any answers you submit for evaluation are the result of your own efforts. In addition, you must cite any books, articles, websites, lectures, and such that have helped you with your work. Similarly, you must list the names of students with whom you have collaborated on problem sets.

**Group assignments (programming):** For the programming assignments, we expect you and your partners **to design and implement the solutions together**. You may also discuss with your classmates in other groups, **but not share code or answers**. Make sure that your group can work through the problem yourself, and ensure that any answers you submit are the results of your own efforts. In addition, you must cite any books, articles, websites, lectures, and such that have helped you with your work, and list the names of students from other groups with whom your group has collaborated.

**Class project:** You are encouraged to consult with your classmates on the choice of topic and to share resources. You may also find it useful to discuss your class project with other students. Otherwise, the guidelines are as above, depending on whether your project is a single-person or multi-person effort.

**NEW Generative AI/LLM policy (1 of 2):** If you find it helpful, you may use LLMs in answering problem sets, both theoretical and programming based, just as you may consult a text book or web source. As there, you should cite any tools that you use, and as there, **you should be able to provide a full explanation of your answers, so that these tools are used as a learning aid but not a replacement for your own learning**. You will not be able to access LLMs during the in-class midterms.

**NEW Generative AI/LLM policy (2 of 2):** For your end of class projects, you may choose to use LLMs as a tool to help with your writing, but as with all other work, you are responsible for the content of your paper, including its correctness and that it does not plagiarize other sources. In particular, you must have a full understanding of the paper that you turn in, and be able to explain its content (away from your AI helper!). Also, it is essential that you cite any tools that you use, **including a few sentences to explain how you used them**.