

COS 226, FALL 2026

***ALGORITHMS***  
and  
***DATA STRUCTURES***

**MARCEL DALL'AGNOL · BOB TARJAN · KEVIN WAYNE**



PRINCETON  
UNIVERSITY

# FINE PRINT



## *Lecture recordings:*

*We will record the slides and instructor audio and post recordings in Canvas.*



## *Student recording/sharing:*

*For privacy, compliance, and legal reasons, students may not record class sessions or redistribute lecture recordings.*





<https://algs4.cs.princeton.edu>

## INTRO TO COS 226

---

- *motivation*
- *course structure*
- *grading and assessments*
- *support and resources*



<https://algs4.cs.princeton.edu>

# INTRO TO COS 226

---

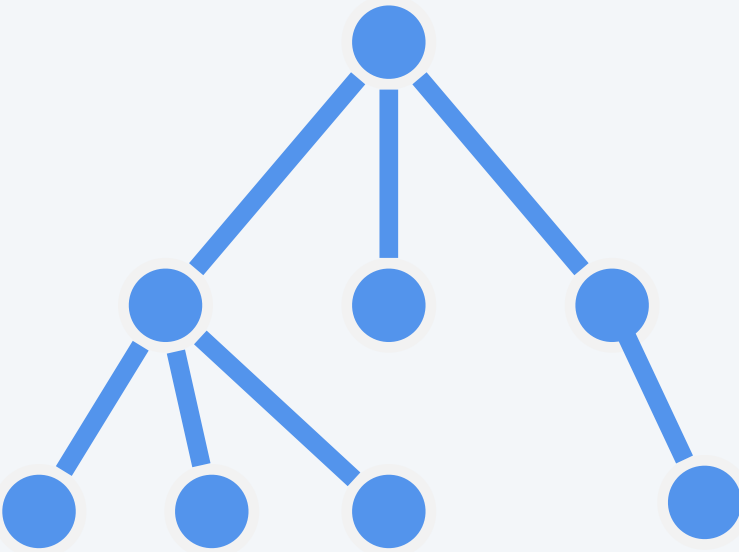
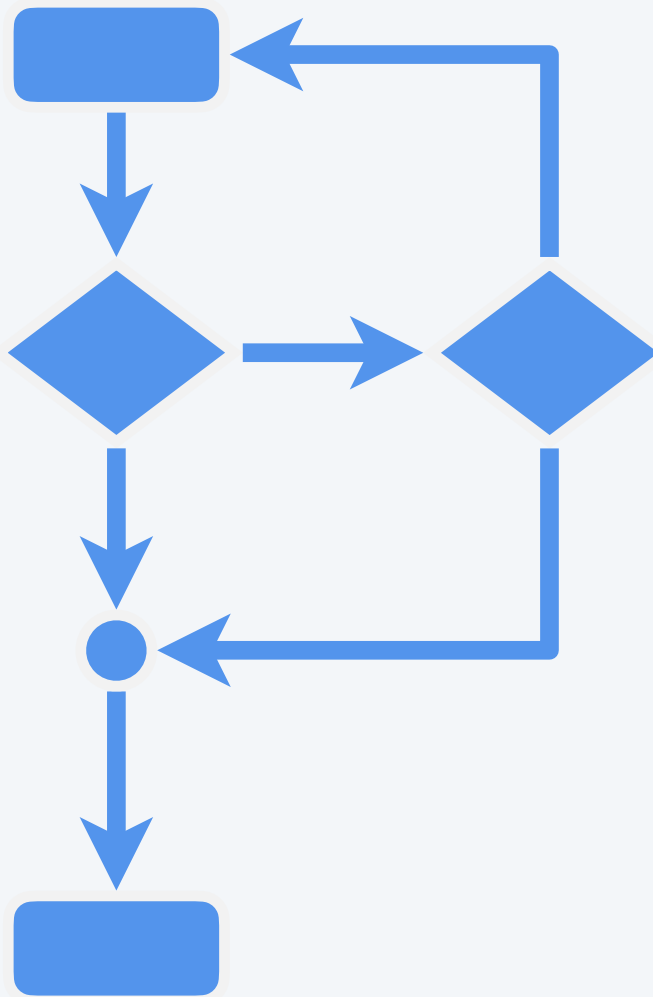
- *motivation*
- *course structure*
- *grading and assessments*
- *support and resources*

# COS 226 course overview

## What is COS 226?

- **Algorithm:** a step-by-step, well-defined procedure for solving a problem.
- **Data structure:** a way to organize data in a computer for efficient access and update.
- Programming, problem solving, and applications.

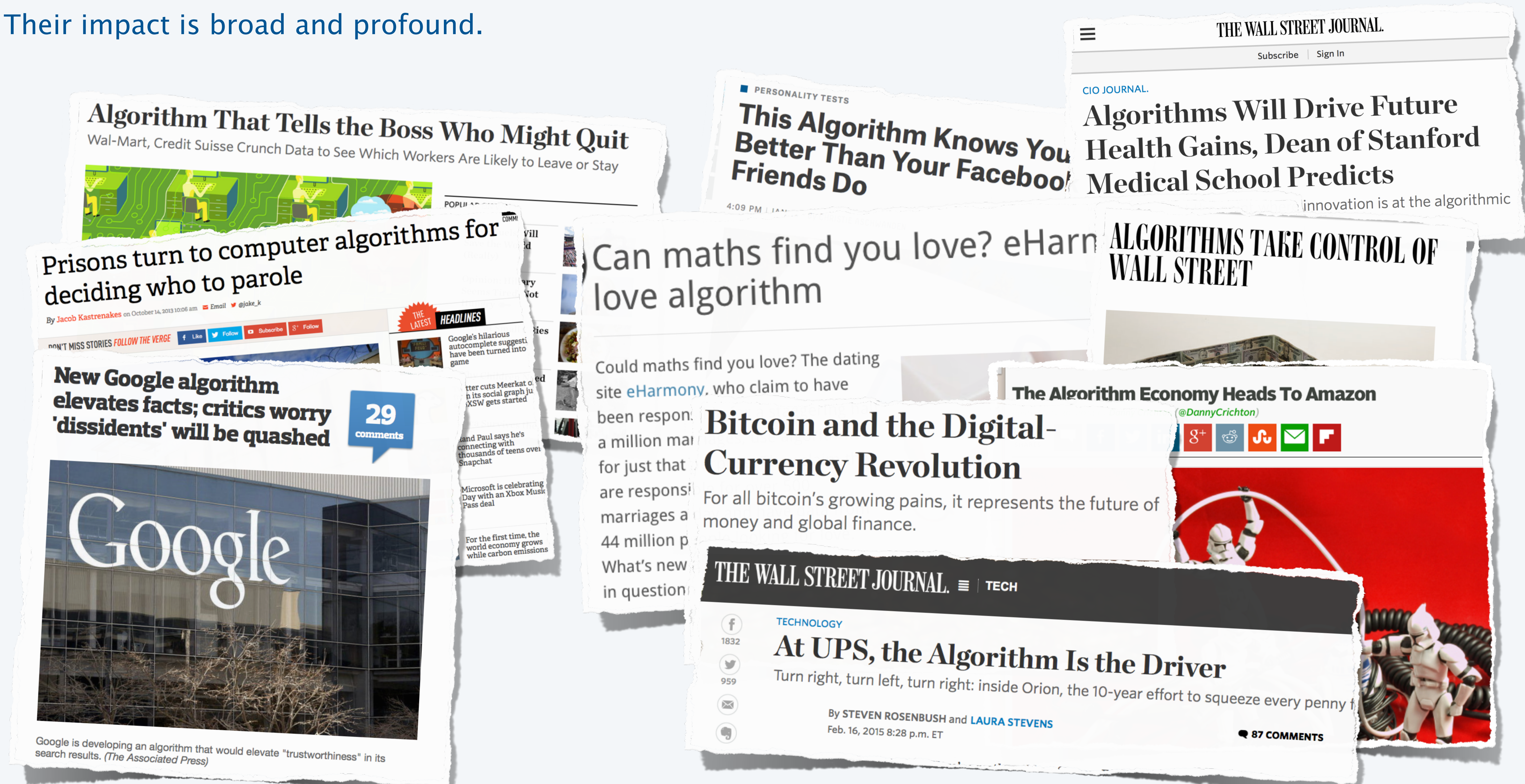
| topic           | algorithms and data structures                              |
|-----------------|-------------------------------------------------------------|
| data types      | <i>stack, queue, union-find</i>                             |
| sorting         | <i>insertion sort, quicksort, mergesort, priority queue</i> |
| searching       | <i>BST, red-black tree, hash table, k-d tree</i>            |
| graphs          | <i>BFS, DFS, Prim, Kruskal, Dijkstra</i>                    |
| advanced topics | <i>intractability, multiplicative weights</i>               |





# Why study algorithms and data structures?

Their impact is broad and profound.

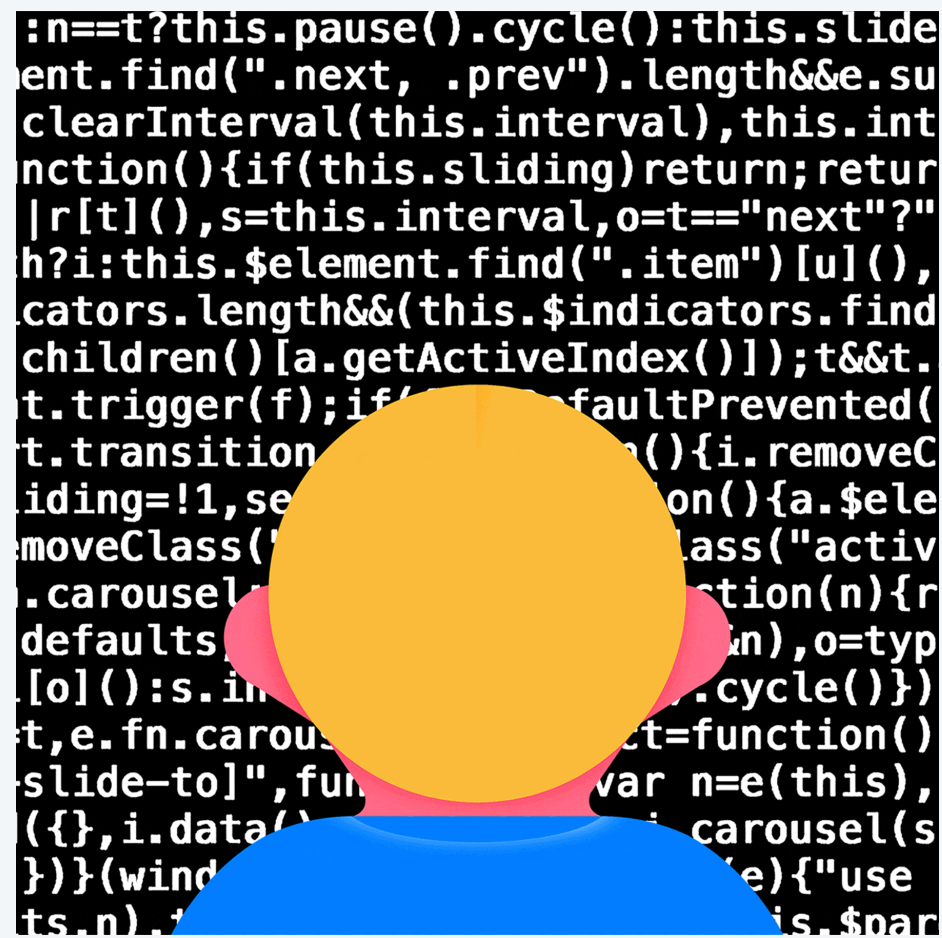




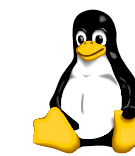
# Why study algorithms and data structures?

---

To become a strong programmer.



*“ I will, in fact, claim that the difference between a bad programmer and a good one is whether [they] consider [their] code or [their] data structures more important. Bad programmers worry about the code. Good programmers worry about **data structures** and their relationships. ”* — Linus Torvalds (creator of Linux and Git)



# Why study algorithms and data structures?

---

For intellectual stimulation.



*“For me, **great algorithms are the poetry of computation.** Just like verse, they can be terse, allusive, dense, and even mysterious. But once unlocked, they cast a brilliant new light on some aspect of computing.”* — Francis Sullivan

# Why study algorithms and data structures?

For fun and, eventually, profit.





# Why study algorithms and data structures?

---

- Their impact is broad and profound.
- To become a strong programmer.
- For intellectual stimulation.
- For fun and, eventually, profit.

Why study anything else?







<https://algs4.cs.princeton.edu>

# INTRO TO COS 226

---

- *motivation*
- *course structure*
- *grading and assessments*
- *support and resources*

# Lectures

---

Attend lectures. New content.

Download lecture slides. Course website → Lectures.

Watch lecture recordings. Canvas → Panopto Video.

Co-instructors.



Kevin Wayne



Marcel Dall'Agnol

Questions. Please raise your hand and ask. ← *carpe diem!*

Electronic devices. Use only for class (slides, notes, iClicker).



iClicker. To earn participation credit:

- Sign in or create an iClicker Cloud account.  *use your Canvas-preferred email*
- Register for this course.
- Answer multiple-choice questions during lecture.



<https://www.iclicker.com>

Which would you most like to have more of?

A. Fame 

B. Fortune 

C. Friends 

D. Free time 



# Precepts

---

**Active learning.** Problem solving, discussion, assignment preparation, and more.



Prof. Marcel Dall'Agnol



Jason Greenfield



Gary Hu



Narutatsu Ri

**Special precept P05.** [Friday 10:40–12pm]

- Intended for students seeking a more advanced treatment of the material.
- Includes topics and theory beyond the scope of the course.
- Add/drop via Stellic.
- Questions? Ask Prof. Tarjan after class today.



Prof. Bob Tarjan  
(Turing Award '86)



<https://algs4.cs.princeton.edu>

# INTRO TO COS 226

---

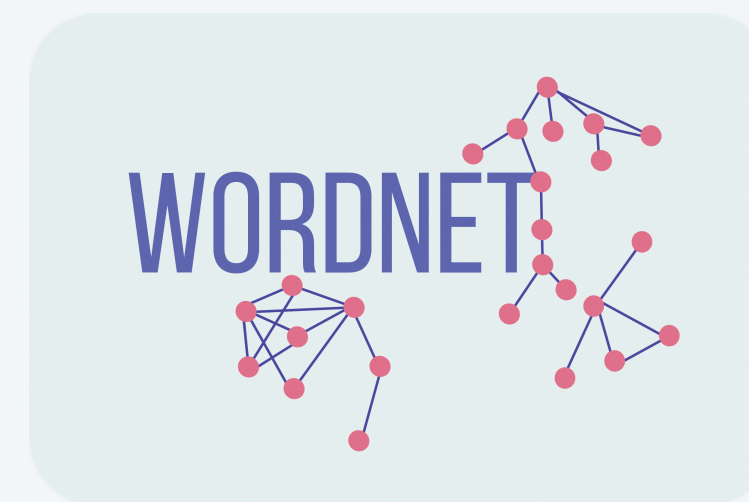
- *motivation*
- *course structure*
- *grading and assessments*
- *support and resources*



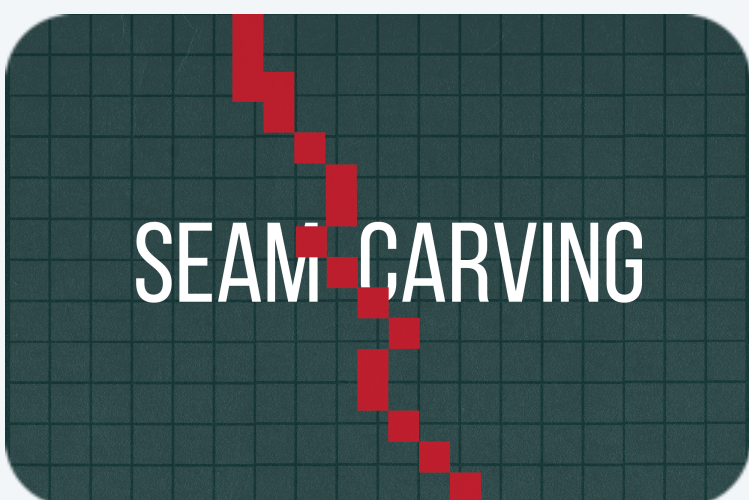
# Programming assignments

---

Implement efficient **algorithms** and **data structures**.



Apply textbook algorithms to realistic **applications**.



**Pair programming** is encouraged on designated assignments.

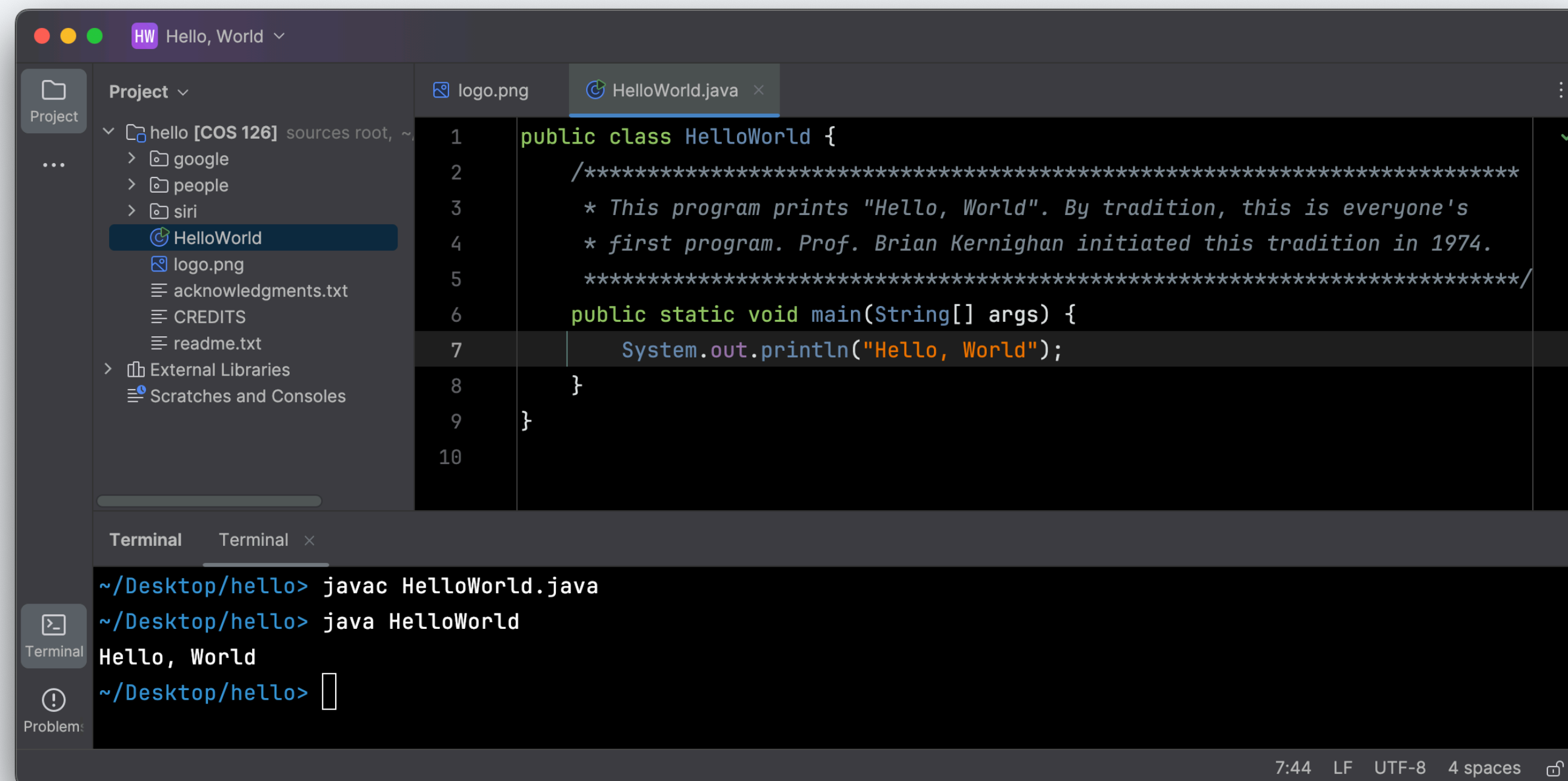




# Programming environment (IntelliJ)

**Recommended IDE:** custom IntelliJ 2026.2 environment.  *use our Fall 2026 version  
(see lab TAs for troubleshooting)*

- Embedded terminal (Bash).
- Highlights common errors as you type.
- Automatic formatting, imports, and code completion.
- ...



# Code reviews

---

**Goal.** Demonstrate understanding of your code; get feedback on design and style.

**Format.** One-on-one interview with a TA.

**Duration.** 20 minutes.

**Resources.** Closed notes, but you may refer to (and run) your code.



## Quiz format. [mastery-based learning]

- 2–3 short questions per lecture.
- Use pencil and paper (not AI).
- Unlimited attempts per question.  *different variant each attempt;  
short delay between attempts*
- Your score = best score over all attempts.

## Fall 2026 PrairieLearn study.

- Text-based vs. interactive question formats.
- First quiz includes the study consent form.
- Student researcher: Adham Ibrahim '27.
- Princeton IRB #19969.



Exam format. [summative assessment]

- Pencil and paper; closed book; one-page reference sheet allowed.
- Questions drawn from lectures, precepts, programming assignments, and quizzes.
- Emphasizes **conceptual** understanding and reasoning over code-writing.

COS 226 MIDTERM, SPRING 2023

3

3. Data structures. (6 points)

(a) Consider the following *parent-link* representation of a *weighted quick union* (link-by-size) data structure.

parent[]

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 4 | 5 | 4 | 5 | ? | 5 | 2 | 5 | 8 | 5 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

Which of the following values could be `parent[4]`?

Fill in all checkboxes that apply.

☐

0

☐

1

☐

2

☐

3

☐

4

☐

5

☐

6

☐

7

☐

8

☐

9

# Grading A+

Programming assignments (7). Due Mondays at 11:59pm.

Code reviews (3). Schedule during your assigned weeks.

Quizzes (12). Due Saturdays at 11:59pm.

Midterm exam. October 15 (in lecture).

Final exam. December 18, 8:30–11:30am.

Participation. Lecture and precept engagement; meet the professor.

Course grade. No curves; fixed cutoffs are applied to your weighted average.

| Graded component               | Weight |
|--------------------------------|--------|
| <i>Programming assignments</i> | 15%    |
| <i>Code reviews</i>            | 15%    |
| <i>Quizzes</i>                 | 15%    |
| <i>Midterm exam</i>            | 20%    |
| <i>Final exam</i>              | 30%    |
| <i>Participation</i>           | 5%     |

| Grade  | A   | A–  | B+  | B   | B–  | ... |
|--------|-----|-----|-----|-----|-----|-----|
| Cutoff | 93% | 90% | 87% | 83% | 80% | ... |



<https://algs4.cs.princeton.edu>

# INTRO TO COS 226

---

- *motivation*
- *course structure*
- *grading and assessments*
- *support and resources*





**Textbook (required).** *Algorithms, 4th Edition* (Sedgewick and Wayne).

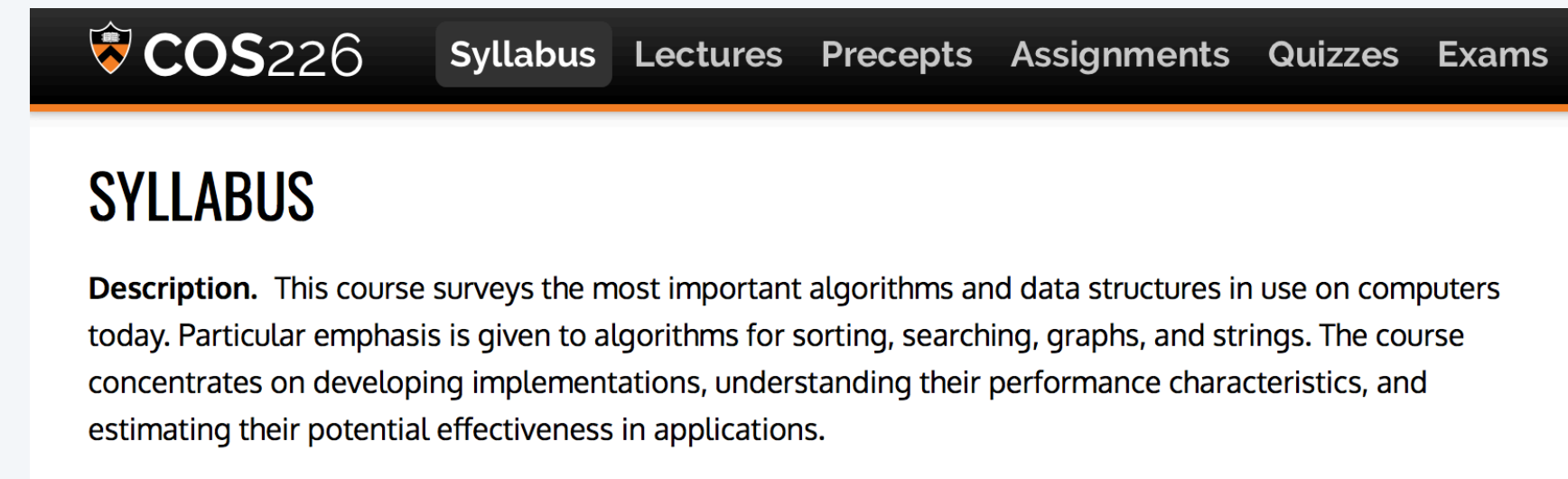
- Written to accompany COS 226.
- Read the assigned section before lecture.
- Available via Princeton eCampus bookstore.



**4<sup>th</sup> edition (2011)**

## Course website.

- Syllabus and course policies.
- Lecture slides.
- Precept worksheets.
- Programming assignments.
- Quizzes.
- Exam archive.

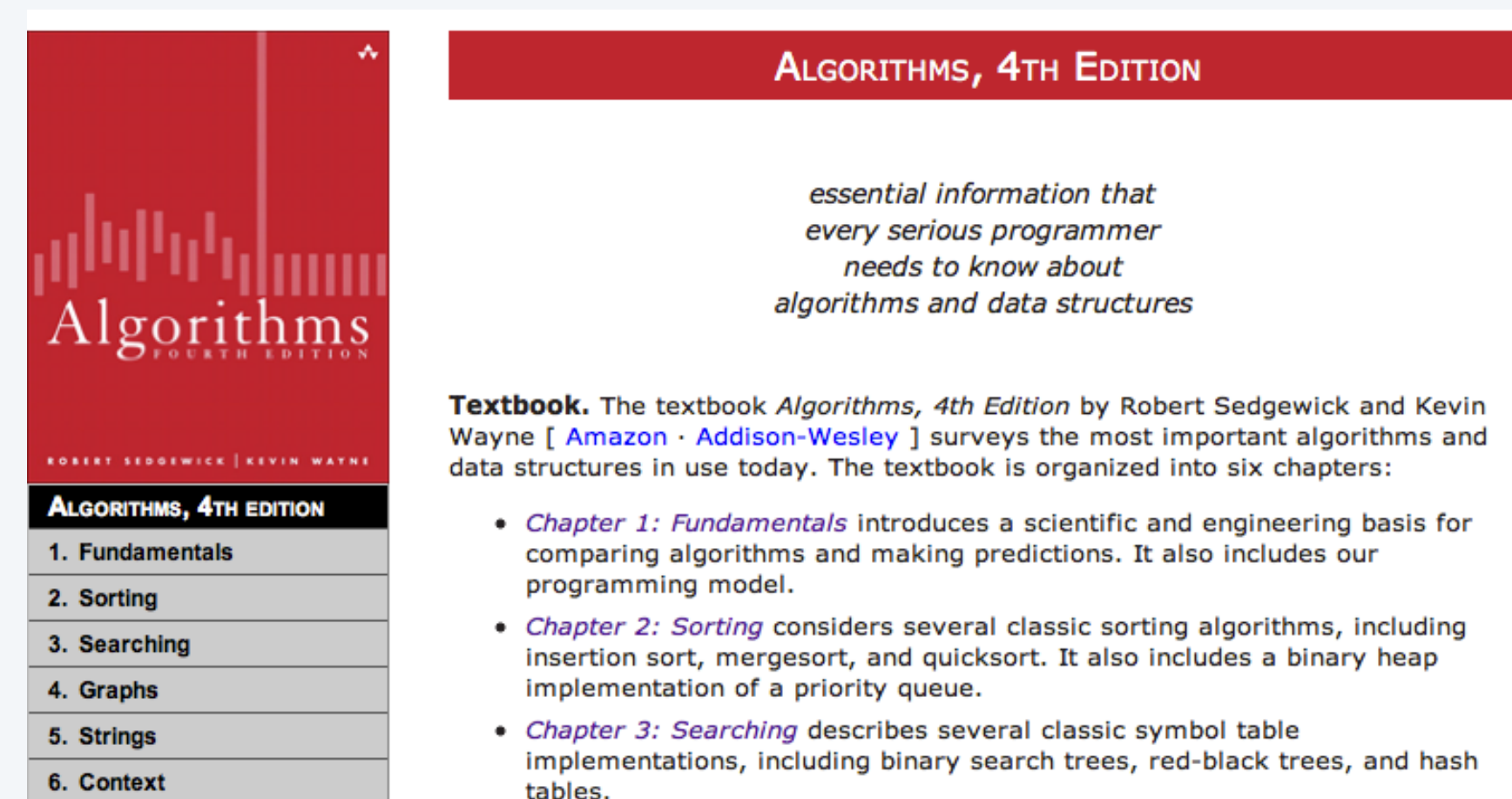


The screenshot shows the COS226 course website. At the top is a navigation bar with links: Syllabus, Lectures, Precepts, Assignments, Quizzes, and Exams. Below this is a section titled "SYLLABUS". The description states: "This course surveys the most important algorithms and data structures in use on computers today. Particular emphasis is given to algorithms for sorting, searching, graphs, and strings. The course concentrates on developing implementations, understanding their performance characteristics, and estimating their potential effectiveness in applications."

<https://www.princeton.edu/~cos226>

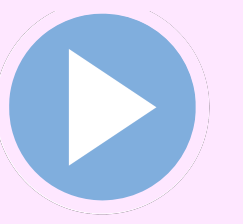
## Textbook website (booksite).

- Brief chapter summaries.
- Code from the book.
- APIs and Javadoc.



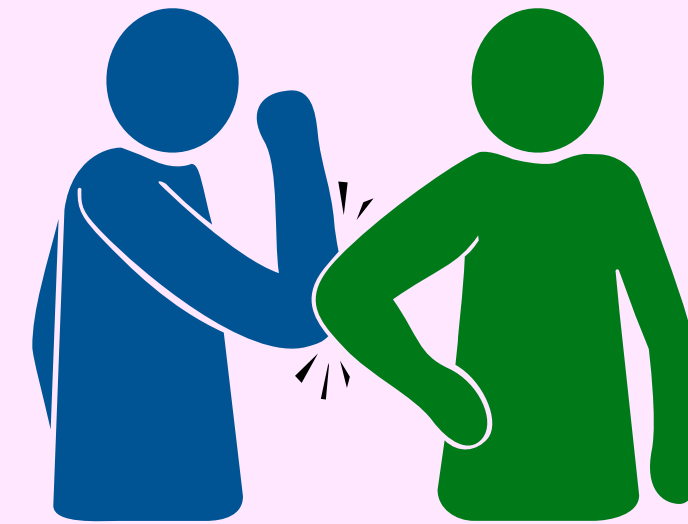
The screenshot shows the website for the textbook "Algorithms, 4th Edition" by Robert Sedgwick and Kevin Wayne. The page features the book cover on the left, which has a red background and the title "Algorithms" in white. To the right of the cover, the text reads: "essential information that every serious programmer needs to know about algorithms and data structures". Below this, a "Textbook" section states: "The textbook *Algorithms, 4th Edition* by Robert Sedgwick and Kevin Wayne [ [Amazon](#) · [Addison-Wesley](#) ] surveys the most important algorithms and data structures in use today. The textbook is organized into six chapters:" followed by a list of chapters: Chapter 1: Fundamentals, Chapter 2: Sorting, and Chapter 3: Searching.

<https://algs4.cs.princeton.edu>



Meet a classmate near you (someone you don't know).

- Introduce yourself.
- Explain why you're taking COS 226.
- Share  $\Theta(1)$  fun facts about yourself.
- Exchange contact info (optional, but no LinkedIn).





## Ed discussion forum. *please use Ed, not email*

- For quick questions.
- Make a post private only when necessary.
- Read the Ed FAQ for guidelines.



<https://us.edstem.org/courses/102391>

## Office hours.

- For longer discussions.
- See the course website for the schedule.
- Please come visit us!



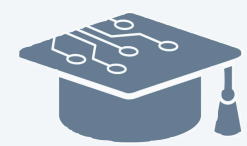
<https://www.princeton.edu/~cos226>

## Intro COS lab.

- Run by undergrads.
- For help with debugging.
- See the course website for the schedule.



<https://introlab.cs.princeton.edu>



| Platform                                                                                              | What it's for                        |
|-------------------------------------------------------------------------------------------------------|--------------------------------------|
|  <b>Ed</b>           | <i>discussion forum</i>              |
|  <b>IntelliJ</b>     | <i>Java programming environment</i>  |
|  <b>PrairieLearn</b> | <i>quizzes</i>                       |
|  <b>TigerFile</b>  | <i>assignment submission</i>         |
|  <b>Gradescope</b> | <i>assignment and exam feedback</i>  |
|  <b>Canvas</b>     | <i>grades and lecture recordings</i> |
|  <b>iClicker</b>   | <i>in-lecture polls</i>              |

← *also use Ed to communicate with course staff  
(not email)*

# Generative AI + learning

## Randomized controlled trial. [Bastani et al., PNAS 2025]

- Subject: high school math.
- Three groups of students; same exams.

| student group           | exam scores (no GenAI) | perceived learning                      |
|-------------------------|------------------------|-----------------------------------------|
| <i>no GenAI</i>         | <i>baseline</i>        | <i>baseline</i>                         |
| <i>GPT</i>              | <i>−17%</i>            | <i>felt they learned about the same</i> |
| <i>GPT + guardrails</i> | <i>≈ 0%</i>            | <i>felt they learned more</i>           |

Takeaway 1. GenAI without guardrails can harm learning.

Takeaway 2. GenAI can distort your perception of how much you learned.

PNAS

ECONOMIC SCIENCES

Generative AI without guardrails can harm learning: Evidence from high school mathematics

Hamsa Bastani<sup>a,b,1</sup> , Osbert Bastani<sup>c,1</sup>, Alp Sungu<sup>a,1,2</sup> , Haosen Ge<sup>b</sup> , Özge Kabakcı<sup>d</sup>, and Rei Mariman<sup>e</sup>

Affiliations are included on p. 7.

Edited by Emma Brunskill, Stanford University, Stanford, CA; received November 3, 2024; accepted May 5, 2025 by Editorial Board Member Mark Granovetter

Generative AI is poised to revolutionize how humans work, and has already demonstrated promise in significantly improving human productivity. A key question is how generative AI affects learning—namely, how humans acquire new skills as they perform tasks. Learning is critical to long-term productivity, especially since generative AI is fallible and users must check its outputs. We study this question via a field experiment where we provide nearly a thousand high school math students with access to generative AI tutors. To understand the differential impact of tool design on learning, we deploy two generative AI tutors: one that mimics a standard ChatGPT interface (“GPT Base”) and one with prompts designed to safeguard learning (“GPT Tutor”). Consistent with prior work, our results show that having GPT-4 access while solving problems significantly improves performance (48% improvement in grades for GPT Base and 127% for GPT Tutor). However, we additionally find that when access is subsequently taken away, students actually perform worse than those who never had access (17% reduction in grades for GPT Base)—i.e., unfettered access to GPT-4 can harm educational outcomes. These negative learning effects are largely mitigated by the safeguards in GPT Tutor. Without guardrails, students attempt to use GPT-4 as a “crutch” during practice problem sessions, and subsequently perform worse on their own. Thus, decision-makers must be cautious about design choices underlying generative AI deployments to preserve skill learning and long-term productivity.

generative AI | education | skill acquisition | personalized tutoring

Generative AI, such as OpenAI’s ChatGPT, has rapidly emerged as a disruptive technology capable of achieving human-level performance on a broad range of tasks (1–5). In many applications, they are expected to augment humans to help them perform tasks effectively and efficiently (6). Recent studies have sought to better understand how humans work in collaboration with these tools (7–9). Broadly speaking, these studies have focused on productivity, finding that workers can perform knowledge-intensive tasks significantly more efficiently when given access to generative AI.

A key question that remains is how generative AI affects how humans learn novel skills, both in educational settings and through the course of performing their jobs. This process of skill acquisition is critical for safeguarding long-term productivity (10). However, many generative AI deployments are designed to automate tasks without consideration for impact on learning. When technology automates a task, humans can miss out on valuable experience, inducing a tradeoff where the technology improves performance on average but introduces new failure cases due to reduced human skill. For example, overreliance on autopilot led the Federal Aviation Administration to recommend that pilots minimize their use of this technology (11); their precautionary guidance ensures that pilots have the necessary skills to maintain safety in situations where autopilot fails to function correctly. The potential for generative AI to interfere with learning is especially concerning due to the inconsistent reliability of this technology; for instance, while generative AI has demonstrated tremendous capabilities such as strong performance on medical exams (2) and competitive programming (3), it continues to suffer from hallucinations where it provides confident but factually incorrect responses (12). As a consequence, users must vigilantly check its outputs and fix any issues present; if they fail to learn the underlying skills, then they may lack the expertise required to do so.

Simultaneously, there has also been interest in leveraging generative AI to improve learning (13, 14), e.g., by incorporating it into existing chatbots for personalized tutoring (15, 16). Given the broad capabilities of generative AI for natural language understanding of tasks (1), the hope is that generative AI tutors can automatically

Significance

While generative AI has been shown to enhance productivity, its influence on learning new skills remains unclear. Our research examines the impact of generative AI, specifically GPT-4, on student learning in math education. Through a large-scale field experiment in a high school, our study reveals that although AI-based tutoring improves performance during practice sessions, students relying on the technology may underperform when access to AI is subsequently removed, indicating reduced skill acquisition. However, we also find that carefully designed safeguards, especially asking the AI tutor to provide teacher-designed hints instead of giving away answers, can mitigate these negative effects. Our findings highlight the need for thoughtful integration of generative AI in educational settings to ensure that human learning is preserved.

Author contributions: H.B., O.B., and A.S. designed research; H.B., O.B., and A.S. performed research; O.B. and H.G. contributed new analytic tools; H.B., O.B., A.S., H.G., and R.M. analyzed data; A.S. and O.K. managed field activities; and H.B., O.B., and A.S. wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission. E.B. is a guest editor invited by the Editorial Board.

Copyright © 2025 the Author(s). Published by PNAS. This article is distributed under Creative Commons Attribution-NonCommercial-NoDerivatives License 4.0 (CC BY-NC-ND).

<sup>1</sup>H.B., O.B., and A.S. contributed equally to this work.

<sup>2</sup>To whom correspondence may be addressed. Email: alpsungu@wharton.upenn.edu.

This article contains supporting information online at <https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2422633122/-DCS/supplemental>.

Published June 25, 2025.

PNAS 2025 Vol. 122 No. 26 e2422633122

<https://doi.org/10.1073/pnas.2422633122> 1 of 8



**GenAI policy.** Using GenAI on any graded COS 226 work is **prohibited**.

**Exception.** Use COS 226 AI Bot in Ed Discussion.   *OpenAI + guardrails  
(tailored to course materials)*

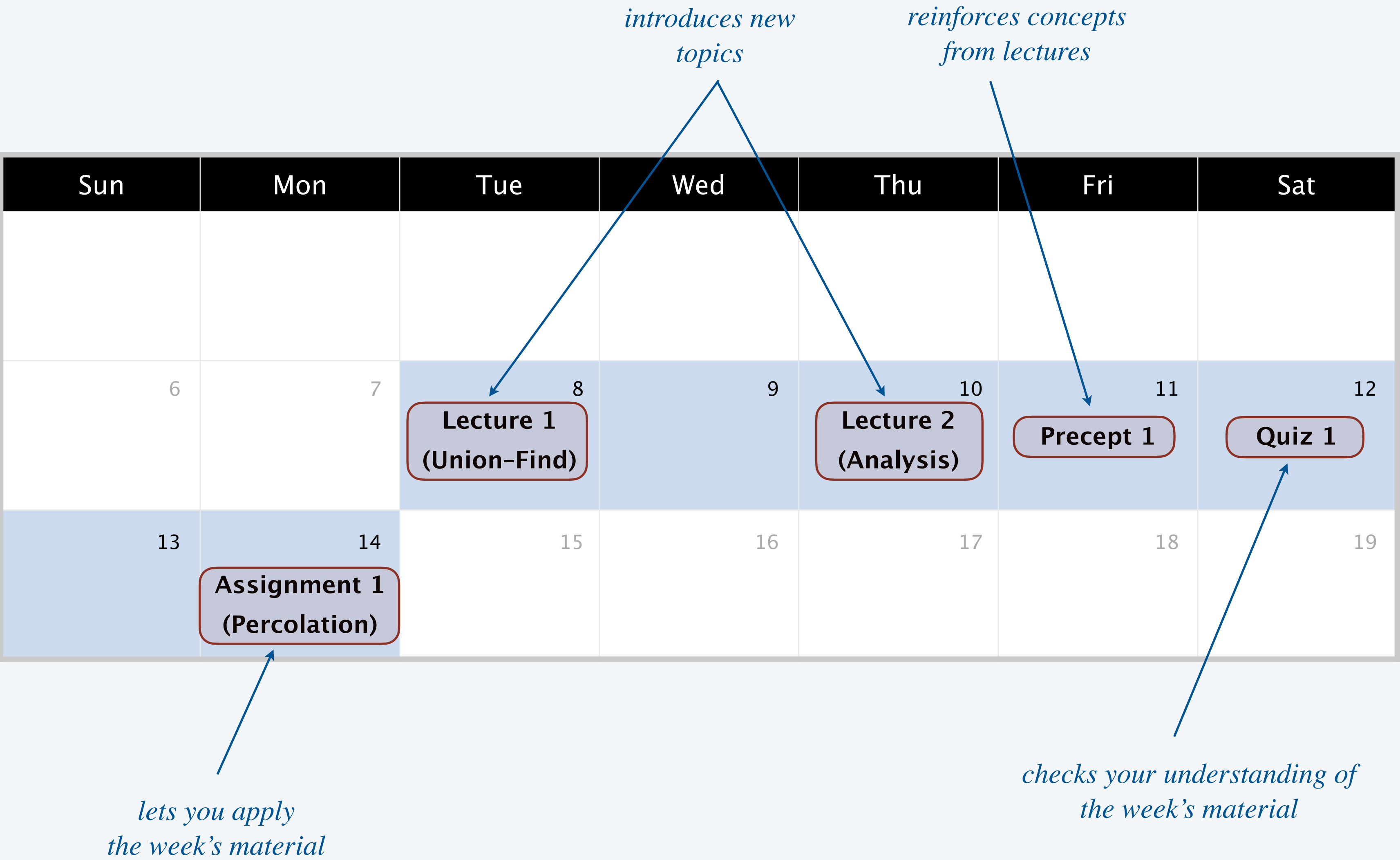


## Pedagogical rationale.

- Build a strong foundation in programming.
- Strengthen problem-solving and critical-thinking skills.
- Prepare to use GenAI effectively and responsibly in the future.
- Succeed on exams and technical interviews.

**Warning.** Substantive GenAI use on programming assignments is often easy to identify. 

# A typical week in COS 226 (starting next week)



## Administrative questions ?

---

Not yet registered? Register today.  *must be enrolled to access course platforms*

Change your precept? Use Stellic.

All nonconflicting precepts closed? Contact our course administrator, Kobi Kaplan.



Kobi Kaplan

Placed out of COS 126 (or took ECE 115)? Review Sections 1.1–1.2 of the textbook.

Need placement into COS 226? See the COS placement officer.

Additional administrative questions? Ask now, after class, or any time on Ed.





# Credits

---

| image                            | source                      | license                   |
|----------------------------------|-----------------------------|---------------------------|
| <i>THX Eclipse Deep Note</i>     | <u>THX Ltd.</u>             |                           |
| <i>Wireframe Tiger</i>           | Audrey Cheng '20            | by author                 |
| <i>Programmer</i>                | <u>Wall Street Journal</u>  |                           |
| <i>Student Raising Hand</i>      | <u>classroomclipart.com</u> | <u>educational use</u>    |
| <i>Hands Raising Smartphones</i> | <u>Adobe Stock</u>          | <u>education license</u>  |
| <i>A is for Algorithms</i>       | <u>comtechpass.com</u>      |                           |
| <i>Assignment Logos</i>          | Kathleen Ma '18             | by author                 |
| <i>Pair Programming</i>          | <u>Adobe Stock</u>          | <u>education license</u>  |
| <i>Code Review</i>               | <u>Adobe Stock</u>          | <u>education license</u>  |
| <i>Elbow Bump</i>                | <u>The Noun Project</u>     | <u>CC BY 3.0</u>          |
| <i>Countdown Timer</i>           | <u>YouTube</u>              |                           |
| <i>Office Hours</i>              | <u>clipground.com</u>       | <u>CC BY 4.0</u>          |
| <i>COS Lab TAs</i>               | <u>Pulkit Singh '20</u>     | by author                 |
| <i>Question Marks</i>            | <u>pikpng.com</u>           | <u>non-commercial use</u> |