Spring 2024

See below for prerequisites and how to apply! The course has no enrollment cap, but enrollment is only by permission of the instructor.

For general course inquiries, email the staff list: cs191-2024-staff@googlegroups.com.

For attendance-related issues, post on Ed (*Absences* category).

Instructor
Professor Harry Lewis, lewis@harvard.edu
I graduated from Harvard College in 1968, was away for 3 years, got my PhD here in 1974, joined the faculty, and never left, though I am technically retired now. I've had lots of interesting students, including Bill Gates and Mark Zuckerberg, and Harvard CS faculty Henry Leitner, David Malan, Michael Mitzenmacher, Rebecca Nesson, Stuart Shieber, and Salil Vadhan. Looking forward to having more like these this spring! If you're curious what computer science looked like here 50+ years ago, you might enjoy an article I wrote for *Harvard Magazine*: "A Science Is Born". I proudly serve on the board of EPIC, the leading American digital privacy organization.

Teaching Fellows

Ido Burstein, idoburstein@college.harvard.edu

I am a senior at Quincy house, studying computer science with a minor in economics. I took the class last year and am really excited to join this year as a TF! When I'm not working on problem sets, you'd probably find me at the Murr playing squash!

Emma Chen, yingchen@g.harvard.edu
I am a second-year Ph.D. student in computer science, co-advised by Professors Vijay Janapa Reddi and Prof. Pranav Rajpurkar. My research focuses on medical AI and its applications in healthcare. Prior to my Ph.D., I studied Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign for my undergraduate degree, followed by a Master's in Computer Science at Stanford University. I then worked for 2 years as a software engineer at Microsoft. In my free time, I write a weekly newsletter called Doctor Penguin Weekly with Dr. Eric Topol, covering the latest important research in medical AI. I am very proud of Doctor Penguin because I drew all the penguins myself (you can see them on our old website https://doctorpenguin.com/).

Larry Denenberg, larry@harvard.edu

Some ten years behind Harry, I graduated from the College in 1977, then PhD in 1984. I spent many years at Bolt Beranek & Newman, building networks and connecting them until we realized it was all one big Internet. I co-founded and sold a startup, but the CEO of the acquiring company fled to Namibia to avoid jail, taking everyone's money. I was at Nevo (v.i.) for a bit, then most recently at TripAdvisor until Corona killed travel.

Gauri Jain, gaurijain@g.harvard.edu
I'm a second year computer science PhD student working with Professor Milind Tambe on topics in AI in Public Health, specifically in maternal and child health in India. Before that I worked as a software engineer at Meta for 2 years on the Marketplace and Integrity teams. In my free time I enjoy playing a variety of sports and instruments!

Ken Ledeen, kledeen@nevo.com

I graduated from Harvard in 1967, concentrating in English while programming to pay the bills. Upon graduation I started a company to create medical computer applications, where I hired awesome developers (e.g. Harry Lewis). We were about 50 years too early. My subsequent career has largely been leading technology organizations, most recently Nevo Technologies (v.s.). It has been a blast to be part of something so exciting and profoundly transformative.

This is my sixth year helping Prof. Lewis with CS191. I look forwarding to seeing how you all continue to expand the digital revolution.

Matthew Lena, matthew.lena@hotmail.com

I entered Harvard College in 1978 and graduated (in Applied Math, the closest thing Harvard had to CS in the old days) in 2007 â€“ no doubt a great relief to my advisor, Harry Lewis, after 29 years. Along the way

I built database and operating systems, taught math and statistics, and became the world's second-foremost expert (of which there are only two anyway) on Phineas Gage. More recently I've done litigation consulting on patents and trade secrets â€“ who stole whose idea and so on. I am particularly interested in concurrency control, fault tolerance, and the lost art of performance analysis.

Sid Pardeshi, spardeshi@g.harvard.edu

I am a G2 MS / MBA Candidate in the class of 2024. An 8+ year ex-NVIDIA Software Architect and gaming veteran, I am also a Generative AI innovator with over 25 US patent applications to my name. When I am not tinkering with AI or working on his startup project (blitzy.ai), you will find me playing Need for Speed or Gran Turismo on my PS5.

Andrew Sabot, asabot@g.harvard.edu

I graduated from University of Toronto in 2019, specializing in Computer Science and Math. Here at Harvard I am a fifth-year Computer Science PhD student advised by H.T. Kung. I took this course in Spring 2020, returned to TF it in 2022 and 2023, and am excited to be part of it again.

Course description

This course examines papers every computer scientist should have read, from Aristotle to the present but especially from 1930 to 1980. **It is neither an introduction to computer science nor a survey.** It is meant to allow advanced students in computer science and related fields to gain a unified view of the discipline as a whole by freeze-framing key moments in its hectic and ever-ramifying evolution.

We will achieve this through careful reading of the papers plus discussions among the whole class and in small groups. On several days a paperâ€™s author will join us â€“ though these discussions have usually focused less on a paperâ€™s ideas than on the context in which they arose, and on the (generally complicated) trajectory of a great computer scientistâ€™s career.

Prerequisites and how to enroll

Discussions go best among students with varied backgrounds, so being a CS concentrator is not required, but they go poorly if some participants offer only uninformed opinions. It's completely fine if you know nothing about *some* areas of computer science, but not fine if you know little or nothing about many areas. Thus **enrollees should have passed at least three courses designated *advancedcs* on the Computer Science Concentration advising website, or have comparable background** through courses taken at other institutions. But the more CS you have under your belt when you take the course, the more you will get out of it, and I do hope to teach it again next year.

If you meet the prerequisite, or are unsure and wish to inquire, fill out the enrollment petition form. If approved, you will be added to the class when you have registered through the standard course registration process.

Learning objectives

- To identify the major subfields of computer science, their intellectual family tree, and the major figures and works of their birth and infancy.
- To be able to place current computer science research in the context of its intellectual lineage.
- To be able to present to an educated audience (not just computer scientists) some of the major ideas of computer science in a succinct and easy-to-understand way.

Grading

Grades will be based on preparation for class (attentive reading as reflected in your diary entries), attendance (scrupulous), discussion participation (thoughtful), and video submission(s). No pass-fail and no auditors (but occasional guests are welcome â€“ just let us know).

Canvas insists on turning administrative records such as attendance into meaningless numbers; we seem powerless to break it of this habit, so ignore its spurious totals and averages. The grades we issue at the end of the course are final; in no case will video grades or course grades be reconsidered.

Attendance

We meet Tuesdays and Thursdays, 2:15-3:30pm, in 114 Western Ave, Room 2111 (across the street from the SEC, second floor). Discussion is the heart and soul of the course, so **attendance is required** (unless you are unwell, of course!) and you are expected to be in class at **2:15pm sharp**. Let us know about any

planned absences, and follow up promptly if you have an unanticipated absence, by posting to Ed in the *Absences* category. If you are scheduled to be a discussion leader, mention that so we can find someone at your table to swap with you.

Readings

- We will read about fifty papers, primarily as ten- to twenty-page excerpts gathered in *Ideas That Created the Future* (MIT Press, available in both print and digital editions, â€œITCTFâ€ below). A few additional selections, and some videos we will be discussing, are linked from the course schedule.
- **I favor marking up actual pages in an actual book, but pdfs of the ITCTF selections may be downloaded from HOLLIS here.** Also at that link are ITCTF's introduction and a joint bibliography of works referenced in the various selections.
- There are some (mostly very minor) errata listed here. Most known errors have been corrected in the third printing. Further errata (however trivial) will be appreciated.
- For those interested, the full original papers are available here, and an extended list of worthwhile papers is here.

What will be expected of you

This course is unusual. Past enrollees have found it lively and fun as well as educational. You will learn a lot of computer science, but that is not our our main goal, which is to help you to see where great ideas come from and what kind of person creates them. And we want you to see yourself among them in the future.

There are no lectures; the bulk of your learning will come from reading the papers and discussing them in class. Most days we will discuss two papers.

- Though discussing the readings with others is not only OK but encouraged, you do need to *actually do the reading*. **Do not use AI tools to summarize the material for you**; if nothing else, sooner or later you'll embarrass yourself by repeating in class some robot's delusions.
- Reading time for the papers varies widely, with length a poor guide, so do yourself the favor of starting each class's reading early enough to be sure you can do it justice. Now and then a paper will appeal to you more than most, and you will be glad to have time to really drill down into it. If you find you don't have time for the reading â€“ limited though that burden is â€“ you will learn nothing from the course, you wonâ€™t be able to contribute, youâ€™ll be letting your classmates down, and you should drop.
- *Before* each class you will submit an electronic diary (visible to course staff only) with your notes on that dayâ€™s reading: points for discussion, things you couldnâ€™t figure out, an insight your classmates might miss. This might be a page of text, but thatâ€™s up to you; what we want to see is that youâ€™re thinking about what youâ€™ll contribute to the discussion. Your diary need not be fluid prose, or even complete sentences, but it must be your own words. **Under no circumstances may you use AI tools to prepare your diary entries.** Itâ€™s fine to revise your diary later but (to repeat) its initial submission must be *before* class.
- On a typical day, I will begin with remarks setting historical context. We will then discuss the papers in randomly assigned tables of about half a dozen. Course staff circulate and join in, but they do not lead the discussions. Instead, members of each table take turns being leader, kicking off discussion and moderating; each of you will lead about three times during the term.
- Days with authors visiting are largely devoted to Q&A from you (definitely not limited to the content of the papers). This is your opportunity to hear what it feels like to be part of a revolution â€“ sometimes realizing it at the time, often not.

Video project

A large part of your final grade will be a 5-minute video submission arguing for a paper that should have been included in the reading list but was not. **Be sure to read forward in the schedule to check that you're not proposing something we'll read later in the term!** (And note that there are a few readings on this syllabus that aren't in *Ideas that Created the Future*; these are off limits as well.)

Your objective is not to pick the perfect paper, but to make the best possible argument for it. The more interesting, non-obvious, and well-articulated is your presentation, the better, but most of all your job is to be persuasive. Think of it as a pitch, except you are pitching not an invention or a startup, but an important idea of computer science and the paper that articulated it.

A full description of the project is here, and some good videos from previous terms are linked here.

You can submit a video on March 22, on April 29, or both:

- If you submit a video by March 22, then about a week later you will receive a grade on the video and a projected final course grade.
  - As long as you continue meeting the regular course obligations (attendance, diaries, discussion leading) consistent with your efforts to that point, your final course grade will not be lower than this projected grade.
  - We will comment on your submission with an eye toward helping you do better on your second submission â€“ *which must pitch a different paper* â€“ should you choose to make it.
- You may also submit a video by April 29 (during reading period).
  - If you submit both times, then
    - *your two videos must pitch different papers,* and
    - your second video may (or may not) raise your final course grade, but cannot lower it.
  - There is no penalty for submitting only in the second round, but then you're giving yourself only one bite at the apple, and that bite won't have the benefit of our feedback on a first submission.
  - There will be little or no feedback (other than the grade) on second-round submissions.

Our objective is to provide as much educational value as we can, and a solid indication of what grade you can expect, consistent with limited staff time.

A note on use of AI tools: In your search for interesting and important papers in a particular area of CS, asking the course staff is always a good idea, and other students and faculty can certainly be helpful. You might even get help from an AI tool, as long as you are duly skeptical about what it tells you. But when it comes to preparing your video (including its narration) you should do it yourself. **AI-generated prose won't get you a good grade**; trust us on this. Plus, one of our goals is to give you experience preparing and delivering a presentation, and when you have to do that on the job, no crutches may be available to you.

Course plan (tentative and subject to revision)

Week 1

Class 0 (Tuesday, January 23, 2024): Course overview and participant responsibilities. Prepare for today by reading:

- *Today's reading and* "*Reader's companion*" *linked here*
- Joseph Weizenbaum. "ELIZA - A Computer Program for the Study of Natural Language Communication Between Man and Machine" (1966, ITCTF #27)

Class 1 (Thursday, January 25): Boolean logic

- *Reader's companion here*
- Aristotle. "Prior Analytics" (~300 BCE, ITCTF #1)
- George Boole. "An Investigation of the Laws of Thought" (1853, ITCTF #4)
- Claude Shannon. "A Symbolic Analysis of Relay and Switching Circuits" (1938, ITCTF #8)

Week 2

Class 2 (Tuesday, January 30): Hardware I

- *Reader's companion here*
- Luigi Menabrea and Ada Lovelace. "A Sketch of the Analytical Engine Invented by Charles Babbage" (1842, ITCTF #3)
- (For context) Computing's cranky pioneer, HRL's review of IB Cohen's biography of Aiken. (Your diary need not address this -- though it could.)
- Howard Aiken. "Proposed Automatic Calculating Machine" (1938, ITCTF #7). This is the funding proposal for the "Aikenâ€“IBM Sequence Controlled Calculator â€“ Mark I", the hulking machine the greets you on your way to lunch at the SEC.

Class 3 (Thursday, February 1): Hardware II

- *Reader's guides here*
- John von Neumann. "First Draft of a Report on the EDVAC" (1945, ITCTF #10)
- Maurice Wilkes. "The Best Way to Design an Automatic Calculating Machine" (1951, ITCTF #15)

Week 3

Class 4 (Tuesday, February 6): Computability and uncomputability

- *The HOLLIS link to ITCTF has been down! Get today's readings, and the reader's companion, here.*
- David Hilbert. "Mathematical Problems" (1900, ITCTF #5)
- Alan Turing. "On Computable Numbers, with an Application to the Entscheidungsproblem" (1936, ITCTF #6)

Class 5 (Thursday, February 8): Information

- Claude Shannon. "A Mathematical Theory of Communication" (1948, ITCTF #12)
- Richard Hamming. "Error Detecting and Error Correcting Codes" (1950, ITCTF #13)

Week 4

Class 6 (Tuesday, February 13): Learning

- (For context) Amanda Gefter. "The Man Who Tried to Redeem the World with Logic" (2015)
- Warren McCulloch and Walter Pitts. "A Logical Calculus of the Ideas Immanent in Nervous Activity" (1943, ITCTF #9)
- Frank Rosenblatt. "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain" (1958, ITCTF #18)

Class 7 (Thursday, February 15): Computers for people

- Reader's companions here.
- Vannevar Bush. "As We May Think" (1945, ITCTF #11)
- J. C. R. Licklider. "Man-Computer Symbiosis" (1960, ITCTF #20) *The first and second printings of ICTCF include a significant erratum; see here*. This has been corrected in the third printing.

Week 5

Class 8 (Tuesday, February 20): Graphics and Interaction **with special guest Ivan Sutherland**

- *Reading (and viewing) guidance here.*
- For today's class, the videos are most important:
  - A 4-minute video of Sketchpad in operation. The video footage itself is original, but was edited and narrated (by Alan Kay) for a 25th-anniversary celebration in 1987. (A 15-minute video, with more detail, is here.)
  - A short video (a few minutes) recently posted by the BBC about Sutherland's head mounted display being the ancestor of the Apple virtual reality headset. The narrator of the segment about Sutherland's device is HRL's Harvard classmate Bob Sproull, who as an undergraduate helped engineer it.
  - Engelbart's famous "Mother of All Demos" (1968, six years after Sketchpad) is an hour long, but well worth the time. It had enormous impact.
- Readings:
  - Douglas Engelbart. "Augmenting Human Intellect: A Conceptual Framework" (1962, ITCTF #22). This is Engelbart thinking through the problem before any of the software seen in the video had even been designed. If you watched the video, you might just skim this.
  - But definitely read Ivan Sutherland's "Sketchpad: A Man-Machine Graphical Communication System" (1963, ITCTF #24), which includes in vestigial form many ideas we now take for granted.
  - Ivan Sutherland. "Technology and Courage" (1982). In the past we have read this for the last class of the year along with other readings about ethics, morals, and character, but this year Ivan wanted to be able to talk to you about it when he visits.
  - Finally, if you haven't already read "A Science Is Born", read it now.

Class 9 (Thursday, February 22): Computers and thought

- *Reading guides here.*
- Leibniz. "The True Method" (1677, ITCTF #2)
- Alan Turing. "Computing Machinery and Intelligence" (1950, ITCTF #14)

Week 6

Tuesday, February 27, 11:59pm: Paper proposals due for March 22 (first-round) video submissions, if you plan to submit one. Full details on video project here.

Class 10 (Tuesday, February 27): Algorithms I

- *Reader's companion here.*

- Joseph Kruskal. "On the Shortest Spanning Subtree of a Graph" (1956, ITCTF #17)
- Gale and Shapley. "College Admissions and the Stability of Marriage" (1962, in the File system under "Gale and Shapley 1962.pdf")

Class 11 (Thursday, February 29): Programming

- Grace Hopper. "The Education of a Computer" (1952, ITCTF #16)
- John McCarthy. "Recursive Functions of Symbolic Expressions" (1960, ITCTF #21)

Week 7

Class 12 (Tuesday, March 5): Networks **with special guest Bob Metcalfe**

- Vinton Cerf and Robert Khan. "A Protocol for Packet Network Intercommunication" (1974, ITCTF #38)
- Robert Metcalfe and David Boggs. "Ethernet" (1978, ITCTF #41)

Class 13 (Thursday, March 7): Artificial Intelligence

- *Reader's guide here.*
- John McCarthy, Marvin Minsky, Nathaniel Rochester, and Claude Shannon. "A Proposal for the Dartmouth Research Project on Artificial Intelligence" (1955, in the File system here)
- Norbert Wiener. "Some moral and technical consequences of automation" (1960, ITCTF #19)
- Wiener, "A Scientist Rebels" (1947, in the File system here)
- "The Sorcerer's Apprentice" from Disney's *Fantasia* (1940, 9 minutes). Who is to blame here? The apprentice? The sorcerer?

<spring break>

Week 8

Class 14 (Tuesday, March 19): Software engineering

- *Reading guide here*
- Winston Royce. "Managing the Development of Large Software Systems" (1970, ITCTF #33)
- Frederick Brooks. "The Mythical Man-Month" (1975, ITCTF #40)

Class 15 (Thursday, March 21): Sharing time and shrinking circuits

- *Reading guide here.*
- Fernando Corbato, Marjorie Daggett, and Robert Daley. "An Experimental Time-Sharing System" (1962, ITCTF #23)
- Gordon Moore. "Cramming More Components onto Integrated Circuits" (1965, ITCTF #25)

Friday, March 22 (11:59pm): First-round videos due, if you are submitting one. Full details on video project here. Please follow the instructions for naming your video file, uploading it, and for naming and uploading the paper your video is about.

Week 9

Class 16 (Tuesday, March 26) Programming methodology **with special guest Barbara Liskov**

- *No reading guide for today, except to point out that in the selection, Prof. Liskov is describing an early version of CLU.*
- Barbara Liskov and Stephen Zilles. "Programming with Abstract Data Types" (1974, ITCTF #39)

Class 17 (Thursday, March 28): Dijkstra

- *Reading guide here.*
- Edsger Dijkstra. "Solution of a Problem in Concurrent Program Control" (1965, ITCTF #26)
- Dijkstra. "The Structure of the `THE'-Multiprogramming System" (1968, ITCTF #28)
- Dijkstra. "Go To Statement Considered Harmful" (1968, ITCTF #29)

Week 10

Class 18 (Tuesday, April 2): UNIX

- *Reading guide here.*

- Dennis Ritchie and Kenneth Thompson. "The UNIX Time Sharing System" (1974, ITCTF #37)
- Kenneth Thompson. "Reflections on Trusting Trust" (1984, in the File system)

Thursday, April 4, 11:59pm: Paper proposals due for April 29 (second-round) video submissions, if you plan to submit one. Full details on video project here.

Class 19 (Thursday, April 4): Complexity

- *Reading guide here.*
- Stephen Cook. "The Complexity of Theorem-Proving Procedures" (1971, ITCTF #34)
- Richard Karp. "Reducibility Among Combinatorial Problems" (1972, ITCTF #36)
- Donald Knuth. "Big omicron and big omega and big theta" (1976, ITCTF #43)
- Fun, optional bonus reading: Donald Knuth. "A Terminological Proposal" (1974, in the File system here)

Week 11

Class 20 (Tuesday, April 9): Programs as formalisms **with special guest Rich De Millo**

- *Reading guide here.*
- C. A. R. Hoare. "An Axiomatic Basis for Computer Programming" (1969, ITCTF #31)
- Richard De Millo, Richard Lipton, and Alan Perlis. "Social Processes and Proofs of Theorems and Programs" (1979, ITCTF #44)

Class 21 (Thursday, April 11): Data organization and retrieval

- *Reading guide here.*
- Edgar Codd. "A relational model of large shared data banks" (1970, ITCTF #32)
- Karen SpÃ¤rck Jones. "A Statistical Interpretation of Term Specificity and Its Application in Retrieval" (1972, ITCTF #35)

Week 12

Class 22 (Tuesday, April 16): Algorithms II

- *Reading guide here.*
- Volker Strassen. "Gaussian Elimination is not Optimal" (1969, ITCTF #30)
- Adi Shamir. "How to Share a Secret" (1979, ITCTF #46)

Class 23 (Thursday, April 18): Security and cryptography **with special guest Whit Diffie**

- *Reading guide here.*
- Whitfield Diffie and Martin Hellman. "New Directions in Cryptography" (1976, ITCTF #42)
- Ronald Rivest, Adi Shamir, Leonard Adelman. "A method for obtaining digital signatures and public-key crypto systems" (1978, ITCTF #45)

Week 13

Class 24 (Tuesday, April 23): Doing the right thing

- *Reading guide here.*
- Nancy Leveson and Clark Turner. "An Investigation of the Therac-25 Accidents" (1993, in the File system)
- William James, "The True Harvard" (1903, in the File system)

Week 14

Monday, April 29 (11:59pm): Second-round video submissions due, if you are submitting one. Full details on video project here. Please follow the instructions for naming your video file, uploading it, and for naming and uploading the paper your video is about.