CS 425 / ECE 428
Distributed Systems
Fall 2026

Aishwarya Ganesan & Ram Kesavan

(Thanks to Indranil Gupta)

Lecture 1: Welcome, and Introduction

Aug 25, 2026

web: courses.engr.illinois.edu/cs425/

Course Staff

- Instructors:
  - (*Aishwarya Ganesan*): 3120 Siebel Center
  - (*Ram Kesavan*): 3126 Siebel Center
  - Office Hours Every Tue and Thu right after class until 4 pm – see website
- TAs: (see course website for OH)
  - *Nishant Sheik (Lead TA)*
    - *TAs for On-campus Students: Wei Zhang, Rafid Murshed, Tianyi Zhong, Weidong Wang, TBD*
  - *Xinying Zheng (lead TA for MCS Coursera, and Deputy Lead TA)*
    - *TAs for MCS Online Students: Parham Chavoshian, Soham Chakraborty, TBD*
  - *All students (on-campus, Chicago, & Coursera) can access all TAs*
- More about course logistics later in this lecture…

What This Course is Really About

- Distributed Systems
- How to design algorithms for them
- How to design these systems
- How they work in real life
- How to build real distributed systems

Computer Science in the time of AI

- Should I bother writing or debugging my MP code?
- AI is a capability multiplier:
  - But, 0 x 1000 = 0; if your understanding of concepts, systems, and algorithms is zero, then you'll produce zero!
  - AI is creating a bigger gulf between a highly capable + experienced engineer and the rest!
- Prompts express intent, but true engineering is about trade-offs, performance, maintenance, and architectural taste.
- If anything, knowing the fundamentals is even more important than before
  - Requires deep engagement with the material
  - Best gained by writing, reading, and debugging code
- My recommendation: write and debug your MPs to better learn the fundamentals
  - Use this time to learn. Learning on-the-job will be difficult—there'll be no time

Definition

Define the Term Distributed System

What is an Operating System?

FOLDOC definition

(FOLDOC = Free On-Line Dictionary of Computing)

Operating System - The low-level software which handles peripheral hardware, schedules tasks, allocates storage, and presents a default interface to the user when no application program is running.

Can you name some examples of Distributed Systems?

Can you name some examples of Distributed Systems?

- Client-Server (NFS)
- The Web & the internet
- IoT (Internet of Things) networks
- DNS
- Gnutella, BitTorrent, Blockchain (peer to peer overlay network)
- “Cloud” service, from AWS, Microsoft Azure, Google Cloud, etc.
- Datacenter, e.g., NCSA, a Google/AWS datacenter

What is a Distributed System?

How does Leslie Lamport define Distributed System?

- *(Leslie Lamport is considered by many to be the most prominent Distributed System Researcher)*

- *"A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable."*

FOLDOC definition

A collection of (probably heterogeneous) automata whose distribution is transparent to the user so that the system appears as one local machine. This is in contrast to a network, where the user is aware that there are several machines, and their location, storage replication, load balancing and functionality is not transparent. Distributed systems usually use some kind of client-server organization.

Textbook definitions

- A distributed system is a collection of independent computers that appear to the users of the system as a single computer.

  [Andrew Tanenbaum]

- A distributed system is several computers doing something together. Thus, a distributed system has three primary characteristics: multiple computers, interconnections, and shared state.

  [Michael Schroeder]

Unsatisfactory

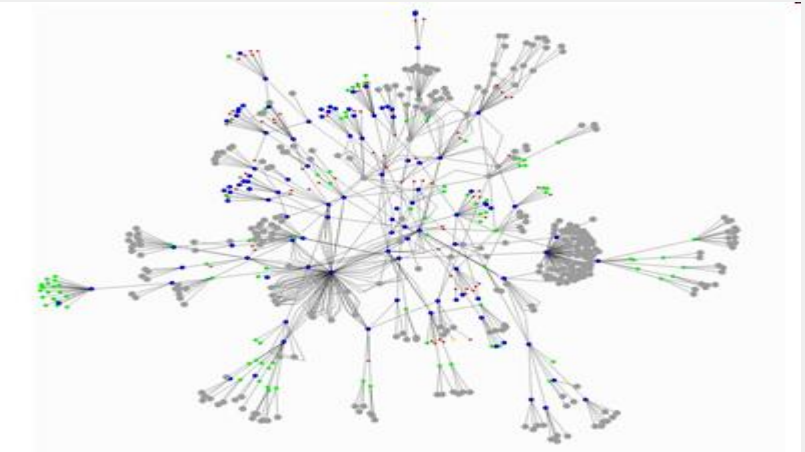
- Why are these definitions short?
- Why do these definitions look inadequate to us?
- Because we are interested in the insides of a distributed system
  - Design and implementation
  - Maintenance
  - Algorithmics ("protocols" or "distributed algorithms")

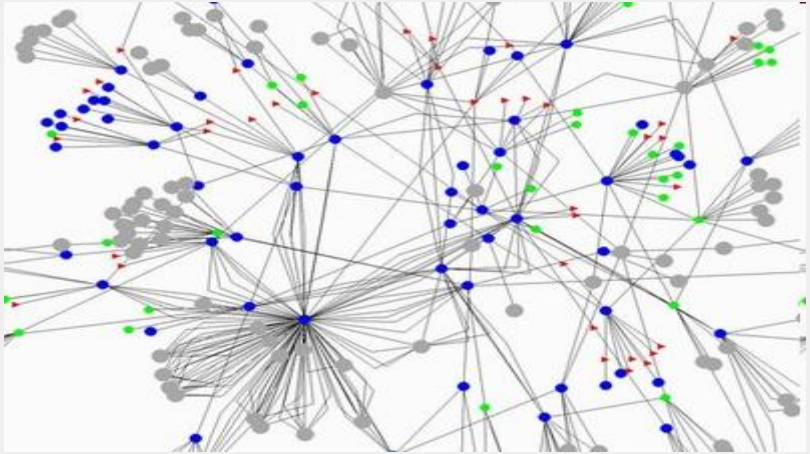
A working definition for us

A distributed system is a collection of entities, each of which is autonomous, programmable, asynchronous and failure-prone, and which communicate through an unreliable communication medium.

- Entity=a process on a device (PC, PDA/mobile device)
- Communication Medium=Wired or wireless network
- Our interest in distributed systems involves
  - design and implementation, maintenance, algorithmics

Gnutella Peer to Peer System





What are the "entities" (nodes)?

What is the communication medium (links)?

Source: GnuMap Project

Datacenter

What are the "entities" (nodes)?

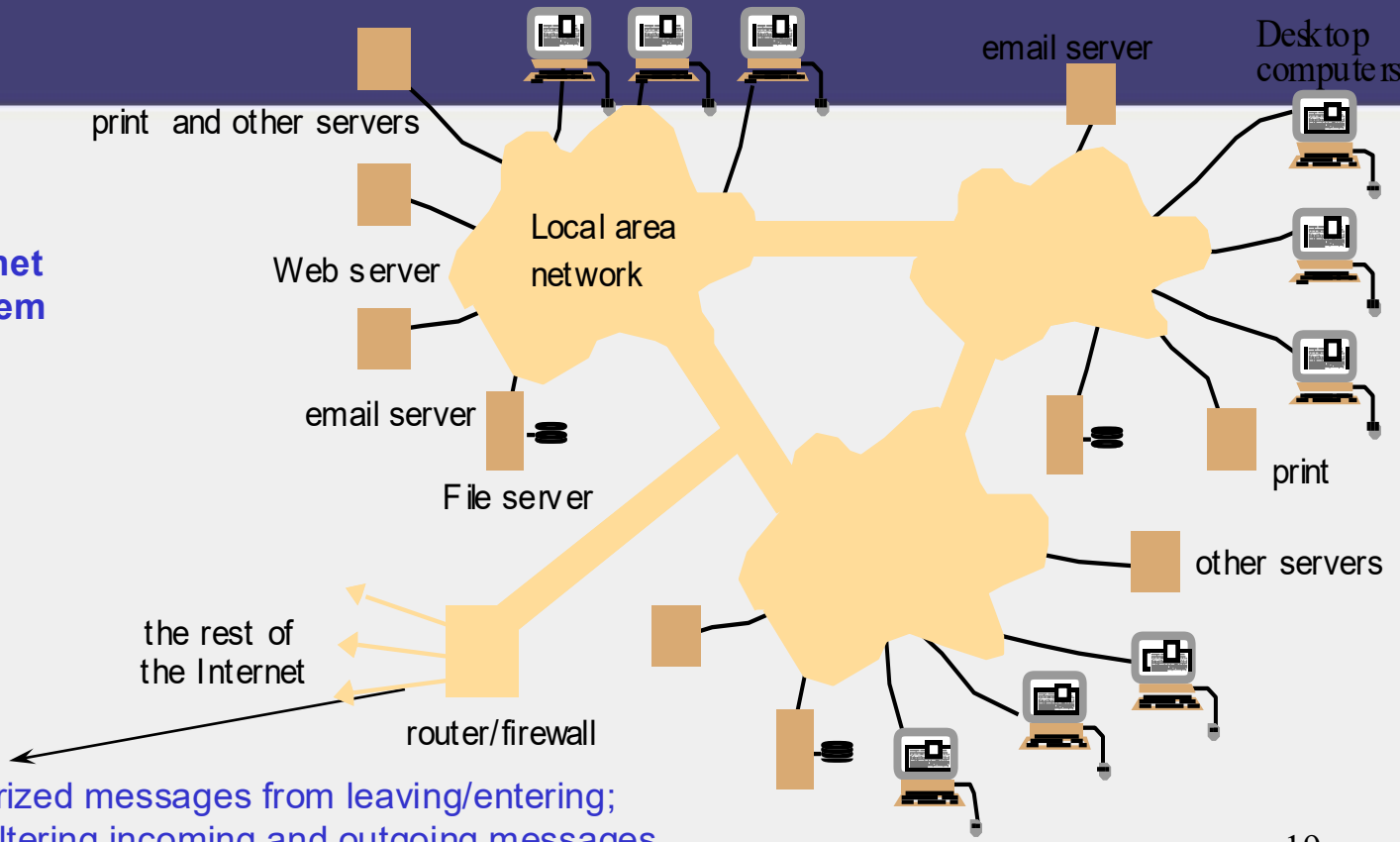
What is the communication medium (links)?

The Internet: A Refresher

- Underlies many distributed systems
- A vast interconnected collection of computer networks of many types
- Intranets – subnetworks operated by companies and organizations
- Intranets contain LANs (local area networks)
- WAN – wide area networks, consists of subnets (intranets, LANs, etc.)
- ISPs – Internet Service Providers. Companies that provide modem links and other types of connections to users
- Intranets (actually the ISPs' core routers) are linked by backbones – network links of large bandwidth, such as satellite connections, fiber optic cables, and other high-bandwidth circuits
- UC2B, Google Fiber (MAN = Metropolitan Area Networks)

An Intranet & a distributed system

Running over this Intranet is a distributed file system



prevents unauthorized messages from leaving/entering; implemented by filtering incoming and outgoing messages via firewall “rules” (configurable)

Networking Stacks

Application	Application layer protocol	Underlying transport protocol
e-mail	smtp [RFC 821]	TCP
remote terminal access	telnet [RFC 854]	TCP
Web	http [RFC 2068]	TCP or UDP
file transfer	ftp [RFC 959]	TCP or UDP
streaming multimedia	proprietary (e.g. RealNetworks)	TCP or UDP
remote file server	NFS	TCP or UDP
internet telephony	proprietary (e.g., Skype)	typically UDP

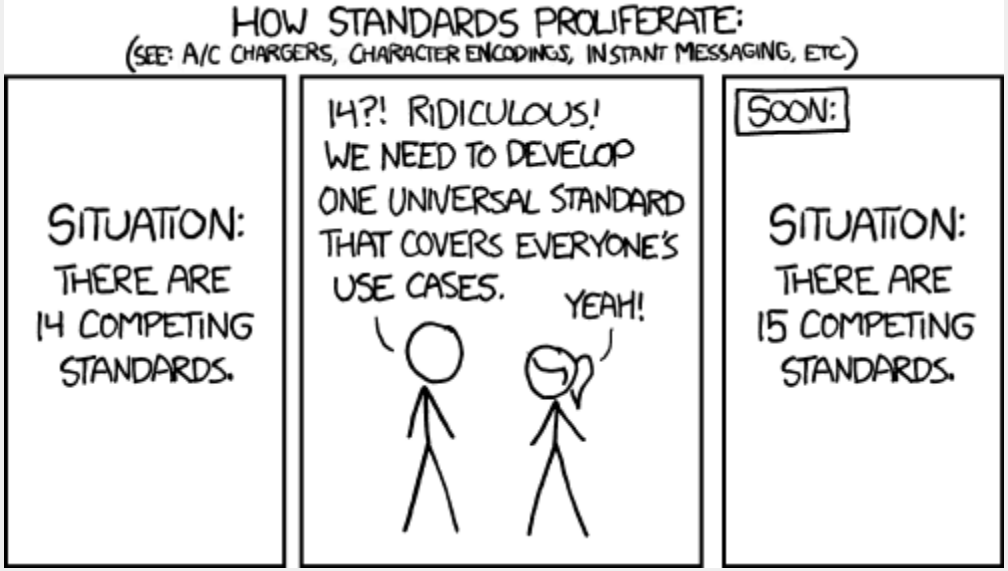
Distributed System Protocols!

Networking Protocols

TCP=Transmission Control Protocol
UDP=User Datagram Protocol

(Implemented via sockets)

The History of internet Standards

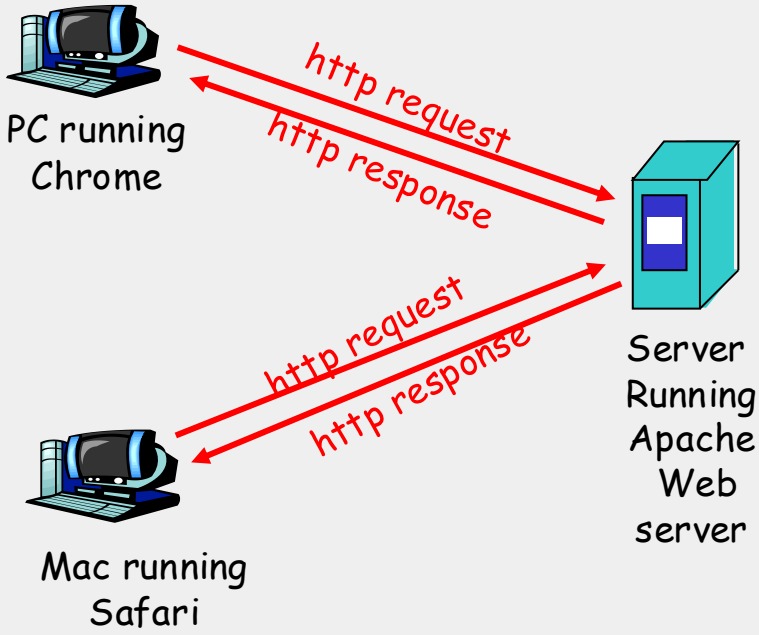


Source: http://xkcd.com/927/

The Heart of the World Wide Web: the HTTP Standard

HTTP: hypertext transfer protocol

- WWW's application layer protocol
- client/server model
  - *client:* browser that requests, receives, and "displays" WWW objects
  - *server:* WWW server, which is hosting the website, sends objects in response to requests
- http1.0: RFC 1945
- http1.1: RFC 2068
  - Leverages same connection to download images, scripts, etc.



The HTTP Protocol: More

http: TCP transport service:

- client initiates a TCP connection (creates socket) to server, port 80
- server accepts the TCP connection from client
- http messages (application-layer protocol messages) exchanged between browser (http client) and WWW server (http server)
- TCP connection closed

http is "stateless"

- server maintains no information about past client requests

Why?

Protocols that maintain session "state" are complex!

- **past history (state) must be maintained and updated.**
- **if server/client crashes, their views of "state" may be inconsistent, and hence must be reconciled.**
- **RESTful protocols are stateless.**

Does our Working Definition work for the http Web?

A distributed system is a collection of entities, each of which is autonomous, programmable, asynchronous and failure-prone, and that communicate through an unreliable communication medium.

- Entity=a process on a device (PC, PDA)
- Communication Medium=Wired or wireless network
- Our interest in distributed systems involves
  - design and implementation, maintenance, study, algorithmics

Important Distributed Systems Issues

- No global clock; no single global notion of the correct time (asynchrony)
- Unpredictable failures of components (failure-prone, unreliable)
  - Fail-stop failures (or sometimes recover after failing)
  - Lack of response may be due to either failure of a network component, network path being down, or a computer crash – challenging
- Highly variable bandwidth: from 16Kbps (slow modems) to Gbps (Internet2) to Tbps (in between DCs of same big company)
- Possibly large and variable latency: few ms to several seconds
- Large numbers of hosts: 2 to several million

Many Interesting Design Problems

-
-
- Real distributed systems
  - Cloud Computing, Peer to peer systems, Hadoop, key-value stores/NoSQL, distributed file systems, graph processing, stream processing, machine learning …
- Classical Problems (Distributed algorithms)
  - Failure detection, Asynchrony, Snapshots, Multicast, Consensus, Mutual Exclusion, Election, …
- Concurrency
  - RPCs, Concurrency Control, Replication Control, Paxos, Raft…
- Security
  - ACLs, Capabilities, …
- Others…
-

Typical Distributed Systems Design Goals

- Heterogeneity – can the system handle a large variety of types of machines and devices?
- Robustness – is the system resilient to host crashes and failures, and to the network dropping messages?
- Availability – are data+services always there for clients?
- Transparency – can the system hide its internal workings from the users? (warning: term means the opposite of what the name implies!)
- Concurrency – can the server handle multiple clients simultaneously?
- Efficiency – is the service fast enough? Does it utilize 100% of all resources?
- Scalability – can it handle 100 million nodes without degrading service? (nodes=clients and/or servers) How about 6 B? More?
- Security – can the system withstand hacker attacks?
- Openness – is the system extensible?

“Important” Issues

- If you’re already complaining that the list of topics we’ve discussed so far has been perplexing…
  - You’re right!
  - It was meant to be (perplexing)

- The Goal for the Rest of the Course: see enough examples and learn enough concepts so these topics and issues will make sense
  - We will revisit many of these slides in the very last lecture!

“Concepts”?

- Which of the following inventions do you think is the most important?
  1. Car
  2. Wheel
  3. Bicycle

“What lies beneath?” Concepts!

How will We Learn?

All this information is contained in handout on course website: "Syllabus/Course Information Sheet/"

- **Web: courses.engr.illinois.edu/cs425/**

- Lectures
  - In-class activities (incl. quizzes)
  - Mediaspace recordings for review

- Textbook, Recommended but not Required
  - Colouris, Dollimore, Kindberg and Blair (5th edition). CDK Textbook
  - If you use a different or older version, be sure to check problem numbers, etc.
  - Textbook is a great source of practice problems for the exams (midterm and final)

How will We Learn? (2)

This information is in "Syllabus" handout on course website (read carefully!)

- **Web: courses.engr.illinois.edu/cs425/**
- Homeworks (HWs)
- Machine Programming Assignments (MPs)
- Exams/quizzes

How will We Learn? (2)

This information is in "Syllabus" handout on course website (read carefully!)

- **Web: courses.engr.illinois.edu/cs425/**

- Homeworks (HWs)
  - Four in total, two before midterm (early-Oct) and two after
  - About 2-3 weeks per homework
  - Solutions need to be typed, figures can be hand-drawn
  - Best 3/4 policy

- Machine Programming Assignments(MPs)

- Exams/quizzes

How will We Learn? (2)

This information is in "Syllabus" handout on course website (read carefully!)

- **Web: courses.engr.illinois.edu/cs425/**
- Homeworks (HWs)
- Machine Programming Assignments(MPs)
  - **Only for 4credit on-campus/Chicago** (others welcome to try; no VMs/grading/extra credit)
  - You will be building a distributed (surprise!) processing system in stages
  - Four in total, two before midterm and two after
  - Demo dates set on website: please be in town on those days (no excuses for interviews, travel.)
  - No LLM/coding agents for MP1 and MP2
    - For MP3/MP4, we plan to allow but strict guidelines on how (e.g., you will submit traces)
    - In-person demos will include questions about your code
- Exams/quizzes

How will We Learn? (2)

This information is in "Syllabus" handout on course website (read carefully!)

- **Web: courses.engr.illinois.edu/cs425/**
- Homeworks (HWs)

- Machine Programming Assignments(MPs)

- Exams/quizzes
  - Midterm in class (**Oct 8th**)
  - Final in person (date decided by campus)
  - In-class quizzes/activities
    - Score is ignored but your participation will be recorded
    - 5 points in your final grade

What will the schedule look like?

- **Web: courses.engr.illinois.edu/cs425/**
- Check schedule on web

- Tentative:
  - Lectures

- Final:
  - HW due dates
  - MP demos (in-person)
  - Midterm in class (**Oct 8th**)

- TBA
  - Final (campus sets the date)

What assistance is available to you?

- Lectures
  - Lecture slides will be placed online at course website
    - "Tentative" version before lecture
    - "Final" version after lecture
    - Ask questions during and after lecture
- Asking Questions outside lectures:

  In order of preference:

  1. **Piazza**: Your first stop for all questions (see if your question has already been asked)
     - We will try to ensure that every question is answered within 24 hours of posting (during weekdays).
     - Students can answer other students' Qs, but to do so respectfully
     - Do not post solutions or code on Piazza – that'll immediately earn you a zero on the HW/MP.
  2. **Office Hours** (posted on course website): Every weekday has multiple office hours scheduled, including evenings.
  3. **Email staff** mailing list (don't email individual instructor/TA except for private issues)

What tools will we use?

- All links are posted on website. Signup by today!
- Gradescope: You one-stop shop to submit all HWs and all MP reports
- Piazza: Your first stop for all questions and important announcements
- Canvas: We will post all your final HW, MP, and exam grades here
- (4cr students)
  - Google Apps: For MP group signups, MP(1-4) demo signups, MP(1-4) repo link submissions
  - GitHub: For MP code repositories
  - MP group signup link posted
  - Others will be posted as we progress…

What tools will we use? (2)

- New this year…

- For in-class activities: Acadly
  - https://app.acadly.com/auth/signup/student
  - Enter Illinois email address
  - Use code: **ZHXVAN**
  - Signup by next class

Course Prerequisites

- Credit or concurrent enrollment in one of CS 240/241/340/341,
  or ECE 391,
  or equivalent OS/networking course

(latter need instructor permission)

Please fill survey by 9/1: https://forms.illinois.edu/chooseAuth/165568737

Individual vs. Group Work

- Homeworks
  - **Individual work only**. All work must be your own.
- MPs
  - In groups of 2
  - Within group: Can discuss everything.
  - Outside group (e.g., on Piazza): can only discuss concepts and question (but not solution)
- For both HWs and MPs
  - You can discuss with others (and course staff) lecture concepts and the HW/MP question/spec itself, but **you cannot discuss solutions or even ideas.**
  - You cannot copy solutions/code or look at someone else's solutions: we check with Moss, etc.
  - No use of AI permitted (you will get wrong solutions, and you won't know it)
- First violation: zero on HW/MP. Second violation: F in course. (Both have happened in the past!)

Course Staff

- Instructors:
  - (*Aishwarya Ganesan*): 3120 Siebel Center
  - (*Ram Kesavan*): 3126 Siebel Center
  - Office Hours Every Tue and Thu right after class until 4 pm – see website
- TAs: (see course website for OH)
  - *Nishant Sheik (Lead TA)*
    - *TAs for On-campus Students: Wei Zhang, Rafid Murshed, Tianyi Zhong, TBD*
  - *Xinying Zheng (lead TA for MCS Coursera, and Deputy Lead TA)*
    - *TAs for MCS Online Students: Parham Chavoshian, Soham Chakraborty, TBD*
  - *All students (on-campus & MCS Coursera) can access all TAs*

Speaking of HWs and MPs

- MPs (4 cr only):
  - MP demos in-person (not online), and the demo dates are posted to website. So please plan ahead!
  - MP1 has been released
  - Must form groups and let us know your group by Monday 8/31 @ 5 pm.
    - If you don't meet this deadline, you won't get VMs to do your MP!
    - Submit form and not e-mail (see course website for instructions)
  - You can start on MP1 right away (don't need lectures for it)
- HWs:
  - HW1 releases this Thursday
- Don't worry – you have time, but start early! Start now
  - Due in about 2-3 weeks
  - MP1 due 9/13 (demos on 9/14)
  - HW1 will be due 9/20 after we release it this Thursday

3 cr vs 4 cr vs MCS Coursera

	3 cr (On-campus	4 cr (On-campus
HWs 1-4 On-campus	Yes	Yes
MP On-campus	No	Yes
On-campus Midterm Exam + On-campus Final Exam	Yes	Yes

+ Pop-up in-class Quizzes for on-campus

Wrap-Up

- (Reading for today's lecture: Relevant parts of Chapter 1)
- All students:
  - Go to course website **https://courses.engr.illinois.edu/cs425/**
  - Todos:
    - (All) Sign up for Piazza, Gradescope, Acadly by today!
    - (4 cr) Form MP group by this week and respond to MP group survey. You can use Piazza "Search for Teammates", or just hang around after class today.
    - (All) Fill out Student Survey sheet by this week
  - Not yet registered?
    - We cannot allow "double booking" with CS425 and another class at same time
    - No waitlist, continue to do all the HWs, (MPs if 4 cr).
    - Usually we get everyone in by end of September (no guarantees though!).