Course Introduction

COMS4113: Fundamentals of Distributed Systems

Instructor: Hubertus Franke

Columbia University

https://frankeh.github.io/cu-ds1-2026-fall

About This Course

- This course, Distributed Systems 1, was created and is normally taught by Prof. Roxana Geambasu, who designed its curriculum, lectures, and homework series.
- She remains the general instructor of the class. Prof. Hubertus Franke is stepping in to teach it for this one semester.
- The materials, structure, and policies in these slides are Prof. Geambasu's; full credit for the course design goes to her.
- Field is moving always, fundamentals don't change that much.
- Material was updated were I seemed it necessary.

Interested in...

- scalable web services
- cloud services
- big data processing and management
- and the large-scale infrastructure systems making these possible

- If so, you're in the right room.

Distributed Systems Classes at CUCS

- Fundamentals of Distributed Systems (DS1, Fall)
  - Teaches basic concepts, principles of large-scale DS design.
  - Discusses application of those concepts/principles in real-world systems (e.g., Google/Facebook/Amazon).
  - Homework series teaches how to build basic DS.

- Advanced Distributed Systems (DS2, at times in Spring)
  - Research seminar that discusses topics in depth by reading the most influential papers that originated them.
  - Includes both classical papers on well-understood topics and new, state-of-the-art papers.

Related CU Classes

- Multiple cloud computing/web programming/big data processing classes are offered @CU.
  - Those classes teach you how to use various popular DSes.
  - This class teaches you the how those and other systems are built, so you can build and use them better in the future.
  - The most similar class is COMS 4121 Computer Systems for Data Science course (led by Asaf Cidon), which is higher level and has less intense homeworks than DS-1.

- DS-1 is designed in the spirit of the OS class: intense but very useful if you want to understand how real, large-scale systems work.

This Class

- Foundational concepts of large-scale DSes: challenges, algorithms, techniques, abstractions.
- The inner-workings of several DSes serving as infrastructure for large companies.
  - E.g.: Google's protobuf/Spanner/MapReduce, Yahoo's Hadoop, Amazon's Dynamo, etc.
- How to build a DS yourself: through a series of coding assignments, you will build your own DS.

Topics

1. **Distributed systems primer**: challenges and goals of distributed systems.
2. **Communication models**: remote procedure calls (RPC), RPC libraries, failure models, semantics.
3. **Distributed computation:** MapReduce design, TensorFlow design, approximate computing engines.
4. **Time and coordination**: challenges, physical and logical clocks, distributed mutual exclusion.
5. **Sharding, replication, and the agreement problem**: commitment and consensus, use cases for each.
6. **Local transactions (background)**: ACID semantics, concurrency control, recovery mechanisms.
7. **Commitment protocols:** 2-phase commit, 3-phase commit, safety/liveness tradeoffs with the two.
8. **Consensus protocols**: Paxos overview, key ideas, basic algorithm, examples, liveness failure mode.
9. **Case study: Google's Spanner:** TrueTime, Spanner and its fault-tolerant, linearizable, distributed transactions.
10. **Other consistency and isolation semantics**: sequential, causal, and eventual consistency; a few isolation semantics; mechanisms to achieve each; tradeoffs between them.
11. **Other infrastructure systems:** cluster orchestration and scheduling with Kubernetes; monitoring and tracing systems.
12. **Testing and model checking**: testing approaches, formal specification and model checking; TLA+.
13. **Distributed systems security primer**: authentication protocols, Kerberos, byzantine fault tolerance.

Example Messages from Fall 2024 DS-1 Graduates

- “I just want to say that what I learned in the course was not only fun but also very helpful especially in what I work in currently (blockchain stuff).”
- “DS-1 was one of my most memorable courses at Columbia and I learned a lot from it which I have been able to apply a bit at my current job. I am sure the new students will have the same experience as well!”
- “I’m starting a job in a few weeks though where I’ll be working in Go on a lower level database. I definitely feel like DS-1 specifically prepared me for a lot of topics I will encounter in my job, and can confidently say it was the hardest and most rewarding course I took during undergrad.”
- “Distributed was, hands down, my favorite CS class in college, and you were an incredible professor. I hope you are doing well!”
- “I really enjoyed taking Distributed Systems and had a lot of fun with the homeworks. I learned so much. I wish you all the best!”
- “It was one of the most engaging classes I took, and I’m actually applying many of those concepts in my current internship.”
- “I really enjoyed taking the course with you and gained so many practical skills that I'm already putting to use at my new job :)”
- “Just wanted to say that your class was very useful to understand how systems work in my current job. Thank you for that!”
- “Distributed Systems was one of my favorite classes, and I’ve been applying a lot of what I learned to my daily work as a software engineer.”

What Others Say

“Your course has profoundly impacted my academic journey since my graduate school days. The depth of content, the structured organization, and the timely integration of the latest advancements from the community have consistently impressed me. It is one of the best learning materials I could find for distributed system course — comprehensively cover different aspects while keeping a strong focus on key concepts”

— August 2023 message from a junior DS faculty at another university

Expectations

- BUT:

- To gain from this class, you must put in the effort:
- Come to class and quiz me -- all the time!
- Do your homework as best as you possibly can; quiz the CAs, consider different designs, discuss them with your classmates.
- Do the readings we assign and discuss with classmates.

- This will prepare you for a marketplace that's becoming quite challenging for entry-level candidates. You must exit from your education program with deeper expertise than ever before.

- As professors, we are adapting our courses in light of that, by doubling down on the depth of our courses so you can differentiate and succeed in the marketplace.

- But you've gotta do the work!!!

Important Addresses

- Website: https://frankeh.github.io/cu-ds1-2026-fall

- Discussions: ED (link on Courseworks)
  - CAs will closely monitor ED, homework questions should go there. You should expect CA response within several hours during workdays.
  - For questions on lecture material, please come to class or my OHs and pose them there! I will not be using ED directly.

Teaching Staff

- **Instructor:** Hubertus Franke, Adjunct Professor of CS
- **CAs:**
  - Hasan Zengin (lead)
  - Brian Paick
  - Jonathan Ang
  - Shen Li
- All CAs took DS-1 in some capacity.

About ME

- Distinguished Research Scientist
  @IBM T.J. Watson Research Center in Yorktown Heights, NY (since 1993)
- Ph.D. EE Vanderbilt University 1992
- Diplom/Master CS Karlsruhe Institute of Technology, Germany, 1987
- Manager and Senior Manager of OS and Cloud 2001-2015
- IBM Master Inventor
- ACM Fellow, AAIA Fellow
- **General interests:**
  - Cloud Infrastructures: Containers, Cloud, Security, AI-native platforms
  - Operating Systems: Linux, AIX, object-oriented OS (K42), Scheduling, memory management, ..
  - Computer Architecture: Multicore processors and Systems on a Chip
  - High Performance Computing: MPI (Message Passing Interface)
  - Software Engineering , Compilers, Robotics
- ~160 publications, ~190 patents

Prerequisites

- You must have solid programming experience (C, C++, Java), preferably system-level programming experience and concurrency.

- Columbia courses (or equivalents):
- COMS W3137 Data Structures and Algorithms
- COMS W3157 Advanced Programming
- COMS W3827 Fundamentals of Computer Systems
- COMS 4118 Operating Systems
  - Particularly critical is experience with concurrent programming

- If you lack these prerequisites, do not take the class, because heavy coding accounts for vast portion of grade (and concurrency is a major challenge you will inevitably encounter).
  - Use HW1 to determine if you have sufficient experience.

Grading

- 70%: five coding homeworks projects, consisting of a total of eight independently graded components.
  - The last homework has only one component, which is mandatory and will account for 10% of the grade.
  - Of the first 7 components, the best 6 grades will be counted toward your final grade, each accounting for 10%.
- 30%: final quiz/exam during the last class / exam day.
- extra credit for extraordinary contributions on ED and/or in class (think: someone who has significantly improved the class).

Grading (cont.)

- In previous years, we've applied the thresholds shown on the right.
- While we reserve the right to alter these thresholds this year to account for major unpredictable disruptions, our expectation is to re-apply them this year.
- In previous years, this gave the following distribution: A+: few; A: more; A-: peak; B+: comparable to A; B, B-, C: tail; D, F: very few.

final-score	letter grade
>=100	A+
[90, 100)	A
[80, 90)	A-
[70, 80)	B+
[60, 70)	B
[50, 60)	B-
[40, 50)	C
<40	F

Policies

- Deadline policy: No homework extensions will be granted, but 72-hour grace period is granted to use as you wish over the entire semester.
  - Once you consume the 72h, submitting a new homework one second late will result in a zero score for that homework.
  - 48-hr max on a single assignment

- Collaboration policy: You can discuss, preferably openly on ED, but DO NOT look at or copy “someone else’s code”!
  - “Someone else’s code” could be: a present colleague, a past student (at CU or somewhere else), the Internet, an AI.
  - We subscribe to the CS department’s Policies and Procedures regarding Academic Honesty and Columbia’s AI policy. Prof. Jae Woo Lee has an accessible description of these policies. Please read all three!

- We will be checking all code for cheating and doing oral examinations. We will be drastically punishing anyone who cheats!

Why the Policies

- Deadline policy:
  - Because some homeworks depend on each other, we must release solutions in between major homeworks.
  - Allowing extensions of any kind derails the homework series' timeline for everyone, so it is highly unfair. Hence a max 48-hr per homework.
  - Still, life happens and issues arise for each of us, hence the total 72-h grace, to use it when you need it, with no questions asked. But use carefully, as you won't get more. (We time solution releases based on this grace period.)

- Collaboration policy:
  - While in your jobs you will undoubtedly be reusing available code and AI tools, you must first experience the intricacies of DS development before you can expect to be able to reason about code you import from elsewhere.
  - You will not write "a lot of code," and this is not "entry-level position" code, but rather complex pieces of logic that you must experience yourself to learn.

Homework 0

- Not graded but mandatory and important:
  - Reviews policies regarding academic honesty and zero tolerance to code cheating. Read them carefully before you sign!
  - Sets up github, which you'll use for all homeworks.
  - Complete this homework even if you are not yet enrolled (but you hope to be)!

- Deadline:
  - If you missed it, see the CA right after class. !!!!

Homework Series (HW1-5)

- Builds in stages a simple but fairly realistic, fault-tolerant, consistent, distributed key-value store, and then model checks a key protocol for it.

- Five assignments:
  First four are based on MIT's series,
  Fifth is created at Columbia to complement an important missing component.

- Assignments are in Go and many build upon each other. We provide solutions in between dependent homeworks.

- We will also give in-class overviews of homeworks and provide solutions on courseworks after 48hr grace period

In Class

- Participation:
  - Not required and attendance is not taken, but please be active in class.
  - Classes are recorded and posted to Courseworks Video Library or Zoom.
  - If you miss a class, please view recording before you join next class!

- Structure:
  - I will cover the week's topic lecture-style.
  - We'll also have interactive activities. Please be active on these!
  - Questions/comments are welcome and encouraged at any time!
  - Bio break at 8:15pm-8:20pm.

- Lectures cover HW topics but not in perfect synchrony. Read ahead and do not wait for all material to be taught before starting HW, esp. for HW1.

Respect & Inclusion

Zero tolerance to
discrimination, harassment, sexual assault, and (cyber)bullying.

If you witness such behaviors, please report them:
https://www.cs.columbia.edu/reporting-discrimination-sexual-harassment-and-cyberbullying/