

CS 117:
Internet-scale Distributed Systems
Lessons from the World Wide Web

Noah Mendelsohn
Tufts University
Email: noah@cs.tufts.edu
Web: <http://www.cs.tufts.edu/~noah>



Discussion: if someone asked
you to design a big distributed
*system...what would you
worry about?*

What you should get from today's session

- You know what course is about and why you would want to be here
- You know who I am
- You know mechanics of the course and what will be expected of you
- If there's time...start on some technical work

Our Team

- **Professor: Noah Mendelsohn**
- **Graduate TA: Zhaoqi (Roy) Zhang**
- **Undergrad TA: Elliot Bonner**

This is a *discussion* class!

Quotes from Spring 2020 student feedback on COMP 117:

- **"[The most valuable aspect of the class was...] the discussions.** Everyone was engaged and insightful. My favorite part of any of these discussions would be how much my opinion would change compared to what I had thought coming into the discussion. It was a great, collaborative, and open environment and it really drove home how the class really was focused on these higher concepts and wasn't just technical stuff."
- "There were a lot of open discussions, and **I was always blown away by the [great] ideas that the other students in the class had that didn't occur to me. Every time, I was sort of like, "oh duh."**"

Your ideas matter no matter how much or little experience you have, please speak up!

Introduction

The course will explore several threads in parallel:

- 1. We will look top-down at various principles of distributed system architecture**
- 2. We learn to build Internet clients and servers that communicate**
- 3. We will contrast several styles of distributed systems**
- 4. We will explore the history of computer networking, and explain why the Web is designed the way it is**
- 5. We will look at a variety of other related topics**

What this course is not

- **A course on designing Web applications (CS 20/120)**
- **An comprehensive OS course (CS 111)**
- **An comprehensive networking course (CS 112)**
- **A comprehensive database course (CS 115)**
- **A comprehensive security course (CS 116)**
- **A how-to cloud computing course (CS 118)**
- **A programming languages course (several)**
- **A traditional distributed systems course**

What this course is not

- A course on designing Web applications (CS 20/120)
- An comprehensive OS course (CS 111)
- An comprehensive networking course (CS 112)
- A comprehensive database course (CS 115)
- A comprehensive security course (CS 116)
- A how-to cloud computing course (CS 118)
- A programming languages course (several)
- A traditional distributed systems course
- **However...we will explore issues relating to all of those!**

About me

▪ **50+ years building & teaching about computing systems**

- Mainly at IBM and Lotus
- Lots of time working w/ university research groups (Stanford, MIT, UCLA)
- Worked mainly at intersection of research and advanced technology development
- Have worked on many important and interesting systems
- Taught programming at Stanford
- *Starting in mid-1990s : known for contributions to Java, XML, Web, etc.*
- *Thru spring 2013: chair of the World Wide Web Consortium TAG*

▪ **Current**

- Professor of the Practice in Computer Science at Tufts University – since 2012
- IBM Distinguished Engineer Emeritus

▪ **Contact and additional info:**

- noah@cs.tufts.edu
- <http://www.eecs.tufts.edu/~noah/> (more available from there)
- <http://www.arcanedomain.com> (personal Web site – see esp. list of recommended CS readings)

I prefer that you call me "Noah" (Prof. Mendelsohn is also OK)

CS 117:
Internet-scale Distributed Systems –
Lessons from the World Wide Web

CS 117:
Internet-scale Distributed Systems –
Lessons from the World Wide Web

CS 117:
Internet-scale Distributed Systems –
Lessons from the World Wide Web

CS 117:
Internet-scale Distributed Systems
Lessons from the World Wide Web

Tim Berners-Lee



Inventor of the World Wide Web
Director of the World Wide Web Consortium (W3C)

The World Wide Web Consortium (W3C)

What the W3C does*

- **Create and promote the standards that make the Web work**
- **Work to ensure the openness of the Web**
- **Etc.**

* Another organization called the IETF also plays a key role in creating core Internet standards

Why Document the Principles of Web Architecture?

*These statements of architectural principle explain the thinking behind the specifications. [...]
They are aimed at the technical community, to explain reasons, provide a framework to provide consistency for future developments, and avoid repetition of discussions once resolved.
[Tim Berners-Lee October 1998]*

From: <http://www.w3.org/DesignIssues/Preface.html>

The World Wide Web
Consortium (W3C)
Technical Architecture
Group (TAG)

The W3C Technical Architecture Group (TAG) - 2012

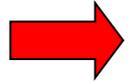


The senior technical body responsible for the Web

What the TAG does

- **Document the architecture of the Web**
- **Resolve issues raised by members of the Web community or W3C workgroups**
- **Coordinate cross-technology architecture developments inside and outside W3C**

Useful publications of the TAG



- Architecture of the World Wide Web (Vol. 1 – there is no Vol. 2!)
<http://www.w3.org/TR/webarch/>
- TAG “Findings” on specific architectural topics:
<http://www.w3.org/2001/tag/findings>
- *We will read some of these during the course – you are encouraged to read more on your own*

References from Tim Berners-Lee

- Tim BL's home page: <http://www.w3.org/People/Berners-Lee/>
- Tim BL's design notes on the Web: <http://www.w3.org/DesignIssues/>
- Web Architecture from 50,000 feet (Tim BL):
<http://www.w3.org/DesignIssues/Axioms.html#Universality>
- Axioms of Web Architecture (Tim BL):
<http://www.w3.org/DesignIssues/Axioms.html>

We will explore many of these references in detail later in the course. For now, you might want to take a look if you're curious.

Course Plan

Course Web Site

<http://www.cs.tufts.edu/comp/117/>

[Course home](#)

[Piazza](#)

[Syllabus/Lectures](#)

[Assignments](#)

[Info](#)

[**Principles**](#)

Assignments & Tests

- **Multiple assignments will sometimes run in parallel – manage your time!**
- **Reading Q&A:**
 - Most reading assignments will come with questions.
 - Two due dates: 1) in time for class discussion, mostly ungraded 2) same questions after discussion, graded
 - Learning to do technical analyses and communicate your insights is important!
 - *For most assignments, you will “provide” answers by filling out and submitting an HTML form that will be supplied to you.*
- **There will be several distributed programming assignments.**
 - You *must* ssh into virtual servers at Halligan to test network programs
 - We will use team programming for larger projects
- **Tests**
 - Two in class tests (one near midterm, one near end of term)
- **Final project**
 - There will be a choice of projects, and you may propose your own. The typical project will be in the form of a paper analyzing some important topic relating to the course. Final projects may involve code, but not necessarily. It will depend on your choice of topic.
 - Assigned later in term, due during finals week

Grade scales and curving

■ **Course grading:**

- *Just do the work that's asked, you'll probably get about a B*
- *Do it with extra style and care, or show some extra creativity, that's an A*
- *Do less than a good job, that's B- or below*

■ **Assignment grading: modeled on CS-40 but using numbers**

- *Scale: 0-15*
- *12 is roughly an A, 9 is roughly B, 6 is roughly C*
- *> 12 is excellent, A+, over-the-top (and rare) – this scale is designed to allow you to get credit for excellence*
- *...DO NOT PANIC IF YOU GET A 9 OUT OF 15, IT REALLY, REALLY IS ROUGHLY A B!*

■ **Curving**

- *I will not curve grades with the intention of ensuring a fixed distribution of A's, B's, C's – my hope is that you will all earn A's, but that does mean you could all get Bs or Cs.*
- *I may curve results up or down if I feel that a test or assignment did not accurately measure what I intended. I may also just decide not to count such work if it would lower your grade.*

■ **Coding**

- *Coding “standards” will be suggested – not in all cases the same as for CS 40*
- *Mainly: your code must be well structured, modular, and easy to understand*
- *The source of your code must be visually appealing and easy to navigate*
- *Use as many well-formatted comments as necessary to make your code comprehensible: a comment is good if the code is more understandable with it than without*

Your final course grade

- **Grade weighting – EXACT DETAILS TBD, BUT THIS IS TYPICAL:**
 - Probable: 20% in class participation and reading q&a; 30% programming projects, 30% tests, 20% final project. I'll announce final weightings later.
- ***If you do all your work on time, you are guaranteed a grade no worse than this.***
- ***You can do better than this by getting extra credit for work that goes “above and beyond”***
 - Some assignments will offer extra credit topics/questions
 - If you have ideas for extra credit work or alternate assignments, ask me
 - You do not need to do extra credit work to get an A, but you do need to do a thorough and creative job on the assignments. You should try for extra credit if you're excited about the material, or have extra insights that you'd like to share.
- ***You may lose credit if your work is late or you miss class too often***
- ***All extra credit, and all penalties for missing or late work are at my discretion***
 - *I try to be fair...I understand that people get sick or have other work, and I have no interest in tripping you up over trivialities. That said, late work puts a strain on me and the TAs, and sometimes we will be discussing the “answers” in class. At the end of the term, extra or lost credit may or may not affect your grade (typically depending on how close you are to a different grade, and how much extra/lost you have).*
- ***I may raise (but not lower!) your final grade***
 - ...if I feel that your overall knowledge of the material has been demonstrated – e.g. in discussions with me
- **There will be no incompletes given. Exceptions will be made only in extreme circumstances after consultation with the dean.**

In short... the computed grade is a lower bound if all your work is on time!

Piazza

- **Piazza is an online discussion and course management system**
- **We will use it for:**
 - Answering your questions
 - Announcements from the course staff
- **Discussion and asking questions**
 - We prefer that you ask questions in public posts on Piazza
 - You can help each other with answers
 - Everyone can see what's causing confusion
 - Obviously: don't post answers to hw questions or fragments of solution code. You can discuss such things in "instructor-only" Piazza posts
 - If you have a truly private matter email ta117ids@cs.tufts.edu or noah@cs.tufts.edu
- **Sign up for Piazza now!**

Reading Materials

- **There is no textbook for this course**

- *There is one book you must buy or borrow...see below*
- You will be assigned selected readings from several books, *(almost) all of which are available on Safari books online*. Information on accessing these is available from the course Web site.
- You *may* want to consider getting one or more of:
 - Kerrisk, M. *The Linux Programming Interface* – a terrific book covering all the Linux APIs + networking concepts
 - Whatever book suits you for learning advanced C++ topics. I use the book by the inventor of C++: Stroustrup, B. *The C++ Programming Language* but you may prefer another.
 - Nemeth, et. al., *Unix and Linux System Administration Handbook (4th Ed.)* – we're not learning system admin, but this has very comprehensive information about Unix/Linux command line programming
 - *Kerrisk, Nemeth and Stroustrup are all on Safari*, and many alternative books on these topics are too

- 
- **Berners-Lee, Tim, Weaving the Web – *YOU WILL NEED IT FOR THE ASSIGNMENT RELEASED TODAY (BUT DUE DATES ON THAT ASSIGNMENT ARE VERY FLEXIBLE)***

FALL 2023: I HAVE COPIES TO LOAN...INCLUDING SOME THAT TIM HIMSELF DONATED. YOU CAN ALSO FIND VERY INEXPENSIVE COPIES ONLINE (OR BORROW FROM SOMEONE WHO TOOK 117 BEFORE.)

Reading Materials

- **There is no textbook for this course**

- *There is one book you must buy or borrow...see below*

- You will be assigned selected readings from several books, *(almost) all of which are available on Safari books online*. Information on accessing these is available from the course Web site.

- You *may* want to consider getting one or more of:

- Kerrisk, M. *The Linux Programming Interface* – a terrific book covering all the Linux APIs + networking concepts
- Whatever book suits you for learning advanced C++ topics. I use the book by the inventor of C++: Stroustrup, B. *The C++ Programming Language* but you may prefer another.
- Nemeth, et. al., *Unix and Linux System Administration Handbook (4th Ed.)* – we're not learning system admin, but this has very comprehensive information about Unix/Linux command line programming
- *Kerrisk, Nemeth and Stroustrup are all on Safari*, and many alternative books on these topics are too



- **Berners-Lee, Tim, Weaving the Web**

- **I will assign selected other readings. All necessary papers will be available online.**

- **When I use slide sets like this, they will be posted online in .ppt/.pptx and .pdf formats, typically just ahead of our class (please bug me if I forget!)**

- **Where published material or slides does not provide sufficient coverage, I will prepare course notes**

Administrivia

- Noah's office hours: Fall 2022 Tuesdays 3:00 – 3:45 PM Cummings 325 (or via Zoom if we are remote) **MY OFFICE MAY BE MOVING SOON**
- Best way to reach Noah quickly: noah@cs.tufts.edu.
- For near emergencies: phone [617-506-3994](tel:617-506-3994) (Google voice)
- TAs:
 - Graduate: Zhaqi (Roy) Zhang (zhaoyi.zhang@tufts.edu)
 - Undergraduate: Elliot Bonner (Elliot.Bonner@tufts.edu)
 - To reach all TAs + Noah: ta117ids@cs.tufts.edu (usually you should use this)
- Remember: most questions should be asked on Piazza

Homework
Assignments
Now Posted

Three homework assignments being given today:

1. Due **Thurs Sept. 7th**: E-mail to instructor
(<http://www.cs.tufts.edu/comp/117/assts/emailassignment>)
2. Due **Thurs Sept. 7th** : Sign up for Piazza
(<https://piazza.com/tufts/fall2023/comp117/>)
3. Due **Friday October 6th or Tues Nov. 14th (*)**: Read chapters from Weaving the Web + 2001 Time Magazine article and answer questions:
(<http://www.cs.tufts.edu/comp/117/assts/weavingtheweb>) Due date depends on whether you get book week of Sept 5th or later.
4. **ALSO: please read the course home and info pages**

There will be overlapping assignments as the term proceeds, but the total workload should be manageable at all time. *Recommendation: start early on assignments as they become available.*

* The due date for the Weaving the Web assignment is traditionally *very* flexible. I prefer you to get this done early before the bigger assignments start, but will likely accept without penalty at least into early December.

Questions?

Credits

- The picture of Falling Water shown on slide #2 is from Happy Via's" photostream on Flickr. (<http://www.flickr.com/photos/via/57100237/> License: <http://creativecommons.org/licenses/by-nc-nd/2.0/>)