Welcome to this course! My name is Jasleen Kaur and I'm the instructor for this course. I conduct research in the design and analysis of networks and distributed systems. I have taught this course many, many times, but this is the first time that the class has grown so big that it needs this room. I have made some structural changes to the course, and am looking forward to a semester-worth of exchanging ideas with each of you!

I LOVE asking questions, and I LOVE getting answers to those questions, so please do make the lectures interesting both for me and for you.

We'll spend today's class talking about the course outline, the course requirements, and your first homework.

But before we begin, I'd like a quick show of hands: How many of you have NOT had any!

There are 4 “handouts” for today:
- Course outline/syllabus
- Lecture 0 slides
- HW 1 description
- Programming guidelines

Please download a copy of each from the course web page at:
http://www.cs.unc.edu/~jasleen/Courses/COMP431/

Attendance (only on first 2 lectures)
Why The Internet?
Why spend a whole course studying it?!

- Arguably one of the most important developments of the last 3 decades
  » Information Sharing
  » Means of Communication
  » Transactions
  » Services
  » Entertainment
  » Creation

Qn: Why? What do we use the Internet for?

The Internet clearly influences most areas of our personal, social, and economic lives:

Let’s think about some of these. Information Sharing! Here’s a timeline photo from Facebook that puts it so well!

Means of communication --- why even voice phone communication today run widely on the Internet! And the photo again captures this well!

In this course, we will study how the foundations of the Internet work.

Qn: BUT, before we go further, what do YOU think IS the Internet? What exactly IS the Internet?

Almost all of those were right answers. So let me use the next couple of slides to throw around some terminologies, some acronyms that define the Internet, and basically I’ll end up confusing you so much that it’ll take the rest of the semester to clear it up!

So if you take the simple example of web browsing, which is still a prominent Internet application, what we’ll learn in this course is what happens between-
- Your typing the address of a web-page, and
- the web-page appearing on your browser.
Specifically, the questions you should be able to answer by the end of this course are:

- In-order example: Ordering materials from a construction supply warehouse and having them arrive in the order they are needed.

There are at least two ways to think about the Internet (and also about what we will cover in this course):

- The first is in terms of the components that make up the Internet. So this would be a nuts-and-bolts view in which we can talk about several acronyms and terms that you may have come across, such as the:
  - web,
  - TCP/IP,
  - inter-network,
  - routers,
  - switches,
   and so on…

- The second way to think about it is the one that is more common, which is in terms of the services and applications that run over the Internet. So all of us do web-browsing, exchange emails, share files.

The important point to note is that these applications we care about do not operate solely by themselves. There are numerous services that all work together to provide a seamless view that an application is doing everything by itself.

—

In this class we expose some of the seams.

The services that enable such applications are all part of the Internet. So we will spend time talking about how these services work?

—

What is the Internet?

User's view: A bunch of computers that can talk to each other. Lots of end systems:

- Thin clients (“terminals”),
- handhelds,
- Network appliances (“toasters”).

Physical view: A bunch of wires or… a mesh of routers.

- If viewed as a “graph” the main components are the edges and nodes.
- All of this is typically transparent to the average user.

In this course we’ll focus on both the end-systems and the goo in the middle.

- Our treatment of the end-systems and the applications they run distinguishes this course from a traditional networking course.

Discussion: How big is a packet?

(Explain the figure!)
The second view is from the perspective of the "network architecture" – exemplified by the network core (or the goo in the middle).

The distinguishing features of the architecture are:

1. How is data switched? Is it sent as a continuous electromagnetic signal, or is broken down into chunks? And if so, how big are the chunks?

   The Internet is an example of a packet-switched network, in which data is broken down into chunks and each is sent independently across the Internet to its destination.

In circuit switching we assume sources transmit (or are capable of transmitting) continuously.

The other main issues that need to be resolved by the network core in a packet-switched network are:

2. How do you identify and address the different elements that are involved in a communication?

3. And how do you route data through a massively-sized network to the intended destination?

A third way to look at the Internet is in terms of the services/protocols it runs:

The network edge

- End systems (hosts)
  - Live at the "edge of network"
  - Run applications
    - Web browser, email, IM, eDonkey, iTunes, telephony, …
    - Run servers

- Each application/service has its own Protocol
  - HTTP, FTP, SMTP, …

- Transport Layers (TCP, UDP) provide "end-to-end" services to applications
  - Reliable data delivery
  - Flow and congestion control

**Network Structure**

The network core

- A mesh of interconnected routers
  - Routers connected by communication links with differing speeds, latency, error rates, …

**The fundamental questions:**

- How is data switched?
  - Packet switching

- How are elements identified?
  - Scalable addressing

- How is data routed through the network?
  - Local and global routing

**Network Structure**

The network edge

- End systems (hosts)
  - Live at the "edge of network"
  - Run applications
    - Web browser, email, IM, eDonkey, iTunes, telephony, …
    - Run servers

- Each application/service has its own Protocol
  - HTTP, FTP, SMTP, …

- Transport Layers (TCP, UDP) provide "end-to-end" services to applications
  - Reliable data delivery
  - Flow and congestion control
So, what is this course about?
The Internet food chain of technology

- Application-level protocols
  - HTTP, FTP, SMTP (e-mail), and the Domain Name System (DNS)
- Socket programming and client/server computing.
- Transport protocols
  - TCP and UDP
- Congestion control principles and algorithms
- The Internet Protocol IP and Internet routing architecture and algorithms
- Physical-layer transmission technologies
  - Ethernet, Wireless LANs, modems

COMP 431 Administrivia
Prerequisites

- I assume that you have had:
  - COMP 411 — Computer Organization
  - COMP 410 — Data Structures
  - A working knowledge of the UNIX program development environment
- I also assume you are:
  - Comfortable programming in Java, especially String and I/O classes
- Program operation will be described using UNIX terminology
  - You should be comfortable with the UNIX file system, file I/O, I/O redirection, basic UNIX program development
  - Online UNIX mini-tutorial this week (get it working by next Tuesday)
    1. Get a CS dept account!
    2. Install and try out a ssh client to access classroom.cs.unc.edu
    3. Try out all examples in tutorial
    4. Complete HW 1

Applications
- Top-half operating system services (APIs)
- Bottom-half operating system services (end-system networking)

Chapter 5 is optional and we won’t likely get to it.

I assume you can write at least a Java program on UNIX without help.

See HW 1!

Ask for show of hands on how many have never run Java programs on UNIX before!!

See if there are students in the class who need departmental UNIX accounts!!

Unix mini-tutorial will be posted online --- your responsibility to grow through it, try out the examples, and go to TA office hours for troubleshooting!! This includes: using ssh, editing on unix, running java program on unix, and simple unix commands for re-directing, etc.

Your responsibility to get departmental Unix accounts!!

Also make sure you have:

- dept unix accounts
- a SSH client that you have already used to log on to classroom.cs.unc.edu (TSC can help!!)
The textbook has a SIXTH edition available too, but as far as this course is concerned, all editions starting from the third edition would be fine, so feel free to get an older edition if it is much cheaper!

There is also a UNIX intro text you can buy (see the handout).

Notes and copies of all course handouts will be on the web.
If you miss class you are responsible for getting your own copy of the notes.

COMP 431 Administrivia
Textbook & Handy Java References

- *Computer Networking: A Top-Down Approach Featuring the Internet, Fourth Edition*

- *The Java Programming Language*
  - 3rd edition, by Ken Arnold, James Gosling, and David Holmes, Addison Wesley, 2000

- *Java in a Nutshell*

COMP 431 Administrivia
Other resources

- Course web page
  - [http://www.cse.unc.edu/~jasleen/Courses/COMP431](http://www.cse.unc.edu/~jasleen/Courses/COMP431)
  - Your source for copies of all handouts, homework assignments, lecture notes, ...
  - If you miss class (bad idea!) you can see if anything new was posted by checking the web site

- Piazza.com – class discussion forum
  - Post your questions here (rather than email)
  - Folks who post answers will get credit for class participation
    - Even if answers are incorrect
    - Anonymous posting allowed (I can see though)
  - Also used for announcements (instead of email)
    ⇒ check it regularly !! (can be set up to send emails once/day)
COMP 431 Administrivia
Grading (all percentages approximate within 10%)

- Programming and written assignments (~35%)
  - Different assignments will have different weightage
- In-class Quizzes – roughly, 1 per week (~15%)
  - Lowest score will be dropped
  - No make-ups allowed – make sure you attend classes!!
- Midterm examination – approximately, March 8 (~20%)
- Final examination (~30%)
  - Tuesday, May 3 at 8:00 AM
- Class participation used to raise grades by 0.3 grade points (e.g. add + to a letter grade; B+ ---+ A-)
  - Both in-class and piazza participation (answering questions) counts
- Sorry, but auditing of this course is not allowed
- All grades will be posted on sakai.unc.edu

COMP 431 Administrivia
Programming and written assignments

- **Written homework** will be collected on sakai and is due by 8:30 am on the specified due date
  - Anything submitted after 8:29:59.99 EDT/EST is considered late
- **Programming assignments** are due one hour **before** the start of class and are submitted by email to TAs
  - Programs time-stamped after 8:29:59.99 EDT/EST on the due date are late!
### Late homework policy

- Late submissions of **written** homeworks will **NOT** be accepted.
  - A grade of “0” if not submitted on time.
- Late submission of **programming** homework is “OK” but...
  - Only if it’s not too late
  - You’re not late too often; And you don’t miss class to get it done
- You have 4 “late days” that may be distributed as you please among all of the **first 4** programming assignments
  - Once the 4th late day is used up there will be no extensions!
  - No late day can be used for the 5th programming assignment

**Warning:** The programming assignments build on each other!
- If you’re late on one assignment you’ll have to work faster on the next one
  - There will almost always be an outstanding programming assignment
- If you don’t get a perfect score on a program you’ll have to fix the problems before starting the next assignment

### Missing class policy

- No make-up will be given for missing quizzes, homeworks, exams
  - Make sure you attend all classes!
- Unless you have an “excused” absence, which includes:
  - Medical emergency, which includes a doctor’s note
  - A family loss/emergency (also with documentation)
- For “excused” absences,
  - Quizzes will be pro-rated
  - Written homeworks will be pro-rated
  - Programming assignments will need to be completed (and may include some pro-rated grade)
  - Mid-term may be pro-rated (from final),
    - Or you may be given a “make-up exam”
COMP 431 Administrivia
TAs & Grading

Qianwen Yin
qianwen@cs.unc.edu

Kevin Wang
wangk1@cs.unc.edu

- For gripes about homework grading please see the TAs first
  » If the dispute persists, contact the instructor

- For gripes about exam grading please see the TAs (and cc the instructor for sure)

COMP 431 Administrivia
Honor Code: Policy on collaboration

- Working in groups on assignments is OK but…
  » You can only collaborate with other students in this course
  » You can only collaborate on understanding the assignment and possible “approaches”
  » Every student must craft their own final solution
  » Every student must fully write up their own solution
  » All collaborators must be acknowledged in writing

- Programming assignments must be completed by each student individually
  » Code may never be shared
  » Collaboration on the mechanics of UNIX/Java programming is OK
  » Debugging or designing each other’s programs is not OK
For Details on Course Policies

- Read the Handouts dated January 12, 2016
  - Course Outline
  - Programming guidelines

READ THESE HANDOUTS CAREFULLY.

YOU ARE RESPONSIBLE FOR KNOWING THE GROUND-RULES FOR THIS COURSE!

COMP 431 Administrivia

Where to go for help?

- Ask questions in class!
- Attend office hours
  - My office hours are (tentatively) 12:15 – 2:00 pm on Thursdays in FB 136
  - TAs office hours (tentative) have been posted on the course web page
    - At least two hours of TA office hour each working week day
- Post questions on piazza:
  - Do not send clarification questions by email – post them on piazza.
  - Do expect that other students will be answering these (and instructor and TAs will “endorse” good answers).
  - Don’t depend on an immediate answer
Lectures should follow the text pretty closely but we add lots of material that isn’t in the book so if you miss class you’ll miss a lot!

I also purposely leave gaps in the lectures that I expect you to fill in. Therefore, you have to study.

—

These gaps will be made explicit in the form of think about this exercises tossed out in class.

—

This is done to get you extra means of assessing your understanding of material (w/o the penalty of getting a bad grade on an assignment/exam).

—

If you understand the question, great! If not, take this as a sign that you need to get help (or study more).

Classic scenario I: Before the midterm someone comes to my office and says I have no idea what this slide is about.

—

If you come to office hours and don’t have marked up slides I’m not going to talk to you (you’re going to die!).

Classic scenario II: After the midterm someone comes to my office and says You never covered this!

—

Promise: Everything on the exam will be explicitly covered in class!

I spend at least 8-10 hours outside of class preparing for this course!

—

And the 8-10 hour estimate for homeworks assumes you’re keeping up with readings and have been taking good notes in class!

COMP 431 Administrivia
How to get an “A” in COMP 431

• Attend class regularly
  » Ask questions!
    » Exercise your understanding of course material on a daily basis

• Read the book (and read the discussions on piazza)

• Study the homework and in-class “thought” problems
  » Don’t just “do” the homework

• Take (and study your) notes!
  » Beware the “But I understand this” syndrome

COMP 431 Administrivia
How to get (no higher than) a “D” in COMP 431

• Assume getting copies of handouts is sufficient
  » You still have to read the book!

• Don’t take notes in class
  » Handouts are purposefully incomplete!

• Miss class
  » What is said in class is important!

• Procrastinate and start assignments at the last minute
  » Assignments are designed to require 8-10 hours of work outside of class each week
**COMP 431 Administrivia**

Frequently asked questions

- I've never programmed in Java before, will I die in this course?
- I've never used UNIX before, will I die in this course?
- Will this course help me become Cisco Certified?

Other Questions?

**COMP 431 Programming:**

**First, An Introduction to Network Protocols**

- A protocol is a specification for a set of message exchanges
  
  - Human protocols
    - Get the time from a stranger.
      - Hi
      - Do you have the time?
      - Yes!
      - It's 2:00

  - Computer protocols
    - Get the CS class times from the web server
      - TCP connection request
      - TCP connection reply

Get http://www.cs.unc.edu/Admin/Schedules
COMP 431 Programming:
Building an FTP client & server

FTP: File Transfer Protocol
- transfer file to/from remote computer
- client/server model
  » client: side that initiates transfer
  » server: remote computer

You will write Java programs for all elements in this picture

COMP 431 Programming:
FTP Protocol elements

- Control messages (command and reply)
- File data

Sample commands:
- sent as ASCII text
- USER <username>
- PASS <password>
- RETR <filename>
  retrieves (gets) file

Sample replies:
- 331 Username OK, password required
- 125 data connection already open; transfer starting
- 425 Can’t open data connection
COMP 431 Programming: Building an FTP client & server

- You will build a working, conforming subset of the Internet FTP protocols over the course of 4-5 homework assignment (each assignment builds on prior ones).

- Your programs will have to **interoperate** with selected FTP programs used in this department (so you have to conform to the protocol specifications).

- HW1 (Step One):
  - Write a Java program that recognizes valid FTP commands (this code will be reused later in your FTP server).

COMP 431 Programming: Building an FTP client & server

A formal specification of some FTP commands:

```
<ftp-cmd> ::= <command>[<SF>]<parameter>[<CRLF>]
```

The following are some of the FTP commands:

- `USER<SP><username><CRLF>`
- `PASS<SP><password><CRLF>`
- `TYPE<SP><type-code><CRLF>`
- `SYST<CRLF>`
- `NOOP<CRLF>`
- `QUIT<CRLF>`

```
<username> ::= <string>
<password> ::= <string>
<type-code> ::= "A" | "I"
<string> ::= <char> | <char><string>
<CR> ::= the carriage return character
<LF> ::= the line feed character
<CRLF> ::= <CR> followed by <LF>
<SF> ::= one or more space characters
<char> ::= any one of the 128 ASCII characters except <CR> or <LF>
```
HW1
Java programming notes

- The machine you should use for programming is:
  » classroom.cs.unc.edu

- In this class, Java means whatever version of Java is currently installed on classroom

- You can develop your code anywhere you like but…

- Your programs will be tested on classroom and correctness will be assessed based only on their performance on classroom
  » Always make sure your program works on classroom!
  » For all types of test input!