

CS472 – Computer Networks: Theory, Applications and Programming  
Dr. Sean Grimes – [spg63@drexel.edu](mailto:spg63@drexel.edu) – Office: 1161

**Office Hours:**

Mondays 5pm – 6pm (office)  
Tuesdays 11:30am – 12:30pm (zoom): --  
Additional availability by appointment.

**Teaching Assistant:**

--

**Couse Objectives**

Introduction to computer networking theory, applications, and programming, focusing on large heterogeneous networks. Broad top-down introductions to computer networking concepts including distributed applications, socket programming, operating systems, router support, router algorithms, and sending bits over congested, noisy, and unreliable communication links.

This is a hands-on course. The lectures will have tooling and programming demonstration, and you will be expected to do similar things for homework and project assignments.

**Topics**

1. Introduction – OSI reference model, TCP/IP intro, RFCs, socket review
2. Network definition (and evolution)
3. Network concepts (packet and network switching, latency, resiliency, packets, encapsulation, etc.)
4. Network standards (RFCs)
5. Network protocols and PDUs
6. OSI conceptual reference architecture
7. Concrete IP network architecture: Ethernet, TCP/IP, UDP/IP, IPv4, IPv6
8. IP addressing (IP address classes, CIDR, ports, NAT, etc.)
9. Socket programming (systems programming)
10. Higher level network programming (HTTP frameworks, GRPC)
11. Application layer – HTTP, FTP, DHCP, DNS
12. Transport layer – TCP & UDP, deep dive into TCP
13. Network layer – deep dive into IP
14. Link layer (frames, error detection, ethernet)
15. Network security – TLS
16. Routing
17. Modern network considerations (WiFi and mobile)
18. Software defined networks (including cloud networking)

**Expected Learning**

By the end of this course the student should be expected to have a solid grasp on:

- Various networking protocols, and how these protocols work together
- The OSI and TCP/IP reference models
- Key network addressing concepts, IP addresses, CIDR, MAC addresses
- The difference between connectionless and connection-oriented protocols
- How protocols are designed and support each other, both horizontally and vertically
- Modern network architecture concerns – security, reliability, resiliency, congestion control
- How data gets routed around networks and different types of routing protocols
- Key aspects of virtual and physical network hardware – routers, switches, WiFi, etc.
- Hands on building of several different types of network protocols via the socket API

### **Textbook**

The lectures for this course will follow the flow of the course textbook.

Computer Network, A Top Down Approach, Kurose & Ross, 8<sup>th</sup> edition.

*Note that if you can find the 7<sup>th</sup> edition cheaper, you should be fine.*

We will be starting off with a broad overview of computer networks and then traverse topics starting at the high level (network concerns closest to the application), then work down the stack to the data link layer covering how bits flow over the network.

From time to time, the following free resources will be referenced – they are also linked under the “Useful Resources” tab on blackboard:

1. Computer Networks: A Systems Approach - <https://book.systemsapproach.org>
2. Computer Networks, Principles, Protocols & Practice - <https://github.com/cnp3/ebook/releases/download/draft-3rd/CNP3-2021.pdf>
3. Beej’s Guide to Network programming Using Internet Sockets - <https://beej.us/guide/bgnet/>

### **Tools (get these installed in your OS)**

1. “Unix” command line tools, e.g., netstat, arp, traceroute, ifconfig, etc.
2. Wireshark
3. C Programming environment (VSCode with gcc on my machine or VIM with gcc are fine)  
Emacs vs VIM....extra credit for team VIM.
4. Not needed right away but we will be using GNS3 and MiniNet for some course lab projects towards the end of the term.

### **Programming Expectations**

This full title of this course is Computer Networks: Theory, Applications, and Programming. As such, this course will require you to use a systems programming language to get some practical hands-on experience working with network protocols. I will be using C in this course and will accept homework in either C or C++. Many of the assignments will provide you with scaffolded code to get you started. We will be reviewing socket programming in class.

Throughout this course I will also be demonstrating some newer systems programming languages that have strong networking support such as Go and Rust. You must use a systems programming language for this course – so no Python, Java, Javascript. C is preferred given it is a prerequisite for this course. Also, there are many libraries out there that can be found to implement assignments in just a few lines of code. You are not allowed to use these libraries in your code but are strongly encouraged to study them.

### **Assignments and Labs – 5 in total.**

This course will have 4 assignments that will be spread out over the course (due approx. every 2-3 weeks). Each assignment will have both a programming and a non-programming deliverable.

Once we get through aspects of the network and transport levels, we will have a network design lab where you will design a simple network and try out some routing protocols to manage traffic across the network that you will be designing.

Given previous experiences teaching this class, students tend to have different levels of readiness, especially with background in programming. I will adjust deadlines for the whole class as necessary, based on my observation of student effort and questions being asked over Discord, and attending office hours (mine or the TAs). One off deadline extensions are not possible, but I do realize that things come up from time to time. If you go into grade center, you will notice that you have a bank of 5 late days to use over the term. Late submissions will be accepted without penalty by deducting from your late day bank. Once you exhaust your bank, late assignments will take a 25% penalty on day 1, a 50% penalty on day 2, and will not be accepted on day 3+. I am also going to make all assignments due at 8AM – just assume that the deadline was really midnight the day before, and I am giving you an 8-hour grace period. Day 1 of late starts at 8:00:01 AM on the deadline. **Unused late days will be applied as extra credit points to the homework portion of your final grade.**

### **Grading Breakdown**

60%: Homework, programming assignments, and labs

40%: Exams (midterm & final)

## Rough Schedule

- 1.(06/23) Introduction to computer networks, socket programming tutorial
- 2.(06/30) Introduction to computer networks, Application layer (maybe)
- 3.(07/07) Application layer
- 4.(07/14) Application layer (deep dive into HTTP 1.0/1.1/2.0/3.0)
- 5.(07/21) Network layer (UDP / TCP)
- 6.(07/28) **Midterm Exam (Monday)**, network layer (UDP / TCP) – deep dive into TCP
- 7.(08/04) Transport layer (IP)
- 8.(08/11) Transport layer (routing) – deep dive into routing protocols
- 9.(08/18) Data link layer
- 10.(08/25) Software defined networking
- 11.(09/01) **Final exam (sometime this week)**

## Grades

Grades (roughly) follow the standard CCI grading scale. I will always round to the nearest whole number. E.g. 92.51 → 93. I will not round up a 92.49, sorry. Line has to be drawn somewhere, and this is where I draw it. I do not give an A+ by default in my classes, an A+ will be earned through exceptional performance throughout the quarter, therefore 92.5 → 100 is an 'A' by default.

## University Policies:

This course follows university, college, and department policies, including but not limited to:

- Academic Integrity, Plagiarism, Dishonesty and Cheating Policy:  
<https://drexel.edu/provost/policies-calendars/policies/academic-integrity>
- Student Life Honesty Policy from Judicial Affairs:  
<https://drexel.edu/studentlife/community-standards/code-of-conduct/academic-integrity-policy>
- Students with Disability Statement: <https://drexel.edu/disability-resources/support-accommodations/student-family-resources>
- Course Add/Drop Policy: <https://drexel.edu/provost/policies-calendars/policies/course-add-drop/>
- Course Withdrawal Policy: <https://drexel.edu/provost/policies-calendars/policies/course-coop-withdrawal/>
- Drexel Student Learning Priorities:  
<https://drexel.edu/institutionalresearch/assessment/outcomes/dslp/>
- Office of Disability Resources: [http://www.drexel.edu/ods/student\\_reg.html](http://www.drexel.edu/ods/student_reg.html)
- Students requesting accommodations due to a disability at Drexel University need to request a current Accommodations Verification Letter (AVL) in the ClockWork database before accommodations can be made. These requests are received by Disability Resources (DR), who then issues the AVL to the appropriate contacts. For additional information, visit the DR website at <https://drexel.edu/disability-resources/support-accommodations/student-family-resources>, or contact DR for more information by phone at 215.895.1401, or by email at [disability@drexel.edu](mailto:disability@drexel.edu).

A note on the academic honesty policy, assignments / labs / exams must be completed by yourself and yourself alone unless otherwise explicitly stated. However, this is not CS172. Use your favorite search engine, stackoverflow, generative AI, etc... to look up APIs, how to use a library, how to do some specific thing in C as long as it's not the main point of the assignment. E.g. if the assignment is how to read data from a socket, looking up how to do this on stackoverflow, copying and pasting that code into your submission, is **not** okay. Any sources you used, even if code wasn't directly copied, need to be cited. **Note:** Do not use generative AI to generate part or whole assignments. You are free to use it to learn how APIs work, get explanations for different methods, or similar use cases. **Use for generating code for your programming submissions or answers to written submissions will be considered a violation of the academic honesty policy and you will earn yourself an 'F' for this course. Any use *at all* on an exam will result in an 'F' for this course.**

### **Course Change Policy**

The instructor may, at their discretion, change any part of the course during the term, including assignments, grade breakdowns, due-dates, and the schedule. Such changes will be communicated to students via the course web site Announcements page. This page should be checked regularly and frequently for such changes and announcements. Other announcements, although rare, may include class cancellations and other urgent announcements will be communicated via the course discord channel.