CMSI 355

NETWORKS

https://dondi.lmu.build/spring2020/cmsi355

Spring 2020—Seaver 207
TR 11:20am–12:35pm, 3 semester hours
Office Hours: MTR 4–6pm, or by appointment

John David N. Dionisio, Ph.D.
email: *dondi@lmu.edu*
Doolan 102; (310) 338-5782

Objectives and Outcomes

This course seeks to introduce you to the computer science subfield of *networks*. As with all aspects of computing, computer networks involve particular data structures and algorithms. But because we leave the confines of a single computer, we encounter distinctive issues in areas such as latency, security, and error-handling. Long after the course concludes, my hope is that you will be able to:

1. **Understand, differentiate, and navigate a network's layers.**
2. **Perform assorted monitoring, administration, security, and communication tasks using various network utilities.**
3. **Write programs that communicate with programs on another computer at the socket and application levels.**

In addition to the course-specific content, you are also expected to:

4. **Follow disciplinary best practices throughout the course.**

Prerequisites/Prior Background

Programming proficiency in at least one high-level language; familiarity with how information is encoded at the bit/byte level; knowledge of data structures and algorithms in general.

Materials and Texts

This course does not have a preassigned textbook, with materials consisting primarily of assorted websites, articles, videos, and sample code to be made available online. However, the following sources can serve as foundational reading:

- Douglas E. Comer, *Computer Networks and Internets*, 6th edition, Pearson, 2014
- James Kurose and Keith Ross, *Computer Networking: A Top-Down Approach*, 7th edition, Pearson, 2017 (appears to have replaced Comer in the Pearson catalog)
- Larry L. Peterson and Bruce S. Davie, *Computer Networks: A Systems Approach*, 5th Edition, Morgan Kaufmann, 2011

Having the latest exact edition isn't critical, so no problems if an older edition is more cost-effective for you.

In addition, this series is considered to be a classic and serves as an excellent, in-depth reference:

- W. Richard Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*, Addison-Wesley Professional, 1994
- Gary R. Wright and W. Richard Stevens, *TCP/IP Illustrated, Volume 2: The Implementation*, Addison-Wesley Professional, 1995
- W. Richard Stevens, *TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP, and the Unix Domain Protocols*, Addison-Wesley Professional, 1996

If you can pick just one, get *Volume 1: The Protocols*.

Course Work and Grading

Your final grade will be based on the percentage of the points you get for the following deliverables against the total number of possible points:

GitHub and YouTube account listing	20 points
Network exploration log	100
One-shot network services	100
…in another language	100
Full application server	200
…and client(s)	100
Total	620 points

Percentages ≥ 90% get an A– or better; ≥ 80% get a B– or better; ≥ 70% get a C– or better. I may nudge grades upward based on qualitative considerations such as degree of difficulty, effort, class participation, time constraints, and overall attitude toward the course.

Term Portfolio

Your accumulated assignments for the semester comprise the *term portfolio*—the final, definitive artifact that demonstrates the course's outcomes. It is how you show whether you have, indeed, accomplished the objectives of this course.

An assignment's number is its due date in *mmdd* format, and it is always due by 11:59:59.999pm of that date. Point values are based on the state of your assignments at that moment.

Your portfolio for this course will consist of the following types of deliverables:

- Network exploration log—a document demonstrating your knowledge and proficiency with networking concepts and tools (*1a–1c*, *2a–2c*)
- One-shot network services—a suite of simple networked services, implemented in two different languages as a concrete demonstration of network abstraction (see what I did there) (*3a*)
- Full application server and client(s)—a larger-scale networked service such as an online game of appropriate scale, with one or more client implementations (again to demonstrate the abstraction afforded by a well-designed service) (*3b*)

Outcomes *4d–4f* apply to all assignments; the full range of *4a–4f* applies to all assignments that involve programming.

For maximum enthusiasm and interest level, the exact functionalities for the programming assignments have not been predetermined and will be pitched and workshopped in class before work on the assignment begins. So start brainstorming now so that we can have a productive specification section when the time comes.

Version Control

Version control is an indispensable part of today's computer science landscape in industry, the academe, and the open source community. We use version control heavily in this course: make sure that you get the hang of it.

None of the assignments can be completed (well) overnight; they should be the result of steady progress from the moment they are assigned to the date they are due. "One and done" submissions will negatively affect the final score.

Workload Expectations

In line with LMU's *Credit Hour Policy*, the workload expectation for this course is that for every one (1) hour of classroom instruction (50 scheduled minutes), you will complete at least two (2) hours of out-of-class work each week. This is a 3-unit course with 3 hours of instruction per week, so you are expected to complete $3 \times 2 = 6$ hours of weekly work outside of class.

Attendance

Attendance at all sessions is expected, but not absolutely required. If you must miss class, it is your responsibility to notify me about this and keep up with the course. The last day to add or drop a class without a grade of W is January 17. The withdrawal or credit/no-credit deadline is March 20.

Academic Honesty

Academic dishonesty will be treated as an extremely serious matter, with serious consequences that can range from receiving no credit to expulsion. It is *never* permissible to turn in work that has been copied from another student or copied from a source (including the Internet) without properly acknowledging the source. It is your responsibility to make sure that your work meets the standard of academic honesty set forth in:

http://academics.lmu.edu/honesty

Americans with Disabilities Act

Students with special needs as addressed by the Americans with Disabilities Act who need reasonable modifications, special assistance, or accommodations in this course should promptly direct their request to the Disability Support Services Office (DSS). Any student who currently has a documented disability (physical, learning, or psychological) needing academic accommodations should contact DSS (Daum 224, x84216) as early in the semester as possible. All discussions will remain confidential. Please visit *http://www.lmu.edu/dss* for additional information.

Topics and Important Dates

Correlated outcomes are shown for each topic. Specifics may change as the course progresses. University dates (italicized) are less likely to change.

January	Overview; conceptual and hands-on to networks and the Internet: history, terminology, layers, tools *(1a–1c)*
January 17	*Last day to add or drop a class without a grade of W*
February	Socket-level network programming; simple servers and clients; well-known non-web protocols; network load testing *(1c, 2c, 2d)*
March	Application-level network programming; HTTP; web services; WebSockets *(2c, 2d, 3a, 3b)*
March 9–13	*Spring break; no class*
March 20	*Last day to withdraw from classes or apply for Credit/No Credit grading*
March 31	*Cesar Chavez Day; no class*
April	Security and cryptography as applied to networks; "cloud computing" and managed services *(2a–2d, 3a–3c)*
April 8–10	*Easter break; no class*
May 5	*Last set of term portfolio assignments due*

You can view my class calendar and office hour schedule in any iCalendar-savvy client. Its subscription link can be found on the course web site (it's too long to provide in writing).

Tentative Nature of the Syllabus

If necessary, this syllabus and its contents are subject to revision; students are responsible for any changes or modifications distributed in class or posted to the course web site.

Course Evaluations

Student feedback on this course provides valuable information for continued improvement. All students are expected to fairly and thoughtfully complete a course evaluation for this course. This semester, all course evaluations for the Seaver College of Science and Engineering will be administered online through the Blue™ evaluation system. You will receive an email notification at your Lion email address when the evaluation form is available. You may also access the evaluation form on the Brightspace dashboard during the evaluation period.

A few minutes of class time will be reserved for you to complete a course evaluation within a fortnight before finals week. Please bring a laptop, smart phone, tablet or other mobile device to class on this date so that you can access the online evaluation platform.

Course Outcomes

1 Understand, differentiate, and navigate a network's layers.

1a	*Know the underlying structure of modern networks.*	Independent of the specific technology (or iteration of that technology), modern networks follow a specific set of abstractions and roles to facilitate the effective transmission of data across all of its connected devices. These outcomes pertain to understanding these abstractions at a level that should hold true regardless of how specific technologies evolve, deprecate, or emerge.
1b	*Describe a network's layers and their roles.*	
1c	*Follow how data travels through a network to get from one computer to another.*	

2 Perform assorted monitoring, administration, security, and communication tasks using various network utilities.

2a	*Use common tools and utilities to inspect and explore a computer's networking settings and environment.*	These outcomes can be classified under an overall category of "network agility:" the ability to navigate and maneuver a networking environment at a concrete, hands-on level. The hope is for you to understand what is "under the hood," and to no longer feel like a computer network is an inscrutable black box.
2b	*Interpret raw network traffic as seen through these tools and utilities.*	
2c	*Send and receive data through these tools and utilities.*	

3 Write programs that communicate with programs on another computer at the socket and application levels.

3a	*The socket level is the closest that we can get to "direct" communication without having to know specific network implementation details.*	Most of the course work will focus on direct and prolonged exposure (in a good way) to network programming. After writing such programs, the hope is that you will gain a greater appreciation of the more widely-used frameworks and libraries that allow software developers to concentrate solely on their application's functionality rather than worry about how data travels from one place to another.
3b	*The application level concentrates on specific functionality and tasks that users need to accomplish.*	

4 Follow disciplinary best practices throughout the course.

4a	*Write syntactically correct, functional code.*	Code has to compile. Code has to work. No errors, no bugs. Use unit tests as much as possible.
4b	*Use coding best practices, demonstrating principles such as DRY, proper separation of concerns, correct scoping of variables and functions, etc.*	This is the basis of good software design. It makes software easier to maintain, improve, and extend. Heed feedback well. *What you learn here will apply to future work in this field, in school and beyond.*
4c	*Write code that is easily understood by programmers other than yourself.*	This outcome involves all aspects of code readability and clarity for human beings, including but not limited to spacing & indentation, proper naming, presenting code in a manner that is consistent with its structure, documentation & comments when appropriate, and adherence to conventions or standards.
4d	*Use available resources and documentation to find required information.*	The need to look things up never goes away. Remember also that the course instructor counts as an "available resource," so this outcome includes asking questions and using office hours.
4e	*Use version control effectively.*	In addition to simply using version control correctly, effective use also involves appropriate time management, commit frequency, and descriptive commit messages.
4f	*Meet all designated deadlines.*	