

# CMSI 355T

## NETWORKS

<http://myweb.lmu.edu/dondi/spring2013/cmsi355>

Spring 2013 — Doolan 106

Variable sessions and meetings, 3 semester hours  
Office Hours: TR 10am–12pm, or by appointment

John David N. Dionisio, PhD  
email: [dondi@lmu.edu](mailto:dondi@lmu.edu)  
Doolan 106; (310) 338-5782

### Objectives and Outcomes

This course is a tutorial, individual-study exploration of the computer science subfield of *networks*. When one machine communicates with another, you have a network. It is hoped that, after self-study of this field, you will be able to:

1. Differentiate and navigate the OSI network model's layers.
2. Perform monitoring, administration, and communication tasks using current Internet protocols.
3. Write programs that use current Internet protocols at a sufficiently native level.

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

4. Follow academic and technical best practices throughout the course.

### Prerequisites/Prior Background

Programming proficiency in a systems-level language, particularly C; prior courses in computer system organization (LMU CMSI 284 or equivalent) and operating systems (LMU CMSI 387 or equivalent). Familiarities with shell scripting and system administration are also beneficial.

### Materials and Texts

- W. Richard Stevens, *TCP/IP Illustrated, Vol. 1: The Protocols*. Addison-Wesley Professional, 1993.

(Volumes 2 and 3 of this series may also be of value, depending on where your individual studies take you.)

In addition, do not hesitate to look for further information regarding the concepts, techniques, tools, and paradigms that we will discuss.

### Course Work and Grading

This course uses standards-based grading: your proficiency in each course objective is directly evaluated according to the outcomes shown on page 4 of this syllabus. Proficiency is measured according to the following key:

+	Advanced proficiency
	Appropriate proficiency
/	Approaching appropriate proficiency
-	Needs practice and support
O	No basis for evaluation

Your submitted work is used to evaluate these outcomes (see below). Letter grades are then assigned as follows:

	+		/	-
A	many		none	none
B		many	few	none
C			some	few
D				some
F				many

A-, B+, B-, C+, and C- grades may be assigned based on “close calls” along the proficiency measure thresholds and qualitative considerations such as degree of difficulty, effort, class participation, time constraints, and overall attitude throughout the course. You may inquire at any time about the proficiency measures that I currently have on record for you.

## Work/Study Journal

You will maintain an online work/study journal of your progress on a blog site of your choosing. This site should contain as complete a record as possible of your tasks, methods, and results throughout the term. Your study journal will be evaluated according to the degree to which its entries demonstrate the outcomes listed for this course (see below). The frequency and number of journal entries affects your proficiencies in that they increase the amount of evidence shown for the various outcomes.

The study journal will be evaluated at the end of the term, on May 10.

## Scripts and Programs

Some outcomes are best demonstrated by writing programs that perform various tasks. Suggestions for such programs will be made during the semester, but in recognition of the individual studies format of this course, you are also free to write programs of your choice.

Source code should be stored on a version control repository of your choice. Your overall “suite” of scripts and programs will be evaluated at the end of the term, on May 10.

## Term Portfolio

Your accumulated study journal blog and source code for the semester comprise the *term portfolio*—the final, definitive artifact that demonstrates the proficiencies you have reached for each course outcome. Although the term portfolio is not materially distinct from the items described previously, it is useful to think of it as a final target as the end of the semester approaches: review and clean up all of your blog entries, finalize scripts and programs, produce wrap-up content such as an overall summary or software documentation.

As stated previously, your study journal and version control repository as of May 10 will comprise the final version of your term portfolio.

## 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.

## Attendance

Meeting and session schedules are determined individually, and may vary according to the specific subject matter and/or course work. The last day to add or drop a class without a grade of W is January 18. The withdrawal or credit/no-credit deadline is March 22.

## University Policy on Academic Honesty

Loyola Marymount University expects high standards of honesty and integrity from all members of its community. All students are expected to follow the LMU Honor Code and Process, as stated in the *LMU Undergraduate Bulletin*.

## 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 (DSS) Office. Any student who currently has a documented disability (physical, learning, or psychological) needing academic accommodations should contact DSS (Daum 224, x84535) 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

Nothing specific here; this is dependent on the specific subject matter and on-going progress.

# Course Outcomes

## 1 Differentiate and navigate the OSI network model's layers.

1a	List and describe the complete OSI network model.	The OSI network model (ISO/IEC 7498-1) is the cornerstone of modern network architectures, defining the layers of abstraction that enable the great flexibility and adaptability that is possible today. The current Internet is a concrete (but not exact) implementation of this model, and knowing how to interact with <i>its</i> protocols serves as an application of one's knowledge of these networking abstractions.
1b	State current networking commands, parameters, configuration files, and other real-world artifacts of the OSI network model's layers.	

## 2 Perform monitoring, administration, and communication tasks using current Internet protocols.

2a	...with current networking commands.	The approach of this networks course is highly applicative and less theoretical. Thus, it is important to demonstrate the ability to <i>do</i> things with real systems as a reflection of your knowledge and understanding of computer networks.
2b	...by correctly modifying network configuration data.	

## 3 Write programs that use current Internet protocols at a sufficiently native level.

3a	Implement one or more Internet protocols at the packet level, effectively mimicking the behavior of one or more standard networking utilities.	The networking commands and utilities that exist today did not just appear— <i>someone</i> had to write them. As a computer scientist studying networks, a part of claiming knowledge and understanding of the subject includes the ability to do this on your own.
3b	Design and implement one or more <u>original</u> Internet protocols at the packet level for a custom application.	

## 4 Follow academic and technical 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	Demonstrate proper separation of concerns.	This is the basis of good software design. It makes software easier to maintain, improve, and extend. Proper separation of concerns includes but is not limited to correct scoping of variables & functions and zero duplication of code.
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 documentation & comments, spacing & indentation, proper naming, 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 commit frequency and descriptive commit messages.
4f	Meet all designated deadlines.	This tutorial course does not have the same formal due date or deadline structure as standard courses. Instead, your proficiency in this outcome is reflected in your ability to move at a reasonable pace given the semester hours devoted to this course. This includes regularly-spaced study journal entries and a steady rate of programming output.