Fundamentals of Computer Networking	**Course Syllabus**
CS4700/CS5700 Spring 2011	**1 January 2011**

Location	208 West Village H
Time	Mondays, 6:00pm–9:00pm
Web site	`http://www.ccs.neu.edu/course/cs5700/`
Forum	On Blackboard
Instructor	Prof. Alan Mislove
Contact	`amislove@ccs.neu.edu` (please put "[CS5700]" in the subject line)
Office hours	Tuesdays, 3:00pm–5:00pm, 250 West Village H
Teaching assistant	Abutalib Aghayev
Contact	`aghayev@ccs.neu.edu`
Office hours	Wednesdays, 4:00pm–6:00pm, 212 West Village H

DESCRIPTION

Computer networking is a rapidly advancing field. The Internet is already an integral part of society. It is therefore important for computer scientists and computer engineers to be familiar with the fundamentals of computer networking. This course will emphasize on the architecture, algorithms, and protocols of the Internet. Topics include local area networking, routing, congestion control, network security, and applications such as peer-to-peer and content distribution networks. This course will be project-centric, and students will work in groups on hands-on projects to learn how to build Internet applications as well as network protocols. The graduate version of this course will have additional requirements for each of the homework and projects.

LOGISTICS

The class will meet for 12 three-hour sessions, plus a two-hour final exam. Each three-hour session will typically be divided into three parts, with short (five-minute) breaks between. The midterm exam will be administered during a regular class session.

PREREQUISITES

The official prerequisites for this course are CS 3600 and CS 2600 (or equivalents). I also expect you to understand the basics of computer architecture and operating systems, and to have experience implementing non-trivial systems-type projects. Specifically, you should know what a processor is, how memory is organized, and what the user/kernel boundary is. You should also be able to read UNIX manual pages, and be able to familiarize yourself with UNIX utilities.

This course will be project-centric, and *all students* will complete in projects in groups of two (or possibly three, if necessary). Thus, to succeed in this course, you must be able to work in a group. I will allow you to form your own groups, and the course staff will serve as a matching service if necessary. As you are free to choose your partner(s), I will not be sympathetic to complaints at the end of the semester about how your group-mates did not do any work.

Finally, to succeed in this course, you should have some experience with programming in C and UNIX development tools (e.g. make, gcc, gdb). It is also highly recommended that you become familiar with using a debugger, as this will greatly aid you in completing the projects. At a high level, you should be motivated, eager to learn, willing to work hard, and make up, on your own, any prerequisite deficiencies you may have.

GOALS

By the end of this course, I expect you to

- Understand the fundamentals of networking protocols, including protocol layering, ba- sic medium access including wireless protocols, routing, addressing, congestion control
- Understand the principles behind the Internet protocols and some application layer protocols such as `http`, `ftp`, and DNS, and a few peer-to-peer systems/protocols such as `BitTorrent`.
- Understand some of the limitations of the current Internet and its service model
- Understand the causes behind network congestion, and explain the basic methods for alleviating congestion
- Design, implement, and test substantial parts of real-world network protocols

TEXTBOOK

The textbook for the course is

> Larry Peterson and Bruce Davie. *Computer Networks: A Systems Approach, 4th Ed*. Morgan Kaufmann, 2007. ISBN 978-0123705488.

and is available from the campus bookstore. You are also recommended to get

> Donahoo and Ken Calvert. *TCP/IP Sockets in C*. Morgan Kaufmann, 2001. ISBN 978-1558608269.

Other useful texts and resources include

> James Kurose and Keith Ross. *Computer Networking: A Top-Down Approach, 4th Ed*. Addison Wesley, 2007. ISBN 978-0321497703.
>
> Andy Tanenbaum. *Computer Networks, 4th Ed*. Prentice Hall PTR, 2002. ISBN 978-0130661029.
>
> W. Richard Stevens. UNIX *Network Programming: Networking APIs: Sockets and XTI; Volume 1*. Prentice Hall PTR, 1998. ISBN 978-0134900124.

HOMEWORK ASSIGNMENTS

This course will have four homework assignments reviewing concepts in class. Homework assignments are to be done by each student individually.

Homework assignments are due at the beginning of lecture on the specified date.

PROGRAMMING PROJECTS

The goal of this course is to teach the both the fundamentals of networking, as well as how to write programs which use networks. As such, there will be four programming projects throughout the semester.

You will form groups of two people to do the programming projects.[1] To collaborate effectively, you should both be involved in all of the major design decisions. You should also determine a partitioning of responsibilities so that you can both work effectively in parallel. For example, one might be responsible for generating all the test code while the other is responsible for the main code. You may switch groups between programming projects.

Important: *You alone are responsible for finding a partner. The class forum (newsgroup) located on Blackboard is a particularly good resource for this. Breaks during lecture are also a good time to look for partners.*

As the graduate version of this course will contain extra requirements, it is strongly recommended (but not required) that teams be formed of either all undergraduates or graduate students. If any of the team members are graduate students, the project will be graded as a graduate student project and will not receive any additional credit.

For the first three projects in this course, we will allow you to use any (reasonable) language of your choice (the fourth project is required to be completed in Java, as it is based on provided starter code). However, we strongly recommend using Java or C to complete the programming projects. Regardless, the submitted code must compile and run without any special requirements on the CCIS Linux machines. If you have any questions about the applicability of a specific language, please email the instructor.

The TA has been instructed to grade in part on design and implementation style and to be increasingly strict about this as the semester proceeds. In other words, it is not enough to get a working solution; you must implement the solution in a clean way that would simplify making further enhancements. It will benefit you in the long run to work on your software engineering skills.

Programming projects are due at 11:59:59pm on the specified date.

FORUM

We will provide a Web forum (the URL is provided at the beginning of this document) that can be used by students to ask questions and exchange wisdom while completing the homeworks and projects in this course. Please use the forum to post questions and answers that may be useful to others. Specifically, questions of the form "How do I link foo?, "Does such-and-such option in the compiler work for you?, or "What is the precise interpretation of homework question III, part b? should be posted on the forum first. If you mail me (or other course staff) these questions, we might not be able to answer them on time.

LECTURES AND SCRIBING

In order to help the class to record information conveyed during lecture, each student will scribe one hour of lecture at some point throughout the term. Scribing consists of taking lecture notes,

[1] If necessary, one group of three will be allowed.

cleaning them up, expanding on examples, and, essentially, making so that they serve as a useful point of reference for the concepts described. Once completed, scribing students should email their scribe notes to the instructor, who will collate them and post them on the course web site. Scribing will count towards your participation grade.

EXAMS

There will be one midterm and one final. All exams will be closed book and closed notes, and computers are not allowed nor is any access to the Internet via any device. The exams will cover material from lectures, readings, and the projects. They will cover the material discussed during the first and second halves of the class, respectively (i.e., they are not cumulative).

GRADING

The breakdown of the grades in this course is

20%	Homework (5% each)
40%	Projects (2%, 10%, 13%, and 15%, respectively)
35%	Exams (15% midterm and 20% final)
5%	Participation

Each project and homework will include a breakdown and description of how it will be graded.

Any requests for grade changes or regrading must be made within 7 days of when the work was returned. To ask for a regrade, attach to your work a page that specifies (a) the problem or problems you want to be regraded, and (b) for each of these problems, why do you think the problem was misgraded.

LATE POLICY

Written homework assignments have strict deadlines. Homework handed in late will be marked off 20% per day.[2] Homework more than 2 days late will not be accepted. Extensions will not be granted.

For programming projects, we will use flexible slip dates. Each student is given an automatic extension of 4 calendar days for the semester. You can use the extension on any project during the semester in increments of a day. For instance, you can hand in one project 4 days late, or one project 2 days late and two projects 1 day late. The slip time will be deducted from each group member's remaining slip time.[3] This should let you schedule due dates around the due dates for other courses. After you have used up your slip time, any project handed in late will be marked off 20% per day. Projects more than 2 days late will not be accepted. Extensions will not be granted.

ON CHEATING

It's OK to ask someone about the concepts, algorithms, or approaches needed to do the assignments. We encourage you to do so; both giving and taking advice will help you to learn. However, what you turn in must be your own, or for projects, your group's own work; looking at or copying other people's code, solution sets, or from any other sources is strictly prohibited. In particular, looking at other solutions (e.g., someone else's solution to a similar project) is a direct violation.

[2] A "day" refers to 24 hours. Thus, a homework turned in 28 hours late will count as two days late.

[3] Note on slip days: Slip days can only be used if all group members have at least one remaining slip day. If not, the 20% off per day policy stated above will be used for all group members.

The project assignments must be *entirely* the work of the students turning them in. If you have any questions about using a particular resource, ask the course staff first.

All students are subject to the Northeastern University Academic Integrity Policy, available at `http://www.northeastern.edu/osccr/academichonesty.html`. All cases of suspected plagiarism or other academic dishonesty will be referred to the Office of Student Conduct and Conflict Resolution (OSCCR).

ADVICE

As the course is project-centric, students are recommended to start early on projects. The projects require substantial design, implementation, and testing effort, especially for students who are unfamiliar with network programming. Students are encouraged to drop by the instructor's or teaching assistant's office hours (or set up a separate appointment via email) if they run into problems while completing the assignments.

Finally, computer use during class is allowed but expected to be for class purposes (e.g. note-taking, reference) only. Please avoid non-class-related computer use so that I do not have to revisit this policy.