CSC458/2209 – Computer Networks
Fall 2015, Section L0201
Department of Computer Science
University of Toronto

Handout #1 – Information Sheet  Date: Thu, September 17th, 2015

Class Hours: Thursday 1:00-3:00 p.m.  Tutorial Hours: Friday 11:00-12:00 a.m.
Class Location: SS1085  Tutorial Location: SS1085
Instructor: Dr. Ayse Karaman
Email: karaman@cdf.toronto.edu
Office: BA3219
Office Hours: Thursday 3:00-4:00 p.m.

Course Description
This is an introductory course on computer networks. Topics covered in this course include packet switching systems, socket programming, network software, hardware, and protocols, network naming and addressing, congestion control schemes, software-defined networking, network security, and wireless networking. The emphasis of the course is network programming and applications.

Prerequisites
You need to have a basic understanding of probability theory, a strong background in C programming, and be familiar with the Unix operating system. The first programming assignment (link available at class web page) is a typical example of the type of work you will be doing in this course. If you still are not sure whether you have the sufficient background to take this course, please send me an email.

Teaching Assistants
- Alireza Shekaramiz (alirezashekaramiz@gmail.com)
- Mohsen Norouzi (mohsen.norouzi@gmail.com)
- Hao WANG (wh.sjtu@gmail.com)
- Saeed Arezoumand (saeed.arezoumand@gmail.com)
- Lilin Zhang (llzhang@cs.toronto.edu)
- Seyed-Hossein Mortazavi (hosein.mortazavi@mail.utoronto.ca)
- Keran Yang (keran.yang@mail.utoronto.ca)
- Shuhao Liu (shuhao@ece.toronto.edu)
- Shiva Ketabi (shiva.ketabi@gmail.com)
Bulletin Board and Class Mailing List

Please use the bulletin board for all your general and/or technical questions. Before posting your question, please check the board to see whether it has already been asked by one of your peers. Everyone in class can see all the questions that have been asked and the replies they received. This would make more efficient use of the board for all of us. We guarantee that any question posted on the bulletin board will be responded within 48 hours.

The class mailing list will be used by the TAs and the instructor for announcements. Please use this list exclusively for hearing these announcements and never ask your questions there.

For questions of personal/private nature that can not be posted on the bulletin board (e.g. your grades), please send direct emails to me or to the TAs.

Exams
- **Midterm**: will be held on October 29th in class.
- **Final**: TBA – please check class web page for date and location of the final exam.

Textbook

Recommended Books

Grading
- Assignments: 50%
  - Problem sets: 20% -- 2*10%
  - Programming assignments: 30% -- 2*15%
- Midterm exam: 20%
- Final exam: 30%
Assignments

There will be two problem sets, both based on the textbook and the material covered in class.

There will also be two programming assignments. Here is a summary of the requirements for both programming assignments.

- You can work in teams of 2-3 as well as individually.
- We will use a virtualized network environment, MiniNet, for programming assignments.
- Each student will get access to a virtual machine (VM) on CDF servers where s/he can complete the programming assignments.
- To ensure compatibility with our marking scripts, please make sure you only use the VM provided through CDF.
- All programs must be written in ANSI “C”. To make the grading uniform, we can't accept assignments in C++, Python, Java, Perl or any other languages.
- Additional information and requirements will be specified in each assignment.

Late Submission Policy

You have one free late submission of 24 hours for one of the assignments (problem set, or programming, but not both). You should e-mail the TAs before the deadline to get the free late submission. This 24 hour limit is hard, and cannot be extended. For any late submission other than the free one, 10% of the mark will be deducted for each day late, up to 20%. Assignments will not be accepted after two days.

Academic Offenses

“Briefly, an academic offence is a bad thing done to get marks you don't deserve. Slightly more formally, an academic offence is an action by a student or course instructor that breaks the rules about academic credit at the University of Toronto.” Cheating is considered a very serious offense. Please avoid it! We are all here to teach and learn after all, and concerns about cheating make an unpleasant environment for everyone.

Permitted Collaboration

The following items are encouraged and allowed at all times for all students in this class:

- Discussion of material covered during lecture, problem sessions, or in handouts
- Discussion of the requirements of an assignment
- Discussion of the use of tools or development environments
- Discussion of general approaches to solving problems
- Discussion of general techniques of coding or debugging
- Discussion between a student and a TA or instructor for the course

---

1 Jim Clark, “Advice about academic offenses”, http://www.cs.toronto.edu/~clarke/accoffences/
Collaboration Requiring Citation

Two students engaging in more detailed discussions must be careful to document their collaboration. Students are required to include the names of those who provide specific assistance to properly credit their contribution, in the same manner as one would cite a reference in a research paper. The expectation is that even with a citation, the author must be able to explain the solution.

Examples of Collaboration That Require Citation

- Discussing the “key” to a problem set or programming assignment. Problem set questions are often designed such that the critical concept takes careful thought and gaining that insight from someone else must therefore be documented.
- Discussing the design of a programming project. Design is a crucial aspect of the programming process and discussion can be valuable. Any design input received from others must be cited.
- Receiving assistance from another student in debugging code. While the TAs are the preferred source for advice, any detailed assistance from someone else must be cited.
- Sharing advice for testing. For example, if someone provides important information on lessons learned (“my program didn't handle the case where the value was 0”) that source must be credited.
- Research from alternative sources. Researching related topics, such as through the Internet, must be documented if the solution submitted is derived from the research information.

Unpermitted Collaboration

All submissions must represent original, independent work. Some examples of activities that do not represent original work include:

- Copying solutions from others. In particular, do not ask anyone to provide a copy of his or her solution or, conversely, give a solution to another student who requests it. Similarly, do not discuss algorithmic strategies to such an extent that you and your collaborator submit exactly the same solution. Use of solutions posted to websites, such as at other universities, is prohibited. Be aware that we photocopy some of the exams prior to handing them back.
- Using work from past classes. The use of another student's solution or the posted class solutions from a previous class constitutes a violation.
- Studying another student's solution. Do not read another solution submission whether in electronic or printed form, even to "check answers."
- Debugging code for someone else. When debugging code it is easy to inadvertently copy code or algorithmic solutions. It is acceptable to describe a problem and ask for advice on a way to track down the bug.²

² Parts of this note are based on handouts from Nick McKeown, and Tom Fountain, who teach CSC244a and EE182 respectively at Stanford. Some portions are based on similar collaboration policies written by Eric Roberts, Julie Zelenski, and the Computer Science Department at Brown University.