CSC 458 Computer Networks (L0101&L0505) Fall 2023

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.

Instructor Information

Name	**Email**	**Office Hour Location**	**Office Hour Time**
Peter Marbach	marbach@cs.toronto.edu *	BA5224	Tuesdays, 5-6 PM
Efran Meskar	e.meskar@mail.utoronto.edu *	BA2272	Tuesdays 3-4 PM

** Please write "CSC458" in the subject header of your emails.*

*** Email your instructor if appointments outside this time are required.*

Course Information

Information pertaining to this course will be available on Quercus (q.utoronto.ca). The course website will have course announcements & materials, assignments, a link to assignment submission website (MarkUs), and relevant readings. Important announcements will be made through the email registered on Quercus, but we expect you to check the course website frequently; some of the announcements may only be posted on the website and not send through email to keep your inbox less crowded!

Course Textbook

Required:	Peterson, Davie, *Computer Networks: A Systems Approach*, 5th ed., Morgan Kaufmann, 2011

Important Dates

Date	**Due**
11:59 PM EST, Sunday, October 1	Programming Assignment 0
11:59 PM EST, Sunday October 15	Problem Set 1
11:59 PM EST, Sunday, October 22	Programming Assignment 1
11:59 PM EST, Sunday, November 12	Video Assignment
11:59 PM EST, Sunday, November 19	Problem Set 2
11:59 PM EST, Sunday, December 3	Programming Assignment 2

Schedule

Week	Dates	Topics	Reading
1	Sep 12	Course Logistics/ Introduction	
2	Sep 19	Link Layer, Error Detection/Correction	Chapter 1
3	Sep 26	Interconnecting LANs; Internet Protocol (IP)	Chapter 2
4	Oct 3	The Internet Protocol; Routing and Forwarding	Chapter 3
5	Oct 10	Internet Topology and Routing	Chapter 4
6	Oct 17	Transport Protocol	Chapter 5
7	Oct 24	Congestion Control	Chapter 6
8	Oct 31	Queueing Mechanisms; Middleboxes	
9	Nov 6 - 10	-- ***Reading Week*** --	
10	Nov. 14	*Overlay networks*	Chapter 8
11	Nov 21	Network Security	
12	Nov 28	Software-Defined Networking	
13	Dec 5	Review	
15	Dec 9 – 20	-- ***Final Exam Period*** --	

Mark Breakdown

Component	Weight
Programming Assignments	30% (PA0 2%, PA1 14%, PA2 14%)
Problem Sets	20% (each 10%)
Video Assignment	20%
Final Exam	30%

- **All assignments must be done individually.**

- **Assignments:**
  - We will post detailed information about the assignments in separate announcements.

- **Late Policy:**
  - Submission deadlines will be Sunday night (11:59PM).
  - You can submit those assignment after the deadline with a penalty of 0.5% for each hour. For example, if you submit it 10 hours after the deadline, 5% of the mark will be deducted because of the late submission.
  - Submissions will not be accepted after 24 hours.

Academic Offenses

Briefly, an academic offense is a bad thing done to get marks you do not deserve. Slightly more formally, an academic offense is an action by a student or course instructor that breaks the rules about academic credit at the University of Toronto. Plagiarism and cheating are considered very serious offenses. You can refer to the university's *Code of Behaviour on Academic Matters* for more information about university's policies on academic offenses such as cheating and plagiarism.

Permitted Collaboration: The following items are encouraged and allowed for all students in this class:

- Discussion between a student and a TA or instructor for the course
- 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, coding, or debugging

Collaboration Requiring Citation: Two students engaging in more detailed discussions must document their collaboration, in the same manner as one would cite a reference in a research paper. For example:

- 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 credited.
- Sharing advice for testing. For example, if someone tells you that "my program didn't handle the case where the input size was 0", you must credit that person in your code/report.
- 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 assignments must be done individually, and all submissions must represent original, independent work. Some examples of activities that do not represent original work:

- Copying solutions from others. Do not ask anyone to provide a copy of his or her solution or, conversely, give a solution to another student who requests it, even to "check answers" after you are done. Similarly, do not discuss algorithmic strategies to such an extent that you and your collaborator submit very similar solutions. Be aware that we will use automated tools to find submitted codes and reports that are similar.
- Any use of someone else's solution is prohibited. This includes studying a classmate solution, a solution from previous semesters, or solutions from other universities.
- 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.