

CSC 458 Computer Networks (L0101)

Fall 2021

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 *	Online for first 2 weeks. TBD afterwards	Tuesdays 7-8pm
Sajad Shirali-Shahreza	shirali@cs.toronto.edu *	Online	Thursdays 14-15

* 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, recorded course lectures, link and instructions to connect to virtual office hours and classes, a discussion boards (as one of the Quercus features), 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! We will use the inverted classroom model this semester: the video lectures are posted on the website and the live classes will be used to have a quick review of the materials, Q&A, and solving example problems.

Course Textbook

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

Important Dates

Date	Due	Topic
11:59 PM EST, Sunday, September 19, 2021	Feedback Survey 1	
11:59 PM EST, Sunday, October 10, 2021	Programming Assignment 1	
11:59 PM EST, Sunday, October 17, 2021	Feedback Survey 2	
Tuesday, November 2	Midterm	Chapters 1-6
11:59 PM EST, Sunday, November 7, 2021	Feedback Survey 3	
11:59 PM EST, Sunday, November 14, 2021	Video Assignment 1	
11:59 PM EST, Sunday, November 28, 2021	Programming Assignment 2	
11:59 PM EST, Wednesday, December 8, 2021	Video Assignment 2	

Schedule

Week	Dates	Topics	Reading
1	Sep 9 – 12	Course Logistics/ Introduction	
2	Sep 13 – 19	Link Layer, Error Detection/Correction	Chapter 1
3	Sep 20 – 26	Interconnecting LANs; Internet Protocol (IP)	Chapter 2
4	Sep 27–Oct 3	The Internet Protocol; Routing and Forwarding	Chapter 3
5	Oct 4 – 10	Internet Topology and Routing	Chapter 4
6	Oct 11 – 17	Transport Protocol	Chapter 5
7	Oct 18 – 24	Congestion Control	Chapter 6
8	Oct 25 – 31	Queueing Mechanisms; Middleboxes	
9	Nov 1 – 7	Overlay networks	Chapter 8
10	Nov 8 – 14	-- Reading Week --	
11	Nov 15 – 21	Network Security	
12	Nov 22 – 28	Software-Defined Networking	
13	Nov 29–Dec 5	Review	
14	Dec 6 – 8	No Lecture	
15	Dec 10 – 21	-- Final Exam Period --	

Mark Breakdown

Component	Weight
Programming Assignments	36% (2 programming assignment, PA1 18%, PA2 18%)
Midterm Exam (online)	25%
Video Assignments	36% (2 video assignments, each 18%)
Feedback Surveys	3% (3 surveys, each 1%)

- **All assignments must be done individually.**
- **Midterm Exam:**
 - The midterm is open book: you can use your notes, slides, and the book to answer them. However, you must complete it individually, and the answers have to be your own answers that you obtained without any help from anyone else.
- **Video Assignments:**
 - There will be two video assignments, each worth 18% of term mark
 - Each video submission has to be between 15 – 20 minutes.
 - For each assignment, you must explain specific topics that are assigned to you in the video. The audience of your video are 4th year undergraduate students who did not take any course on computer networks.
 - Your submissions will be marked based on correctness, completeness, presentation, and creativity.
 - Videos are stored on MyMedia and the link to the video is submitted through MarkUs.
- **Programming Assignments:**
 - There will be two programming assignments.
 - We will use a virtualized network environment (MiniNet) for programming assignments.
 - All programs must be written in ANSI “C”.

- Programming assignment 1 worth 18% of term mark: 14% for correct functionality, 2% for coding style, and 2% for documentation (e.g., inline comments).
- You only submit completed code for the first programming assignment 1.
- Programming assignment 2 worth 18% of term mark: 10% for the written report (including items such as presentation and correctness), 4% for correctly generated graphs, and 4% for code that generates those graphs.
- You must submit a written (electronic and typed) report in addition to your code and generated graphs for programming assignment 2.
- You must submit your code and report through MarkUs.

- **Feedback Surveys:**

- There will be three feedback surveys during the semester. These surveys will enable us to collect feedback from all students through the semester to improve the course delivery.
- You receive 1% for just completing each survey. The answers that you provide to survey questions does not affect your survey mark or other assignment marks.
- In addition to these feedback surveys, we welcome and encourage any comments, suggestions, and feedback during the semester.

- **Late Policy:**

- Programming and video 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.