

CSC435 Distributed Systems Syllabus

Professor Clark Elliott

Spring 2022-2023

Logistics:

Class meets: Section 910 meets remotely, asynchronously. Section 901 meets on campus on Thursday evenings as needed, with required in-class, proctored exams. **All lectures are recorded for asynchronous study.**

Informal live Zoom meetings will take place weekly as needed to answer questions. The current Zoom link is always found under D2L | Content | Admin Documents

Class website: <http://depaul.edu/~elliott/435>

email: Elliott@depaul.edu

[Include "435: " prefix in subject line and MEANINGFUL mail header!]

Grader email: elliottgrading_AATT@gmail.com. (Administration of submissions only)

Course Management: d2l.depaul.edu

Textbooks:

Required text: Required text: van Steen, Maarten and Tanenbaum, Andrew S., (2017), "Distributed Systems, 3rd Edition," Published by Maarten van Steen ISBN-13: 978-1543057386. Has *occasionally* been freely available online. The electronic version is recommended with the free **downloadable** full local Kindle Reader software. (The web reader is horrible. Don't use it.) If you can't get it free, then buy the book. It is much less expensive than most textbooks.

Recommended background text (not required): Kurose, James F., and Ross, Keith W. (2007 - -) "Computer Networking: A Top-Down Approach Featuring the Internet, [any] Edition," Boston: Addison Wesley. (Later editions are available, but any version from 2007 on will contain the bulk of the background material if you want to save money.) This is a great text to have on your shelf.

Grading:

There may be slight changes in the balance of the grading scheme. If so, you will be officially notified. A complete breakdown of points is given online at D2L under the *Grades* tab.

Exams 36%
Quizzes 13.5%
Assignments 50.5%

Grading Scale:

95%	A
90%	A-
86, 83, 80	B+, B, B-
78, 74, 70	C+, C, C-
65, 60	D+, D

I reserve the right to raise the grade of a student that has demonstrated exceptional contributions in some particular portion of the class (forum participation, programming, research/study log, etc.).

I reserve the right to add occasional small bonus points to a student's course grade for showing leadership on the forums and in the assignments. These are subjectively assessed by the professor.

All grades are subject to [Academic Integrity Sanctions](#) See the class website, and below in this syllabus for details. Note that there are course grade *penalties* for academic integrity cases.

Topics:

We will follow the chapter outline in the textbook "Distributed Systems" by Maarten van Steen and Andrew S. Tanenbaum, though some material from the book will not be covered. We will have extensive additional material in the lectures taken from other sources. We will cover many aspects of distributed systems dealing with system structure, models of time, client/server protocols, state maintenance, security, distributed process coordination, Hadoop and so on. These topics are foundational in almost every contemporary area of computer science. We will implement client/server applications including a webserver, and the basics of public key encryption and blockchain technology.

Note that we *may* cover possibly unsettling topics such as the coming revolution in human-computer interfaces and the integration of human brain processing into distributed systems. If you are uncomfortable with these topics, stop watching the lecture and participating in discussions immediately, and send email to the instructor.

Class structure:

This is primarily a lecture and reading class. However, there is a significant programming component to help solidify the concepts.

Discussion forum participation is *strongly encouraged*.

Maintaining a written research/study log is *required*. You **MUST** use your own words to record the ideas covered or risk failing the class.

Class will be challenging. Students are expected to do the reading without prompting from the instructor. Some topics will be covered in the lectures, but because time is short a number of topics will be covered only in the reading, and will appear on exams.

Java or C++ programming background is required. However, if you program well in other similar languages you can probably pick up enough java to pass the class. The programming assignments are used only to guarantee deep understanding of the material. Thus, we will not be assessing programming style as long as the programs achieve the results. Programming tips may be discussed, but programming will not be taught.

Learning Goals:

At the end of class you will:

- Have a broad understanding of the central problems in distributed systems.
- Develop a “tool bag” of distributed systems concepts applicable to real world problems
- Have a good understanding of the compromises—the choices—that must be made when designing a distributed solution to IT problems.
- Have a good understanding of algorithmic approaches to distributed systems solutions.
- Know how to write basic programs that address certain challenging distributed systems problems.
- Have taken part in high-level discussions of distributed system problems of interest.
- Demonstrate master's-level knowledge of the course materials on exams.

Office hours for the course are available from my faculty link at cdm.depaul.edu

All assignments, the assignment schedule, and the course materials, are available online at either d2l.depaul.edu or the class website. D2L Dropbox is used for all assignments.

The specific readings are available at the class website but in general will follow the text, chapters 1-4 and 6, with much additional lecture material.

Submission File Formats:

All submissions to D2L MUST BE IN THE SPECIFIED FORMAT or they will not be accepted for credit. No other formats will be graded. Often this means in STANDARD ZIP FORMAT, including submissions of a single file for programming assignments. No 7zip files, no rar files. No exceptions. Contained within the ZIP archive, all text submissions must be made in Microsoft Word format, or in plain HTML, or plain text. NO PDF FILES. (Free programs are available to produce each of these formats.) Java files should be submitted in plain text form, within a ZIP file, suitable for command-line compile.

In some cases zip files are prohibited. Follow the instructions for each assignment.

Students are responsible for downloading their assignments after uploading, to make sure that files have not been corrupted. Corrupted files will not be graded.

NO LATE ASSIGNMENTS will be accepted for credit, unless otherwise noted. No exceptions.

Academic Integrity:

Students are required to view the short lecture on Academic Integrity and to pass the Academic Integrity Quiz with a 100% score before other coursework will be accepted for credit.

Cheating, plagiarism, and unethical conduct are not allowed, and will be sanctioned, including referral to the dean's office, and failure in the class (default) or (occasionally) failure on the assignment and an additional 20-40% reduction in the course grade. Please refer to the academic handbook by which rules you are expected to abide. This course will be subject to the university's academic integrity policy. More information can be found at <http://academicintegrity.depaul.edu/>. If you have any questions be sure to consult with the professor.

Violations include, but are not limited to: making false claims on any checklist for work that has not been done; including ANY un-cited work of others in any documents you turn in; turning in work, including any program, that has been authored by someone other than yourself and in some cases including *any* work of others, whether cited or not—see the rules for each assignment; using artificial intelligence tools to author text of any assignments.

Publicly sharing or posting online any prior or current materials from this course (including exam questions and/or answers) is considered to be providing unauthorized assistance prohibited by the policy. Both students who share/post and students who access or use such materials are considered to be cheating under the Policy and will be subject to sanctions for violations of Academic Integrity.

ChatGPT and other LLMs

Sadly, because it can be time-consuming and challenging to reliably detect the use of AI LLMs to produce written assignments, I have retired most such assignments until such time as I can determine that they can be fairly graded.

You are free to use ChatGPT (or other LLMs) for background research as you would Wikipedia and Google Search. However, you must (a) create actual finished text and programs yourself, and (b) state clearly that you used ChatGPT (or other LLM software) to help with your work. You are forbidden to copy and paste directly from an LLM such as ChatGPT while claiming authorship of the work. This is an academic integrity violation. I reserve the right to determine whether I believe an LLM was used to produce an assignment and (b) refuse to accept it for credit, and (b) initiate a plagiarism case. I am the sole judge.

Computer program plagiarism

If you write your own code, and your own comments, you will be fine and should have no concerns.

Students are not allowed to electronically copy someone else's whole program solutions to an assignment and modify them. The rules: (a) every line of code must come from your design and be typed in by you, and (b) you must develop your assignment from scratch (or from base code I give you), building the final solution a little bit at a time, based on your own programming solutions.

You *are* allowed to scan the web and textbooks to see *general* programming design for various kinds of sub-problems (e.g., how to implement an ArrayList, or how to create a comparator for a sorting algorithm) having nothing specifically to do with the assignment for which you are writing program code. If you use someone's ideas for simple utility functions, you should include a URL for where you found the work. One helpful rule: If copying and pasting an *electronic* copy of someone else's short utility code is useful in your own program, you probably should not be using it. If you write your own version *based* on what you have learned from them, you may be OK.

I've never seen it, but if you are concerned that your own original solution to a class programming assignment will be miraculously similar to someone else's work, then you can always create a series of subdirectories with checkpoints of your work as you develop your code. Now you have evidence of your development work.

Checklists

Checklists for this course are a **contract** between the student and the professor. The default for all checklist items is *No*. If you change an item to *Yes* you are claiming, absolutely, to have done the work indicated. An inaccurate checklist that claims work not actually done results in zero grade, a significant points penalty in the course, and, possibly a failing course grade, and sanction through the dean's office. We will *assume* that you are attempting to cheat your peers by claiming work that was not done. If you are really in doubt, change *No* to *Maybe* or *Probably* and give a short explanation at the bottom of the checklist. Examples would be, "I wrote the code and it usually works, but sporadically fails every tenth time or so" and "I wasn't sure if you meant three list items or three nodes; I did implement three nodes, but not three list items" and so on.

Example: Sam spends 35 hours on her jokeserver assignment, and does a very good, but not quite complete job on it. She turns it in before completing randomizing of the jokes (a minor part of the assignment worth only a few points). She checks *Yes* next to "randomizes jokes" and turns in her checklist with her assignment. Ordinarily she would have received 95 of 100 points for her work. **Result:** Zero points on the assignment. Additionally 20% course penalty resulting in a D in the class. Referral to the dean's office. No remedy.

Citations

There are a number of written assignments, and programs, turned in for this class. The default *assumption* is that any work you turn in is your own. If you submit ANY work of others that is not clearly cited as being the work of others, you will be sanctioned: zero grade on the assignment, outright failure in the class, referral to the dean's office for possible expulsion from the university.

Example: Louis spends 100 hours over the course of the quarter, completing a thirty-page ethics paper. He includes a really interesting pair of paragraphs that he found on the web, but fails to cite the original work and author. He submits his paper for grading. Ordinarily he would have received full credit for his paper, and possibly even extra points for exceptional work in this area. **Result:** Zero grade for his paper. Failure in the class. Referral to the dean's office. No remedy.

For some assignments *no work of others is allowed*, even if you have cited it.

Example: Eddie spends 120 hours over the course of the quarter producing an excellent 8,000-word research/study log for which the rules clearly state he is not allowed to use ANY work of others—even if he cites it. However, Eddie does not pay attention to the rule, includes two paragraphs from the web—clearly cited—in his log, then turns it in. **Result:** Zero grade for his study log. Failure in the class. Referral to the dean's office. No remedy.

When in doubt, cite! (But note that you still might be guilty of plagiarism if you've included too much of the work of others.) Change the font of included text, and possibly use quotation marks, to make absolutely clear that it is the work of others. Make absolute certain that you are allowed to include the cited work of others in a particular assignment. (E.g., not allowed in the study logs!)

Gray Areas

The concepts and programming constructs in this class have been well-covered by others. There will be little that qualifies as original research. As long as you *write your own programs* and write all of the text that you submit (that is, the keystrokes are generated by your own brain) you should be fine, even though there may be small overlaps with the work of others. Example: a small utility java function for iterating through a list that appears in more than 100 forms on the web.

Example: In his programming assignment Joe included his own version of a simple utility list iteration function he found on the web in an unrelated program. He included the URL where he got it in the comments. **Result:** no problem.

Example: Joe include several passages from the book and some references from the web, in his discussion forum postings. He changed the font of the text to give contrast to his own commentary, and clearly cited where the passages came from. **Result:** no problem.

"Minor points" notation:

From time to time I use the point box as a communication vehicle in two specific ways, and I reserve the right to add minor points for this purpose:

- One point extra: I am tipping my hat to you for particularly fine work. That is, if you get 101 points on a 100 point programming assignment, I may be saying, "Hey, I noticed the five extra modules you wrote, and that you used TSL instead of sockets! Good job!"
- Two points extra: If you receive two extra points, I am acknowledging an *exceptional* contribution well beyond expectations, so 102 points on a 100 point assignment is something to feel really good about, and is a rare compliment.
- Grade of "1": used as a placeholder to let a student know that I have reviewed an assignment, and am waiting for further information or work as per correspondence. A "1" will *always* be resolved to a different grade.
- Grade of "2": a serious warning that you need to communicate with me about possible plagiarism or some other irregularity that is being investigated.

More Policies

Changes to Syllabus

This syllabus is subject to change as necessary during the quarter. If a change occurs, it will be addressed during class, posted in D2L, and sent via email.

Online Course Evaluations

Evaluations are a way for students to provide valuable feedback regarding their instructor and the course. Detailed feedback will enable the instructor to continuously tailor teaching methods and course content to meet the learning goals of the course and the academic needs of the students. They are a requirement of the course and are key to continue to provide you with the highest quality of teaching. The evaluations are anonymous; the instructor and administration do not track who entered what responses. A program is used to check if the student completed the evaluations, but the evaluation is completely separate from the student's identity. Since 100% participation is our goal, students are sent periodic reminders over three weeks. Students do not receive reminders once they complete the evaluation. Students complete the evaluation online in [CampusConnect](#).

Academic Policies

All students are required to manage their class schedules each term in accordance with the deadlines for enrolling and withdrawing as indicated in the [University Academic Calendar](#). Information on enrollment, withdrawal, grading and incompletes can be found at: cdm.depaul.edu/enrollment.

Students with Disabilities

Students who feel they may need an accommodation based on the impact of a disability should have CSD contact the instructor privately to discuss their specific needs. All discussions will remain confidential. To ensure that you receive the most appropriate accommodation based on your needs, contact the instructor as early as possible in the quarter (preferably within the first week of class), and make sure that you have contacted the Center for Students with Disabilities (CSD) at: csd@depaul.edu.

Lewis Center 1420, 25 East Jackson Blvd.

Phone number: (312)362-8002

Fax: (312)362-6544

TTY: (773)325.7296

Other Course Policies

Attendance: Students are expected to attend each class when there is an on-campus section (unless otherwise notified), or view the class online. Attendance will not be formally taken beyond the start of the quarter, except as required for on-campus class sections. Unless otherwise noted ALL the course material presented in the lectures is suitable for exams.

Class Discussion: Student participation in class discussions is expected, and this will take place in class for local students, and online for all students. But see the following:

Uncomfortable with lecture or discussion topics: This is a university course. DePaul is a major urban university with many students from all over the world. If you are uncomfortable with any of the lectures, and/or discussions, for religious, gender, social or any other reasons, then stop the video, leave the room and leave the discussion immediately. Make contact with the professor in a timely way to discuss your concerns and work towards a resolution. We cover challenging topics in this university environment, but we are also sensitive to our students' needs.

Attitude: A professional and academic attitude is expected throughout this course. Measurable examples of non-academic or unprofessional attitude include but are not limited to: talking to others when the instructor is speaking, mocking another's opinion, cell phones ringing, emailing, texting or using the internet whether on a phone or computer. If any issues arise a student may be asked to leave the classroom.

Cell phones / laptops in class: If you need to use your cell phone for any reason, or your laptop for any reason other than following the class slides, and taking notes, *leave the room*. You may quietly leave and re-enter as often as necessary unless I note otherwise. Your peers devote many hours out of their busy lives, and thousands of dollars, to come to class. They deserve a vibrant, focused, environment. If you have a special case, discuss it with the instructor ahead of time. NO TEXTING, EMAIL, FACEBOOK, etc. in the classroom.

Civil Discourse: DePaul University is a community that thrives on open discourse that challenges students, both intellectually and personally, to be [Socially Responsible Leaders](#). It is the expectation that all dialogue in this course is civil and respectful of the dignity of each student. Any instances of disrespect or hostility can jeopardize a student's ability to be successful in the course.