Comp 533: Distributed Systems

General Course Info

Term: Spring 2025
Department: COMP
Course Number: 533
Section Number: 001
Time: TR 12:30-1:45pm
In-Person Location: FB 007
Piazza: https://piazza.com/unc/spring2025/comp533
Zoom Lectures:
https://unc.zoom.us/j/98434671245?pwd=eDNJUHZlNUxiS1IpTUhOanhTYUVwZz09
TA Zoom Office Hours: *https://unc.zoom.us/j/98762626789*
Dewan Zoom Office Hours: *https://unc.zoom.us/j/97902404607*
Google Drive Folder:
https://drive.google.com/open?id=1kj6JoODwQJCGePblJBoTjnH2MK3gFtzf&usp=drive_fs

Short Description

Prerequisite, COMP 301; a grade of C or better is required.

Course Description

This course will cover abstractions and algorithms for the related fields of parallel and distributing computing (PDC). Java and C will be the vehicles for concretely illustrating them, but the underlying concepts will be language-independent. The lectures will be driven by a series of implementation-based assignments. The high-stake ones will be homework assignments implemented in Java, and the low-stakes ones will be in-class exercises involving no programming or programming in Java or C.

Abstractions: Distributed computing is essentially concurrent computing in different processes. Moreover, a process communicating with other processes typically has multiple activities that must be scheduled concurrently. Therefore, we will build on the Java concurrency abstractions you learned in 301 for creating cooperative

threads. We will look at analogous IPC (Inter Process Communication) abstractions for creating cooperative processes. These will include both Remote Procedure Call (RPC), which is easy to program, and Non-Blocking I/O (NIO), which is flexible and scalable. In addition, we will look at data parallel abstractions, (including single program, multiple data (SPMD) abstractions) provided for both concurrent and distributed programming.

Algorithms: We will cover fork-join parallelism, parallel reduction, distributed map-reduce, two-phase commit, and Paxos.

We will not cover topics that are the focus of other courses at UNC such as proofs of consensus algorithms (Comp 735), distributed file systems (Comp 790), "big data", data centers, and other distributed applications (Comp 790s), and parallelization/distribution of complex single-processor algorithms (Comp 633).

Target Audience

The target audience is students wishing to gain an in-dept ,understanding of fundamental concepts in parallel and distributed systems , with a focus on their design and implementation.

Goals and Key Learning Objectives

At the end of the course, you will have a basic understanding of how distributed software works, the potential uses of this software, the design and implementation space of distributed abstractions; the performance of alternative distributed abstractions; how to run experiments to measure performance, and implementation of key fault-tolerant distributed algorithms for achieving consensus. As a distributed program is also a concurrent program, you will also sharpen your understanding of threads and thread synchronization. Because of the emphasis on assignments that build on each other, you will gain practice with the use and implementation of advanced software engineering concepts such as layers, generic types, factories, and abstract factories.

Assignments

In homework, you will implement two sets of algorithms addressing HPC and broadcast, respectively. The first set will implement a Map-Reduce algorithm using sequential, concurrent and distributed programming, respectively. The second set will add distributed functionality to an existing non-distributed simulation using different IPC mechanisms and broadcast algorithms.

In addition, you will implement time-bound in-class assignments, designed to serve three purposes: 1) warm you up for related homework assignments, 2) give you practice with time-bound work, 3) cover concepts not addressed by homework assignments.

Programming Environment

We will use the Eclipse programming environment even though many of you are unfamiliar with it. The reason is that unlike other environments such as IntelliJ and Visual (Code) Studio it is an open-source environment – perhaps one of the most striking open-source project. This means plugins exist for this environment that do not exist for the others. In particular, we will use Java style checks. **You are free to use some other programming environment or even the command-line and a text editor, but are likely to be less productive if you do so.** We will answer programming environment questions only about Eclipse.

Instructor Info

Name:	Prasun Dewan
Office:	FB150
Email:	dewan@cs.unc.edu (Use Piazza private messages)
Phone:	9195906123
Web:	http://www.cs.unc.edu/~dewan
Office Hours:	See pinned Piazza post

What's in a Name: How to Address Me

The following ways of addressing me are fine and I am used to them:

Prasun

Dr/Prof. Dewan

And in case you are worried about mispronouncing my first name, it rhymes with Bassoon.

I wince if someone addressed me simply as "Dewan", which some students have done in recent semesters.

Graduate Teaching Assistant

Name: Gagou, Sammy
Email: sgagou@unc.edu (Use Piazza private messages)

Undergraduate Teaching Assistant

Name: Azad, Rasheeq
Email: razad@unc.edu
Office Hours: See pinned Piazza Post

Lecture Mode

During the lockdown, I experimented with using Zoom chats to for Q/A. In comparison to my traditional Q/A interaction mode, it increased interactivity manyfold.

Because of the increased interactivity, I have retained this structure even when the lockdown ended and in-person classes started.

This mode will be supported in this course**. This means you must bring a laptop to class and log in to the Zoom session with your speaker and mic muted so that you can use Zoom chat for interaction.**

You will be allowed to interact remotely as long as you do so rarely, in special circumstances.. The university and most students in my classes feel students learn much more when they attend in-person. Therefore if we find that a set of students are noticeably using remote attendance regularly, screen sharing and audio ill be muted to incentive greater learning.

All lecture zoom sessions will be recorded mainly to get textual transcripts of the interaction so we can reflect on and improve the lectures.

Office Hour Mode

Zoom will also be used for all office hours visits. You will need to let us know at least one hour before an office hour if you plan to visit. A Piazza message explains exactly how. You should visit the TA for assignment problems and the professor for conceptual problems. The TA may not be conversant with the concepts taught in class and the professor may not be familiar with subtle hard to catch mistakes in installing and using software and implementing the assignments.

All office hour zoom sessions will be recorded by default mainly to get textual transcripts of the interaction so we can reflect on and improve the visits. These transcripts will be anonymized before we look at them.

If you feel even slightly uncomfortable with the idea of a Zoom recording of an office hour visit, please let us know, and we will understand and disable recording for you.

Asynchronous Help

We will use Piazza for questions that do not require live interaction with a TA or professor. Usually, such interactions should occur after you have tried to use Piazza to get help. If you need help with code, make your post private. Otherwise, make your post public so others can benefit from the discussion and can possibly help. If you have a personal situation, then of course, make it private, but tag it as personal_situation.

AI Help

AI help is not allowed in in-class assignments.

In Fall 2024, in comp 524, we allowed the use of AI tools to solve homework programming problems, and asked students to submit logs of their interaction with the tools. Only one student submitted a log for one assignment, and it demonstrated a more or less mindless use of the tool, in which the student submitted the assignment requirements, received a solution, and then asked the AI tool to try and fix every error reported by our testcases. Perhaps the same or different student reported in their end-course evaluations that they used this approach, and blamed the instructor for allowing its use.

However, we do not want to prevent others from using the tool in more constructive ways. The jury is out on whether the use of AI in coding/debugging assignments, overall, promotes or hinders learning. See for example: From "Ban It Till We Understand It" to "Resistance is Futile": We do not pretend to know the answer. We will leave it you on how you want to use it to complete assignments. We do recommend you do not use the mindless use mentioned above. However, as in 524, we require you to document each use of an AI engine for coding a 533 homework assignment a Piazza post. This post should document the AI system used (e.g. Gemini/GPT/CoPilot). It should have the tag(s) of the assignment(s) for which help was sought, and also the special tag: AI_Used. If you managed to suppress code in interactions with the AI tool, you are free to make the post public; otherwise make it private.

Not acknowledging the use of AI tools will be considered a violation of the honor code, as mentioned below. As you can imagine, there is considerable

ongoing work on detecting AI-based plagiarism. My group plans to expand the set of such tools as we do research in this area.

Honor Code

You are encouraged to discuss the assignments with fellow students and help them debug programs but required to write/code the solutions/programs individually. **Also, you cannot use solutions from previous offerings of the course.** You cannot make incorrect diary reports or plagiarize the diary text. You cannot simply copy quiz answers from another student in this or a previous offering though you are encouraged to discuss them with others. Making your assignment code public, through say GitHub, is essentially sharing your code and thus not allowed. You cannot use AI to solve programming problems without documenting such use in a Piazza post, as outlined above.

Not following these rules is a violation of the honor code policy

Key Dates

Midterm: Class time, Thursday Mar 6th (the week before Spring Break)
Final: 4pm, Saturday May 3th
Wellness Days: Thu, April 17

Attendance Policies

Students are required to attend each class in person unless there are extenuating circumstances (travel, sickness, exam in some other class, …). If you cannot attend in person, let the instructor know through a private post so the instructor can share the Zoom screen and enable audio. The post should be tagged as remote_attendance.

If you cannot attend in person or remotely, you should access the class material posted for missed classes, and contact classmates to become aware of the announcements that were made.

You are required to log in to the Zoom session to answer chat-based questions.

Textbooks and Resources

There is no textbook covering the topics of this course.
All resources are available from subdirectories of the shared google drive directory:

Grades

A grade will be assigned based on performance on homework programming assignments, quizzes, class participation, and exams. Here is the breakdown:

Regular Credit

Midterm	25%
Final	30%
Assignment Regular Credit	40%
Class Participation (My QA diary)	5%

Extra Credit

Assignment Extra Credit	TBA

Assignments are of different weights determined by their max scores. For instance, an assignment worth 10 points (regular credit) has a weight $1/10^{th}$ of an assignment worth 100 points (regular credit). Your total assignment grade is the sum of all of your individual assignment scores divided by the sum of all assignment max scores. By doing extra credit, you can get more than 100% on the assignments.

Fudge Factor

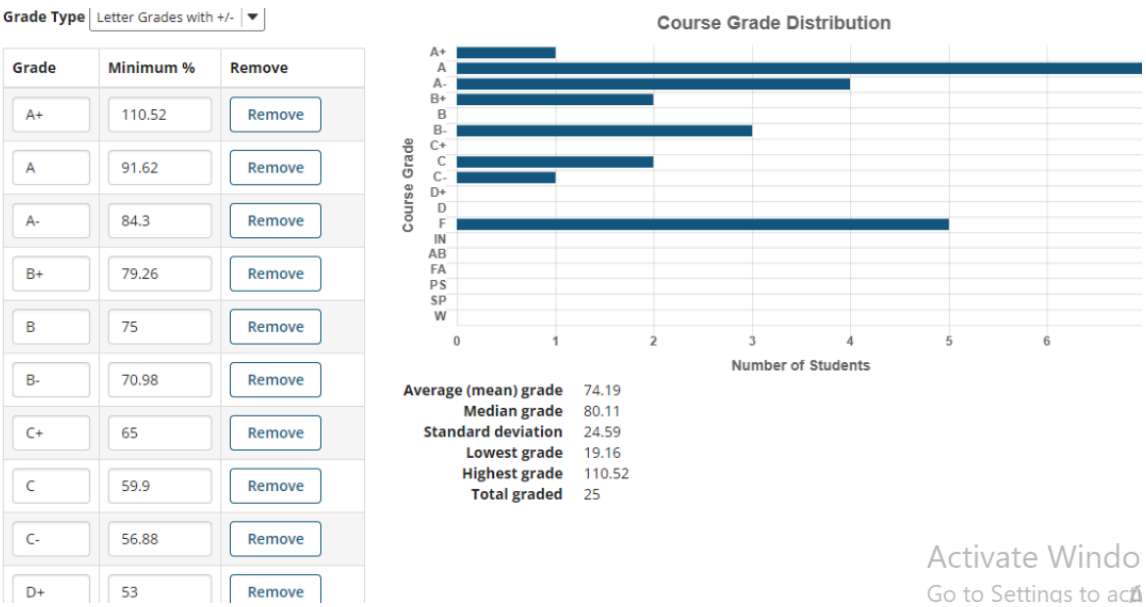
I reserve the right to apply a 5% fudge factor to give consideration to other factors such as extraordinary quality of class participation.

Mapping between Grades and Percentages

There is no fixed mapping between the overall percentage and final grade. A median score on exams and quizzes and 90% score on the regular credit part of assignments guarantees you at least a B. Total score greater than 100% is a guaranteed A. I tend to look for gaps to cluster students.

Here is the distribution of grades in Spring 2022. Just as past performance in stocks is not a predictor of their future performance these numbers are not indicative of how grades will be assigned this year because of differences in the nature of exams and the extent to which students drop.

:



Class Participation

Class participation has to do with questions posed **by the instructor** that **you** tried to answer to the whole class.

Often I echo the questions asked by the students, and these questions do get added as instructor questions.

You do not get class participation credit for questions you asked and I or someone else answered

You will maintain a diary listing responses to my questions. A Piazza message explains exactly how.

Responses do not have to be correct but should show some attempt to think about the answer. Every response will get the same credit, there will be no judgment regarding its correctness. **A response saying you have no idea on how to answer the question is not a valid response and should not be put in the diary. You should also not put in the diary a response made after the zoom discussion on the question closes**. Zoom chat records will be made available to help you compose the diary entry

Each diary will be graded based on the number of entries in it. An automatic tool will extract this number. Instructors may check the validity of the contents. It is of course an honor court violation to deliberately add spurious entries. Spurious entries include fabricating answers not articulated by you to the whole class .

The max score will be the largest number of entries by any student. Thus, if you add 3 entries of My QA diary, and the maximum number of entries added for My QA is 6, then you get 50%. You should fill the diary for a class by the end of the day the class was held, when things are still fresh in your head. Otherwise, you may or may not get credit for it - depending on when we grade the diaries. At this point we do not know the frequency of grading.

Assignment Grading

Both source code and runtime behavior will be graded using automatic tests, whose mistakes will be corrected by manual grading,

You are required to submit each assignment to the Gradescope server. Late penalty is computed automatically by the Gradescope tests.

You can also run our tests locally on your computer to easily validate your solution before submitting it to Gradescope. These tests log your interaction anonymously, again to improve our future offerings. **If you feel uncomfortable with this, please do not run them locally – we will allow you to submit to Gradescope an arbitrary number of times.**

Late Penalty

Assignments are due at 11:55 pm on each specified due date. Homework assignments will be penalized 10% up to one week late and 30% for one to two weeks late. They will not be accepted more than two weeks later. **No assignment can be submitted after the last day of class.**

Early Submission Extra Credit

If you submit by the early submission date of the assignment, you will get 5% extra credit, that is, your score will be multiplied by 1.05. Aiming for early submission extra credit will ensure you do not incur late penalty. On the other hand, submitting later does allow for more extra credit, and many students have chosen to do so.

Exam Nature

It is unlikely you will be asked multiple-choice questions. Instead, exam questions will likely mirror the questions and discussion in class. They are closed book and no access to the computer is allowed. Practice exams will be in the shared google drive folder. In-class exercises and Q/A should also

Workload

The time you spend on the course depends on many factors such as how difficult to track your specific programming mistakes are. The average student should spend 9 hours outside lectures on regular credit portions of the homework assignments.

Many courses have a smaller workload. The amount of code you will write will be small but dense. The assignments will come at regular intervals so you will need to find time throughout the semester to work on assignments.

Accessibility Resources and Service (ARS)

The University of North Carolina at Chapel Hill facilitates the implementation of reasonable accommodations, including resources and services, for students with disabilities, chronic medical conditions, a temporary disability or pregnancy complications resulting in barriers to fully accessing University courses, programs and activities.

Accommodations are determined through the Office of Accessibility Resources and Service (ARS) for individuals with documented qualifying disabilities in accordance with applicable state and federal laws. See the ARS Website for contact information: https://ars.unc.edu or email ars@unc.edu.

Course Schedule

The current plan is to cover the following topics in the order given below:

1. Course Information
2. Relationship between Concurrency and Distribution.
3. Introduction to Java Threads, and Deadlocks
4. In-Class Exercise on Java Concurrency and Fork-Join Problem
5. Java Monitors.
6. Introduction to Remote Procedure Invocation and Remote Thread Creation
7. In-Class Command-Line Exercise on Java Monitors and Remote Threads
8. Remote Method Invocation Basics
9. C-based Procedural Concurrency Abstractions: C Pthreads
10. In-class exercise on Pthreads
11. Reduction
12. C-based OpenMP support for Reduction
13. In-class exercise on OpenMP
14. C-based MPI support for Single Program Multiple Data Model (SPMD)
15. In-class exercise on MPI and SPMD
16. Map Reduce
17. High Performance Computing vs Cooperation
18. Blocking Messages Passing, Java Sockets
19. Non-Blocking Message Passing, Java NIO
20. Consensus, Single Reader Multiple Reader, Two-Phase Commit

21. Paxos: Distributed Strong Consensus

Disclaimer

The professor reserves the right to make changes to the syllabus, including project due dates and test dates. These changes will be announced as early as possible.