CSE 4/586 Introduction to Distributed Systems Fall 2018

Course Description

A distributed system is one in which components located at networked computers and devices communicate and coordinate their actions by message passing for sharing resources and for distributing computational workload. This course will address some of the fundamental challenges in the design, implementation and deployment of large scale distributed systems including connection establishment, event handling, inter-process communication, storage management, static and dynamic component configuration, concurrency and synchronization. It will also cover issues related to distributed objects such as mobility, scalability, security, naming and location. Reliability is a core issue that will be covered across all the topics discussed in the course. Possible solutions will be analyzed at various levels of granularity using objects, processes, services, components and frameworks. Distributed fault tolerance, consensus mechanisms, distributed ledger technology supporting the emerging blockchain technology will also be discussed.
On completion of this course, a student will be able to design and implement a reliable distributed system, and will be able to analyze a distributed system for its architecture, algorithms, protocols and services. Students will have good understanding and working knowledge of reliable distributed systems.

Course Information

Website:	http://www.cse.buffalo.edu/~bina/cse486/fall2018
Instructor:	Bina Ramamurthy (bina@buffalo.edu)
Lecture Time:	MW: 5.00-6.20PM
Lecture Location:	Cooke 121
Office Hours:	MW: 4:4.50 PM

Recitations:

11768	M	10:00 AM - 10:50 AM	Norton 210
11769	F	3:00 PM - 3:50 PM	Norton 213

Textbook and other material

Text book: Distributed Systems: Concepts and Design, by G. Coulouris, J. Dollimore and T. Kingberg, Fourth Edition, Addison-Wesley, 2005.
Besides the main textbook we will use a number of other material print and Internet publications, the references for which will be given to you at appropriate time during the semester.

J.A. Stankovic, "Distributed Computing," in Readings *in Distributed Computing* Systems,T.L. Casavant and M. Singhal, Eds., pp. 6--30. IEEE Computer Society Press, Los Alamitos, CA., 1994. Read here.

Pre-requisites

You will need to have taken CSE505/CSE305 or an equivalent course that gives you a background in programming language internals; you should have a good foundation in problem solving, design representation, and object-oriented design methodology and application and design and development in Java/C#/python. You should also be familiar with object-oriented modeling, modern code design and debugging practices.

Grading Distribution

Final letter grades will be based on the (combined) overall percentage of all the items listed below. A (95 --), A- (90 -- 94), B+ (85 -- 89), B (80 -- 84), B- (75 -- 79), C+ (70 -- 74), C (65 -- 69), C- (60 -- 64), D+ (55 -- 59), D (50 -- 54), F (less than 50). This policy is subject to change. If needed, the individual components and the overall grades will be appropriately curved.

Grades will consist of the following components:

Component (Quantity)	Points
Attendance including pop quiz	50
Project (3)	100X3
Exam 1 (1)	100
Exam 2 (1)	200

Projects

The due date for each project will be announced when it is assigned. All the source code, documentation, makefile, data files, and README files are to be submitted on-line. The details of how to submit given along with your first project. You will have to follow the rules for the other projects too.

I reserve the right to change the project specifications at any point before the due date to address problems that may arise during the course of the project. If your design is modular the changes will not be difficult to implement. A detailed grading guideline will be given to you along with the project specification. Use this as a guide for your design and implementation. It is absolutely necessary to keep up with the programming projects in the class. There will be a 25 point deduction for each day the project is late after the due date.

Develop your code using the ***Incremental Development technique.*** Do not try to sit down and code the entire assignment in one sitting. Instead, take one section at a time, implement, test it, back up the code, and move on to the next section. You will turn in each lab before 11:59 PM on the due date via the departmental ***submit*** command. You must also include appropriate testing programs to show the validity of your solution. In addition, you must include external documentation discussing the "how's and why's" of your design and implementation. You will be required to demonstrate your lab to your TA. The TA will also run test examples against your code to check your solution's overall correctness. The TA will provide a demo schedule. It is your responsibility to demo your project, or you will receive a zero for that portion of the grade.

When your grade is assigned for the lab, the TA will indicate critical areas that must be fixed in order to solve the next assignment. In general, solutions are not provided for the lab. It is not that we do not have them, or am unwilling to distribute them but it's that there is no one answer to any project solution.

Exams

There will be a midterm that will be administered and graded before the resign date. Exam1 will cover all lecture and reading assignments before the exam, as well as concepts from the lab assignments. The Exam2 is a comprehensive exam, covering all lecture, lab, and homework areas. All exams are closed book, closed notes, and closed neighbor. We do not give make up exams for any reason. If you miss an exam, you will receive a zero for that portion of the grade.

Attendance Policy

You are responsible for the contents of all lectures and recitations (your assigned section). If you know that you are going to miss a lecture or a recitation, have a reliable friend take notes for you. Of course, there is no excuse for missing due dates or exam days. Attendance will be taken and will determine the attendance grade component for the computation of your final grade.

During lectures, we will be covering material from the textbook. We will also work out several of the problems from the text. Lecture will also consist of the exploration of several real world problems not covered in the book. Lecture dynamics will depend very much on students' participation and students are strongly encouraged to ask questions related to the material covered in the class. You will be given a reading assignment at the end of each lecture for the next class.

Recitations are designed to review difficult concepts in the class and to spend additional time discussing the lab work required for the course. The recitation is your time to communicate with your TA about the course. Use the opportunity to the fullest.

Office Hour Policy

If you can't meet during these hours, you will have to communicate with us via Email. Office hours are intended to resolve questions about the material that could not be answered in lecture or recitation. Come to office hours prepared! Office hours are **NOT** for the following: to repeat missed lecture material, to repeat missed recitation material or to have the instructor or TA solve an assigned problem for you. During office hours, we will **NOT** write or debug your code for you! Instead, we will direct you as to where to concentrate your debugging efforts.

Grading Policy

All assignments will be graded and returned in a timely manner. When an assignment is returned, you will have a period of one week to contest any portion of the grade. The TA who graded your assignment will be the first person to resolve a grading conflict. If the conflict cannot be resolved, the instructor will mediate the dispute. The judgment of the instructor will be final in all such cases. When contesting a grade, you must be able to demonstrate how your particular solution is correct. Also, when contesting a grade, the instructor or TA reserves the right to re-evaluate the entire lab or exam, not just the portion in dispute.

Incomplete Policy

We only grant incompletes in this course under the direst of circumstances. By definition, an incomplete is warranted if the student is capable of completing the course satisfactorily, but some traumatic event has interfered with their capability to finish within the timeframe of the semester. Incompletes are not designed as stalling tactic to defer a poor performance in a class.

Academic Integrity Policy

UB's definition of Academic Integrity in part is, "Students are responsible for the honest completion and representation of their work". It is required as part of this course that you read and understand the departmental academic integrity policy located at the following URL:
https://engineering.buffalo.edu/computer-science-engineering/information-for-faculty-and-staff/academic-integrity.html

There is a very fine line separating conversation pertaining to concepts and academic dishonesty. You are allowed to converse about general concepts, but in no way are you allowed to share code or have one person do the work for others. You must abide by the UB and Departmental Academic Integrity policy at all times. **NOTE:** Remember that items taken from the Internet are also covered by the academic integrity policy! If you are unsure if a particular action violates the academic integrity policy, assume that it does until you receive clarification from the instructor.

Web Site

The CSE486 website should be checked frequently for important news. Course assignments, slides, grade reporting, and general hints and tips will be posted on the website.

Students with Accessibility Issues

If you have special needs due to a disability, you must be registered with the Office of Student Accessibility. If you are registered with them please let your instructors know about this so that they can make special arrangements for you.