University at Buffalo Department of Computer Science and Engineering

Syllabus

CSE 486/586: Distributed Systems

Ethan Blanton

Spring 2023

Revision: 2023-02-02

All students are expected to read and understand this syllabus. Failure to adhere to the policies in this syllabus may have consequences, including a negative impact on student grades, failure in the course, or administrative action against the student. *It is your responsibility to ask questions if anything in this document is unclear to you.*

This course is in-person and real-time. Students are expected to attend their assigned lecture unless they have a University-approved reason to miss.

Instructor

Ethan Blanton
eblanton@buffalo.edu

Office Hours (subject to change; consult the course web site):

Monday	16:00–16:50	Davis 334
Wednesday	13:00–13:50	Davis 334

Teaching Assistants

TA names and their office hours can be found on the course Piazza instance.

Sections

Section	Course ID	Lecture Time	Location
CSE 486 LEC-000	10518	MWF 11:00–11:50	Knox 104
CSE 586 LEC-000	10514	MWF 11:00–11:50	Knox 104

1 Course Web Site

https://cse.buffalo.edu/~eblanton/course/cse586/

Locations, times, information regarding instructor and TA office hours, assignment deadlines, and other information can be found on the course web site.

2 Lectures

Lectures will be delivered on campus in their assigned locations. Students are expected to attend their assigned lecture unless they have a University-approved reason to miss. Attendance will not be regularly taken, but may be taken at any time at the instructor's discretion (and may affect grade outcomes per the quiz policy in Section 9). Lecture recordings will be provided after the fact for review, or for asynchronous viewing in the event that a student must miss lecture for a University-approved absence, on a best-effort basis.

3 Course Description

CSE 486/586 is a 3 credit course.

> CSE 486: Addresses some of the fundamental challenges in the design, implementation and deployment of large-scale distributed systems. Concepts covered include concurrency, synchronization, connection establishment, event handling, inter process communication, storage management, and service registration, discovery, and lookup. Also covers issues related to distributed objects such as life cycle management, mobility, security, naming, location, evolution, and autonomy. Analyses *[sic]* and implements possible solutions using objects, processes, services, components and frameworks. Offered irregularly.
>
> CSE 586: This course addresses some of the fundamental challenges in the design, implementation and deployment of large scale distributed systems including connection establishment, event handling, interprocess communication, storage management, static and dynamic component configuration, concurrency and synchronization. It will also cover issues related to distributed objects such as mobility, security, naming, location, evolution, autonomy and negotiations. Possible solutions will be analyzed and expressed using objects, processes, services, components and frameworks at various levels of granularity. This course focuses on practical solutions using the latest server-side and middleware technology.

4 Course Materials

The required text for this course is *Distributed Computing: Principles, Algorithms, and Systems* by Ajay D. Kshemkalyani and Mukesh Singhal, published by Cambridge University Press. Its ISBN is 978-0-521-18984-2.

Lecture slides will be provided electronically throughout the course of the semester.

Various readings from external sources may be assigned, in which case they will be provided or will be available through University resources (*e.g.*, the library or periodical subscriptions).

References to materials not required or assigned may be made, and students are encouraged to follow up on these references, but this will not be required for successful completion of the course.

5 Communication

All electronic communication from students to course staff regarding this course must occur in one of two ways:

- Messages on the course Piazza instance, or

- Email using your *official UB email account*.

For topics of a sensitive nature, please email the course instructor directly from your University-supplied email address. Emails from non-University addresses will be disregarded due to privacy concerns and FERPA regulation. For all other contacts, please do **NOT** email the course instructor directly; instead, make a private or public post to Piazza, as appropriate. Private posts of non-sensitive nature should be sent to *all course staff*. This will ensure the most timely possible response.

Students are expected to monitor the course Piazza instance on a daily basis, checking it on every day that the University is open, as important course announcements will be posted to Piazza. Schedule changes, assignment handouts, homeworks, required readings, and other materials may be posted to Piazza, and it is the student's responsibility to keep track of these things. Failure to read Piazza messages will not be accepted as an excuse for missed projects, exams, or other course requirements.

Students will be added to the Piazza instance by the course instructor. If you are not, please contact the instructor by email to rectify this.

6 Prerequisites

All students registered for CSE 486/586 are expected to have significant programming experience (equivalent to a UB undergraduate of upper class standing), a basic understanding of computer networks and operating systems, and an understanding of concurrency issues in multi-threaded programs. Students of both CSE 486 and CSE 586 should use the official CSE 486 prerequisites as a guideline for understanding their expected background.

Students have experience with or be able to pick up the Go programming language (https://golang.org/) in order to complete course projects. A detailed treatment of Go will not be provided as part of the course material.

CSE 486: CSE 220 and CSE 250; Approved Computer Science, Computer Engineering, Bioinformatics/CS Majors Only. Students must complete a mandatory advisement session with their faculty advisor.

7 Course Requirements

The following items are required of every student, and failure to complete them may affect student grades as described in Section 9, *Grading Policy*, below.

7.1 Lecture and Presentations

Students must attend every lecture. Students must watch any pre-recorded videos containing course content that may be released from time to time.

It will be assumed that students are familiar with all material presented in class, and any material presented in lecture or course videos may appear on any test, quiz, homework assignment, or other evaluation. *Attendance and attention to lecture and course videos are critical to success in this course.*

7.2 Assignments

Several programming assignments will be required of all students. Programming assignments are intended to improve student understanding of the course material as well as demonstrate student mastery of certain core concepts.

Programming assignments, unless otherwise explicitly specified, are *individual activities*, and collaboration between students to complete any such assignment is a violation of the course academic integrity policy.

Written homework may be assigned, and students are expected to complete any such assignments in a timely fashion, although they will not be graded. These written assignments are intended to allow students to self-evaluate their level of preparedness and mastery of the course material, and students are encouraged to seek assistance from the instructor, teaching assistants, or each other in understanding and completing written assignments.

7.3 Readings

Readings from the textbook and supplementary materials will be assigned regularly, and students are expected to complete these readings in a timely fashion (no later than one week after they are assigned). Readings are selected to improve student understanding of the course material and/or present auxiliary material that the instructor believes is relevant and important. Material from readings may appear directly or indirectly in assignments, on quizzes, or on exams.

7.4 Tests and Quizzes

A quiz evaluating students' understanding of the University and Department academic integrity policies must be completed by all students, and all students must achieve 100% accuracy on this quiz.

Quizzes may be introduced at any time by the instructor, covering any material previously covered in lectures, readings, or written homework assignments. These quizzes may or may not be announced in advance. (In particular, "pop quizzes" may be utilized to evaluate student attendance, engagement, and present understanding of course material.)

There will be one midterm and one final exam. The midterm exam will cover all material presented in the course to date, including (but not limited to): lectures, programming assignments, written homework assignments, and assigned readings. The (cumulative) final exam will cover all material covered in the course for the duration of the semester, including (but not limited to): lectures, programming assignments, written homework assignments, and assigned readings.

7.5 Submission Policy

Programming assignments will be assigned with a deadline. All assignments are to be submitted by this deadline. In the event of any ambiguity in the deadline, times are assumed to be in the *current local time zone of the University*. Penalties for missing this deadline are as follows.

- Projects submitted before the deadline will incur no penalty.
- Projects submitted after the deadline, but within 24 hours of the deadline (excluding Saturday, Sunday, and University holidays) will incur a 20% penalty.
- Projects submitted more than 24 hours after the deadline as described above will not be accepted and will receive no credit.

Neither the instructor nor the teaching assistants will provide assistance for programming assignments after the assigned deadline.

7.6 Programming Assignment Re-grading Policy

If you believe that a programming assignment has been graded incorrectly, you may submit it for re-grading. A request for a re-grade must be submitted within one calendar week of receiving the grade for a project, and must include:

- The original score achieved on the assignment
- A description of the specific error in grading that is being contested
- Relevant code demonstrating the submitted code's correctness or the grading script's incorrectness, if available

Re-grading of programming assignments is intended *only* to address errors in grading. No grades will be improved for any other reason, although they may be reduced; in particular, note that *your grade on any part of the assignment, not just the portion being re-graded, may be reduced* if re-grading discovers additional errors. This includes automated evaluations that passed because they did not trigger bugs that were discovered in manual evaluation for the re-grade, or bugs that show up only intermittently that happen to be encountered on the re-grade.

7.7 Exam and Quiz Re-grading Policy

If you believe that an exam or quiz has been graded incorrectly, you may submit it for re-grading. A request for a re-grade must be submitted within *one calendar week* of the exam or quiz being returned to you, must be submitted in writing (email is acceptable) to the instructor, and must include:

- The original, unmodified, exam or quiz answer
- A clear statement of the error

Re-grading of exams and quizzes is intended *only* to address errors in grading. No grades will be improved for any other reason, although they may be reduced if *errors are found in any portion of the assignment, not just the portion being re-graded.* Using re-grading as a bargaining tool to increase your score is likely to result in a lower grade, as the exam will be scrutinized in detail for errors that may have been missed the first time.

7.8 Make-up Policy

No deadline extensions or make-up work will be permitted except for approved University absences. Please see the University attendance policy for more information.

No make-up exams will be given whatsoever except for **documented extreme circumstances**. *24 hours of advance notice via e-mail must be provided if at all possible* before missing an exam session. If advance notice is not possible, documentation supporting this must be provided. Absence from an exam session due to illness **must** be supported by a note from a physician specifying that the student was too ill and/or contagious to attend on the exam date.

You are responsible for remembering and attending exam sessions. Please use extra assistance to remind yourself if necessary.

7.9 Contingencies Related to University-wide Disruptions

In the event that the execution of this course is disrupted due to a situation outside of our immediate control (such as the lockdowns for public health at the outset of the COVID-19 pandemic), course staff will endeavor to provide continuity as best as possible. Unless circumstances otherwise dictate, lectures will move to online, live broadcast via a service such as YouTube Live; labs and office hours will move to a service such as Zoom; and the established course schedule will be maintained as far as is practical. A new syllabus will be issued only if required by the University, or if there are significant changes to the course material or grading policy.

7.10 Accommodations for Student Health

If a student is required to miss class for health-related reasons that could not have been reasonably foreseen (*e.g.*, quarantine due to inadvertent exposure to COVID-19 or contraction of the virus, or contraction of another communicable illness), reasonable accommodations will be made. Students who are quarantined but otherwise capable of continuing with course work are expected to view lectures online, attend virtual office hours, turn in assignments, and keep up with the course as much as is practical. The course is structured to allow this. Students who are too ill to pursue their course work will be accommodated to the extent practical.

8 Course Schedule

The course schedule, including exam dates and times, is provided here for convenience. Note that course staff do not schedule some of the following items, including final exam time and location. *You are responsible for verifying your final exam time and location on HUB.* Inclement weather, local emergencies, unsafe building and/or campus environments, or other circumstances may cause the University to change this schedule. Course progress and pedagogical concerns may cause rescheduling of lectures, exams, and activities, or changes to required readings. You will be notified via Piazza or UB email of changes within the control of course staff. Course staff will attempt to keep you appraised of changes outside of staff control, but you are responsible for monitoring University communications to this effect.

Note in particular that the schedule of topics is being rearranged and strengthened during this iteration of CSE 486/586, and this may have some impact on the due dates of programming assignments. Every effort will be made to adhere to the schedule laid out here, but changes may be warranted as the semester progresses.

You are responsible for monitoring any changes to this schedule, according to communications from course staff or the University. Failure to be aware of schedule changes is not sufficient reason for extended deadlines, make-up exams, or other accommodations.

Date	**Description**
2023-01-30	First Day of Class
2023-02-24	Programming Assignment 1 due
2023-03-17	Programming Assignment 2 part 1 due
2023-03-20–	
2023-03-25	*Spring Break*
2023-03-29	Midterm Exam
2023-04-21	Programming Assignment 2 part 2 due
2023-05-12	Programming Assignment 3 due

2023-05-12	Last Day of Class
2021-05-15	Final Examination

The following schedule of topics is a draft, and is subject to change.

Week	**Topic**
2023-01-30	Introduction / Internetworking / Go
2023-02-06	Distributed Systems Model and Failures
2023-02-13	Time and Global States
2023-02-20	Naming and Distributed Hash Tables
2023-02-27	Multicast and Gossip Protocols
2023-03-06	Elections and Consensus
2023-03-13	Consensus
2023-03-20	Spring Break
2023-03-27	Midterm Exam Weds (March 29)
2023-04-03	Byzantine Failures, Mutual Exclusion
2021-04-10	Raft and Quorum Protocols
2021-04-17	Transactions and Consistency
2021-04-24	Consistency
2021-05-01	Security Considerations
2021-05-08	Distributed Applications

9 Grading Policy

No "I" (Incomplete) grades will be given for this course except for **documented extreme circumstances** or situations required by University policy. *Failure to complete work on time does not constitute an extreme circumstance.*

Grades will not be changed at the end of the semester for any reason other than a documented error in grading according to the policies outlined in Section 7.6 and Section 7.7. No grade negotiation will be permitted. In particular, no grades will be changed to preserve scholarships, fellowships, University positions, immigration status, internship or job offers, or any other outside factor. Grades reflect student performance and mastery of course material.

The credit breakdown for the course will be as follows:

Course Requirement	**Course Grade**	**Percent**
Less than 100% on Academic Integrity quiz	F	*N/A*
Programming Assignment 1		10%
Programming Assignment 2 pt. 1		10%
Programming Assignment 2 pt. 2		15%
Programming Assignment 3		15%
Midterm Exam		20%
Final Exam		30%

At the instructor's discretion, up to 15% of the course grade may be removed from the percentage allotted to the final and midterm (in any proportion) and allocated to quizzes, including pop quizzes, without prior notice. *Attendance in lecture is mandatory.*

Failure to submit the Academic Integrity quiz with complete correctness (100% credit) will result in failure in the course. You may submit this assignment as many times as required to achieve complete

correctness.

Final grades will be assigned from the above percentages as follows, although individual component scores may be adjusted or a curve of the instructor's choice may be applied if the instructor deems it warranted. Lower percentages are inclusive, upper percentages (excepting 100%) are not; that is, a 90.0% would be an A-, not a B+.

A	95-+ %	C+	77-80%
A-	90-95%	C	73-77 %
B+	87-90%	C-	70-73 %
B	83-87%	D+	67-70%
B-	80-83%	D	63-67%
		F	0-63%

10 Behavioral Expectations

Students are expected to behave in a way that is respectful to their fellow students and the course staff, and uphold the CSE values. In addition, the University at Buffalo has a list of behavioral expectations that includes:

- Attending lectures and paying attention. Students should not ask an instructor in class to go over material they missed by skipping a class or not concentrating. If lectures are watched asynchronously, students should watch them on the day they are given or as close as reasonably possible.
- Not coming to class late or leaving early. If a student has to enter a class late, he or she should do so quietly and should not disrupt the class by walking between the class and the instructor. Students should not leave class unless it is an absolute necessity.
- Not talking with other classmates while the instructor or another student is speaking. If a student has a question or comment, he or she should raise a hand, rather than starting a conversation about it with a neighbor.
- Showing respect and concern for others by not monopolizing class discussion. Students must allow others time to give their input and ask questions. Students should not stray from the topic of class discussion.
- Not eating and drinking during class time.
- Turning off electronic devices including cell phones, pagers, and watch beepers.
- Avoiding audible and visible signs of restlessness. These are both rude and disruptive to the rest of the class.
- Focusing on class material during class time. Sleeping, talking to others, doing work for another class, reading the newspaper, checking email, watching videos, and exploring the Internet are discouraged.
- Not packing bookbags or backpacks to leave until the instructor has dismissed class.

In addition to this list, students should minimize their use of laptop computers, tablets, and smart devices. Note-taking and the use of assistive technologies are appropriate, but web browsing, watching videos, chat programs, *etc.*, are inappropriate behavior for the classroom.

11 Academic Integrity

Students will abide by the CSE Academic Integrity Policy, the University Academic Integrity Policy, and the Undergraduate or Graduate amendments thereof, as appropriate.

The Academic Integrity policy for this course, and my other courses, can be found on my web site, under Policies. You should read it for additional information and clarifications not found here.

All resources used in completing assignments for this class *must be given appropriate attribution*, and the *only resources allowed for the completion of programming assignments, quizzes, or exams without specific permission* are as follows.

- Lecture material from this course
- Required or recommended readings from lecture material
- The required course textbook *Distributed Computing* by Kshemkalyani and Singhal
- *Operating Systems: Three Easy Pieces* by Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau.
- *The Computer Science Book* by Tom Johnson
- The Go programming language documentation on `https://golang.org/`

In particular, Stack Exchange, code from other students in the course or students who have completed this course or related courses at other universities in previous semesters, GitHub repositories, code or algorithms from other web sites or books, and other resources are not allowed without explicit permission from the instructor.

If there is any question about whether a resource is acceptable for use in completing a course assignment, students are encouraged to ask the instructor or a TA *before* making use of it. Asking about a resource is **not** a violation of academic integrity, even if the resource is not allowed for the course.

Quizzes and exams may have further restrictions on allowable resources; for example, a student's own work from previous assignments may not be an allowable resource on an exam.

Violation of these policies will result in a failing grade for the course and referral upward for additional sanctions according to University policy.

Sharing of course materials *after* completing this course is also a violation of the academic integrity policy for this course.

11.1 Sharing Course Materials After Completion

Materials used in this course, including project handouts *and your own project implementations* remain a part of this course after you complete it. Sharing those materials after completing this course may still constitute an academic integrity violation. Academic integrity proceedings may be started **even after you have passed this course**.

In particular, be aware that if you post your course project materials in a public place after completing the course, **you may be subject to academic integrity proceedings**. This can result in a **retroactive failure** in this course.

11.2 Amnesty for Violations of Academic Integrity

A student who has committed a violation of this academic integrity policy may receive limited amnesty for the violation by *notifying me, in writing,* of the violation **before I have begun to assess the violating assignment**. This notification must include the student's name, person number, UBITname, and

state the assignment in question and the nature of the violation. Upon submitting such a statement, the student will receive no credit for the violating assignment, but *no further sanctions will be taken, and the violation will not be reported.* Once I have begun assessing the assignment in question, no such statements will be permitted. Since it may not be obvious to students when assessment begins, such statements should be submitted as soon as possible after the violation occurs. While assessment may begin at any time, in general I will not look at student submissions until a project deadline has passed.

See my online Academic Integrity Policy for an example scenario and more information.

12 Program Outcomes and Competencies

This course is included as an elective in both the BS Computer Engineering program, accredited by the Engineering Accreditation Commission (EAC) of ABET, and the BS Computer Science program, accredited by the Computing Accreditation Commission (CAC) of ABET.

The course introduces students to the following CAC student outcomes, for which graduating students must demonstrate:

(CAC-1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions.

(CAC-2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

(CAC-6) Apply computer science theory and software development fundamentals to produce computing-based solutions.

The course introduces students to the following EAC student outcomes, for which graduating students must demonstrate:

(EAC-1) an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics

(EAC-7) an ability to acquire and apply new knowledge as needed, using appropriate learning strategies

Student outcomes will be evaluated as follows.

CAC 1	CAC 2	CAC 6	EAC 1	EAC 7	Assessment Types
✓	✓	✓	✓	✓	Programming Assignments
✓			✓		Exams

13 Accessibility Resources

From the UB Reasonable Accomodation Policy:

> The University at Buffalo is committed to providing equal access to individuals with disabilities, including physical access to programs and reasonable accommodations for members of the university community.

The UB Accessibility Resources Office provides assistance for students who require reasonable accommodations due to disability. They may be found at 60 Capen Hall or contacted by phone at 716-645-2608. Students must register with their office to receive accommodations for physical or learning disabilities.

14 Critical Campus Resources

Sexual Violence UB is committed to providing a safe learning environment free of all forms of discrimination and sexual harassment, including sexual assault, domestic and dating violence and stalking. If you have experienced gender-based violence (intimate partner violence, attempted or completed sexual assault, harassment, coercion, stalking, etc.), UB has resources to help. This includes academic accommodations, health and counseling services, housing accommodations, helping with legal protective orders, and assistance with reporting the incident to police or other UB officials if you so choose. Please contact UB's Title IX Coordinator at 716-645-2266 for more information. For confidential assistance, you may also contact a Crisis Services Campus Advocate at 716-796-4399.

Mental Health As a student you may experience a range of issues that can cause barriers to learning or reduce your ability to participate in daily activities. These might include strained relationships, anxiety, high levels of stress, alcohol/drug problems, feeling down, health concerns, or unwanted sexual experiences. Counseling, Health Services, and Health Promotion are here to help with these or other issues you may experience. You can learn more about these programs and services by contacting:

- Counseling Services:
  - 120 Richmond Quad (North Campus), 716-645-2720
  - 202 Michael Hall (South Campus), 716-829-5800
- Health Services:
  - 4350 Maple Road (at Sweet Home), 716-829-3316
- Health Promotion:
  - 114 Student Union (North Campus), 716-645-2837

Acknowledgments

Some language in this syllabus is drawn from University policies (as noted), the UB Course Syllabi Requirements document, department guidelines, and other University resources. Some language and structure in this syllabus is drawn from Steve Ko's CSE 486/586 syllabus from Spring 2017 and from Matthew Hertz's CSE 115/503 syllabus from Spring 2019. Additional improvements were made by Karthik Dantu in 2019.