Syllabus
CSE 486/CSE 586: Distributed Systems

University at Buffalo, Computer Science and Engineering

Spring 2026

Ethan Blanton

Revision: 2026-01-23

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

	Office Hours		
Ethan Blanton	Mon	13:00–13:50	Davis 334
eblanton@buffalo.edu	Wed	09:00–09:50	Davis 334

Student Assistants

SA names and their office hours can be found on the SA office hours calendar on Piazza.

Sections

Section	Course ID	Lecture Time	Location
CSE 486 LEC-000	10525	MWF 11:00–11:50	Knox 109

Course Website

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

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

1. 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 8). 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.

2. Course Description

CSE 486/CSE 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.

3. Prerequisites

All students registered for CSE 486/CSE 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 must have experience with, or be able to pick up, the Go programming language 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.

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. This text is available to all students in electronic form through the UB library system.

Lecture slides will be provided electronically throughout the semester on a best-effort basis.

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 and UBlearns classroom 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. Course Requirements

The following items are required of every student, and failure to complete them may affect student grades as described in Section 8, *Grading Policy*, below.

6.1. 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. Lecture attendance will not be taken regu-

larly, but in-lecture quizzes may be given from time to time. Historical data suggests that students who attend lecture regularly perform significantly better than students who do not.

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.*

6.2. 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.

6.3. 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.

6.4. Tests and Quizzes

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: 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: lectures, programming assignments, written homework assignments, and assigned readings.

6.5. 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 (Saturday, Sunday, and University holidays do not count toward these 24 hours) will incur a penalty of 20% of the possible score.
- 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 student assistants are obligated to provide assistance for programming assignments after the assigned deadline.

7. Course schedule

The order of course topics, and important dates (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 as covered in lecture may change slightly due to student interaction and interest, and that this may have small effects on the due dates of programming assignments. Every effort will be made to adhere to the schedule laid out here, but small 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**
2026-01-21	First day of class
2026-02-13	Programming Assignment 1 due
2026-03-06	Programming Assignment 2 part 1 due
2026-03-11	Midterm Exam
2026-03-16–2026-03-20	*Spring break*
2026-04-10	Programming Assignment 2 part 2 due
2026-05-05	Programming Assignment 3 due
2026-05-05	Last day of class
2026-05-08	Final Examination

The following schedule of topics is an estimation, and is subject to change. Readings labeled "DC" are from *Distributed Computing: Principles, Algorithms, and Systems* by Ajay D. Kshemkalyani and Mukesh Singhal

Week	Topic	Reading
2026-01-21	Introduction / Model of Distributed Systems	DC Ch. 1: 1.1–1.3, 1.5–1.8
2026-01-26	Internetworking	*"End-to-end Arguments in System Design";* Saltzer, Reed, and Clark
2026-02-02	Failures and Time	DC Ch. 15 15.1, 15.2, 15.7; Ch. 3: 3.9
2026-02-09	Logical Time and Naming	DC Ch. 2: 2.1–2.3, 2.6; Ch. 3: 3.1–3.4
2026-02-16	Distributed Hash Tables	*Kademlia: A Peer-to-peer Information System based on the XOR Metric;* Maymounkov and Mazières
2026-02-23	Global States and Multicast	DC Ch. 4: 4.1–4.3; Ch. 6: 6.5
2026-03-02	Multicast and Gossip Protocols	DC Ch. 6: 6.6
2026-03-09	Midterm Exam Weds (March 11)	
2026-03-16	*Spring Break*	
2026-03-23	Elections, Consensus, and Byzantine Failures	DC Ch. 5: 5.10; Ch. 14: 14.1–14.3, 14.5.1
2026-03-30	Mutual Exclusion and Raft	DC Ch. 9: 9.1–9.4; *In Search of an Understandable Consensus Algorithm;* Ongaro and Ousterhout
2026-04-06	Raft	*(same)*
2026-04-13	Quorum Protocols, Transactions and Consistency	DC Ch. 9: 9.7, 9.8
2026-04-20	Locking and Security Considerations	
2026-04-27	Security Considerations and Further Exploration	
2026-05-04	Final Review	

8. 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 8.1 and Section 8.2. 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
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.*

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+.

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

8.1. 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.

8.2. Exam Re-grading Policy

If you believe that an exam 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 being returned to you, must be submitted in writing (email is acceptable) to the instructor, and must include:

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

Re-grading of exams 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 exam, 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.

8.3. 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.

9. 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.

In summary, avoid disrupting the classroom via late arrivals or early departures; distracting behaviors such as talking, watching videos, playing games, or viewing non-course content; eating and drinking; *etc.*

10. 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 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.

- 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://go.dev/
- Lecture material from this course
- Required or recommended readings from lecture material

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.

Generative artificial intelligence (AI) or large language models (LLM) are not permitted resources for this course. This includes sites and technologies such as ChatGPT, Bard, the conversational interfaces on search engines like Bing or Google, and tools such as Cursor or GitHub Copilot, as well as *all other* AI assistants. Usage of these technologies to produce material to be turned in for this course, to assist with producing or understanding material in this course, to debug programs for this course, or any other purpose related to this course, is a violation of the course academic integrity policy.

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 SA *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.

10.1. Sharing Course Materials After Completion

Sharing of course materials *after* completing this course is also a violation of the academic integrity policy for this course, as discussed in the University AI policies under "Improper Distribution of Course Materials." 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.

10.2. 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 the instructor, in writing,* of the violation *before assessment of the violating assignment has begun.* 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.

11. 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.

f/ (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

12. 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."

If you have any disability which requires reasonable accommodations to enable you to participate in this course, please contact the Office of Accessibility Resources in 60 Capen Hall, 716-645-2608, and also the instructor of this course during the first week of class. The Office of Accessibility Resources will provide you with information and review appropriate arrangements for reasonable accommodations, which can be found on the web at: http://www.buffalo.edu/studentlife/who-we-are/departments/accessibility.html.

13. 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.

Counseling Services Students may experience a range of issues that can cause barriers to learning or reduce their 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 concerns. Students can learn more about these programs and services by contacting:

- Counseling Services:
  - 120 Richmond Quad (North Campus), 716-645-2720
  - First Floor Michael Hall (South Campus), 716-829-5800
- Student 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.