COLLEGE OF ENGINEERING, UIC

CS 454 Principles of Concurrent Programming

4/3 Credit Hours (Graduate/Undergraduate)

Spring 2026

Instructor & Course Details

Instructor: Luís Pina
Email: luispina@uic.edu (you should use Piazza for any class-related communications)
Drop-In Office Hours:
- In-person: **Monday/Wednesday 3h30pm-4h30pm**
- Remote (via Zoom): **Friday 2pm**
- It is possible to schedule extra office hours by appointment

Drop-In Hours Location:
- In-person: CDRLC 4448
- Zoom: https://uic.zoom.us/j/86231794707?pwd=WZh07qfKQvo0AJOv03jyqwnsPSIJqE.1

TA: Joseph Wiseman
Email: jwisem6@uic.edu
Drop-In Office Hours:
- In-person: **Tuesday 3h30pm-4h30pm**
- Remote (via Zoom): **Thursday 2pm**

Drop-In Hours Location:
- In-person: CDRLC 2404
- Zoom: https://uic.zoom.us/j/86231794707?pwd=WZh07qfKQvo0AJOv03jyqwnsPSIJqE.1

Course site: https://cs474-uic.github.io/cs454-s26-site/
Course modality: On campus
Days/Times: **Monday/Wednesday 2pm-3h15pm**
Location: CDRLC 2406 (also on Zoom)

Course Information

Catalog Course Description and Prerequisite/Corequisite Statement

This course examines the theory and practice of multiprocessor synchronization, and will be focused on the foundations and basic principles of concurrent programming. The course will start with an idealized model of computation, and move on to real models such as the standard Java memory model or C++11 memory model. The foundational approach to concurrent programming will then be supplemented with reasoning about the performance of concurrent data-structures used in a realistic

environment, and how decisions made at the architecture level impact the design of high-performant concurrent data-structures. This will involve revisiting the idealized model of computation introduced earlier, with a more pragmatic approach that fits modern (and future) architectures.

Prerequisite/Corequisite:

- Basic understanding of the software development steps: compilation, linking, debugging, etc.
- Students are expected to have completed CS 361 Systems Programming with a grade of C or better, or its equivalent from other universities.
- You should have a good knowledge of C and (at least) a rudimentary knowledge of the Java language. This course will have a sizable portion of programming in the Java language, and we will introduce Java basics in class assuming a decent C proficiency.

Growth Mindset

Course materials and assignments can be complex and challenging, but they are crucial to your intellectual and personal growth and development. There are times you may need extra help. Students who attend class consistently, complete all assignments, thoughtfully engage with feedback on work, develop good study strategies, visit the tutoring center, and contact faculty when struggling can develop a thorough understanding of the course material and ultimately succeed in the course.

Course Goals and Learning Outcomes

- **Know** the fundamental principles that govern concurrent programming.
- **Understand** how modern computer systems shape concurrent programming.
- **Be able to design and implement** a correct concurrent program.
- **Reason** about the correctness of concurrent programs written by others.
- **Be able to design and implement** an efficient concurrent program.
- **Reason** about the efficiency of concurrent programs written by others.
- **Be proficient** in different styles of concurrent programming in a modern programming language.

Course requirement status: Elective
Topics overview: Mutual exclusion, Shared memory; Thread communication; Memory models; Correctness of concurrent code; Concurrent objects; Spin-locks; Monitors; Barriers, Concurrent data structures; Futures and asynchronous programming; Work scheduling and work distribution; Non-blocking algorithms: Lock-free and Wait-free; Modern hardware support for concurrency: Multiprocessors, atomic instructions, cache hierarchies; Transactional memory.

Required and Recommended Course Materials

Required materials:

- **Required textbook:** Maurice Herlihy and Nir Shavit, The Art of Multiprocessor Programming, Morgan Kaufmann, 2008
  - You don't need to read the textbook before class. You don't need to buy the book as it is available for free online using the UIC's library resources: https://i-share-uic.primo.exlibrisgroup.com/discovery/fulldisplay?docid=alma9949477150300589 7&context=L&vid=01CARLI_UIC:CARLI_UIC&lang=en&search_scope=MyInst_and_CI&adaptor=Local%20Search%20Engine&tab=LibraryCatalogPlus&query=any,contains,the%20art%20of%20multiprocessor%

20programming&sortby=date_d&facet=frbrgroupid,include,9063158651041097571&mode=basic&offset=0

- JSR-133: Java Memory Model and Thread Specification
- The Java Language Specification, Java SE 11 Edition

Required Technology

- Personal computer
- Internet access to follow assignment introductions and solutions, submit assignments, video screencasts
- Development environment: InteliJ IDE, there's a video explaining how to set this up on your machine
- Accounts for services used in this class: Gradescope, iClicker, Github

Respect for Copyright

Please protect the copyright integrity of all course materials and content. Please do not upload course materials not created by you onto third-party websites or share content with anyone not enrolled in our course.

Course Policies & Classroom Expectations

Grading Policy and Point Breakdown

You must complete Assignment 0 and the Syllabus Quiz during the first 2 weeks of class to receive a grade higher than F in this course:

Assignment 0 ensures you understand how to start and submit the programming assignments. It consists of following a video tutorial and: (1) Starting an assignment on Github Classroom; (2) Checking your assignment grade through the autograder; (3) Performing a trivial change to the original code given the feedback on the autograder; (4) Submitting the updated code; and (5) Checking your final grade on the autograder.

Syllabus Quiz ensures that you read and understand this document by answering 20 questions, with a focus on the logistics and academic integrity.

Grade components:

Undergraduate	Rubric	Graduate
15%	Attendance	5%
50%	Programming assignments	50%
15%	Midterm exam	15%
20%	Final exam	20%
—	Research presentation	10%
100%	**TOTAL**	**100%**

Attendance

Attendance will be measured using one of two alternatives:

1. In-person clicker-style quizzes during class. Attend the class in-person and answer all clicker-style questions via iClicker. For full marks on one class you have to answer 80% of all questions, regardless of correctness. This means that incorrect answers count as much as correct answers.
2. Remote attendance: Attendance via Zoom is allowed provided that the majority of students attend in-person. Students should use Zoom as an exception for days you cannot be on campus. If in-person attendance drops below 50%, iClicker availability via Zoom will be disabled.
   (a) Instructors will perform random "spot checks" by asking students to press a particular button. If a student fails to do so, the instructors will assume they are passively pressing buttons instead of paying attention, and such student will not get attendance for that class.
   (b) Failing several random "spot checks" will be considered a violation of the academic honesty policy, and will incur in the penalties described later in this document.
3. Asynchronous offline lecture-capture watch. A quiz will be made available after each class on Gradescope, in which you have to answer a subset of the clicker-questions correctly as you watch the lecture recording.
   (a) When multiple options are correct, you have to select them all. This should be clear when following the lecture recording.
   (b) Some clicker questions in class may be skipped in the quiz. Echo360 will have a modified slide deck in which each Gradescope question is clearly marked.
   (c) The quiz must be completed until Sunday at 5pm of the same week as the class.
4. For full attendance, you have to get full marks on 80% of all classes by either attending them in-person or filling out the respective offline quizzes (you don't have to do both). You can miss 4 classes (equivalent to two full weeks of lectures) and still get full attendance. For example, for undergraduate students:
   (a) If you miss 4 classes, your attendance will be 15%
   (b) If you miss 5 classes, your attendance will be (27 – 5) / 27 x 15% = 81% x 15% = 12%

Programming assignments

Programming assignments will be made available and submitted through Github. You need to create an account with Github and fill out a form with your account name; the URL of the form will be posted on Piazza.

- To submit a programming assignment, you simply push your changes to the repository that will be created for you. There will be an autograder made available through Github. The autograder will run every time you submit new code, and it will generate a grade based on how many tests your submission passes.
- The date of submission is the date of the last commit in your repository.
- This course will use the InteliJ IDEA as the editor for your programming assignments. IDEA is free, available for all common platforms (e.g., OSX, Linux, and Windows) and supports editing, debugging, and submitting your assignments. You are responsible for installing IDEA on your machine. We will cover in class how to import, edit, and submit your assignments.
- Non-determinism: What makes concurrent programming so hard is the amount of non-determinism

(e.g., something seems to work on a machine but fails on another). As such, a portion of each assignment will be based on questions or screencast videos (submitted through Gradescope) that require you to explain a particular aspect of your submission (e.g., what steps did you take to ensure that two threads cannot change the same variable concurrently).

- Uploading videos to Gradescope may take some time, so you should not leave this to the very last minute. Please contact the instructors though Piazza if you experience technical difficulties to upload your submission.
- You can record such a screencast video by using Zoom, which you should already have installed for attending remote office hours. Simply start a meeting without any attendants, then share your screen, then start recording.
  * Note that Zoom requires some time to process your video before it is available for downloading. Students cannot upload videos late because Zoom was processing them, failure to upload videos will result in a 0
  * Technical difficulties reported 2h or less before the deadline will result in a zero grade for the screencast component of that assignment.

- Assignments will be released and due according to the following schedule:

Assignment	Release date	Due date
0	**01/12/2026**	**01/24/2026 5pm**
1	**01/21/2026**	**02/07/2026 5pm**
2	**02/11/2026**	**02/21/2026 5pm**
3	**03/04/2026**	**03/14/2026 5pm**
4	**03/18/2026**	**04/04/2026 5pm**
5	**04/08/2026**	**04/25/2026 5pm**

Exams

Exams will take place in-person.

- Midterm exam: During class hours (**2pm-3h15pm**) on **March 2nd 2026**
- Final exam: Time, date, and location to be determined

Research Presentation

Graduate students will have to become familiar with a topic related to concurrent programming that was not approached in class. The topic should be related to multi-threading and concurrent programming in any form: special hardware to support high levels of concurrency, data-structures with high performance when accessed by multiple threads, programming models focused on multiple threads, state-of-the-art research on multi-threading performance and/or correctness, etc. The research presentation is submitted as a 10 minute video that covers the topic, and will be graded according to the following rubric:

25% Define the problem

25% Why trivial solutions don't work

25% Describe the approach

25% What is one limitation/problem of the solution presented

Research presentations are required for graduate students only. Students must propose a topic by **April 3rd** and submit their presentation by **May 4th 5pm**. Failure to select a topic by the due date will result in a 0% grade for the presentation component.

Final grade scale: **A** 90% or higher, **B** 80% or higher, **C** 70% or higher, **D** 60% or higher, **F** otherwise. **Important note:** the actual cut-offs may be slightly lower, the instructor will determine the curve once all the components for all the students have been graded (after the final exam).

Policy for Missed or Late Work

Assignment resubmission: After the due date, the instructors will post a screen-capture video of the instructors solving the assignment from the starting code to a 100% grade. You can follow this video and update your submission to fix what you got wrong. You can then resubmit your assignment and get credit for that.

- You can resubmit your assignment within one week after the due date
  - Until **next Saturday 5PM**
- Your original submission must be at least 30% to qualify for any resubmission
  - 50% minimum for graduate students
- Your final grade is the average of both submissions
  - Original submission 80%, resubmission 100%, results in 90%
- Resubmissions are completely optional, you can keep the original submission grade if you don't resubmit your assignment.

Exams: No makeup exams and quizzes will be permitted.

Attendance and Participation Policy

The **attendance policy** is described above.

Participation bonus: Each programming assignment will have a bonus component based on Piazza participation. Each question/answer that you contribute that is tagged with the assignment tag and marked as good by an instructor is worth 5%. Each duplicated question you find by answering with the original question and answer, under the same circumstances as above, is worth 5%. You will be able to get up to 10% or more in bonus points for each assignment.

- Bonus points count towards the original submission when resubmitting:
  - Original submission 30%, 5% bonus points, resubmission 100%, results in 67.5% (the original submission is considered to be 35%)

Academic Integrity

As a student and member of the UIC community, you are expected to adhere to the Community Standards of academic integrity, accountability, and respect. Please review the UIC Student Disciplinary Policy for additional information.

Unless stated otherwise, all work submitted for grading must be done individually. While instructors encourage you to talk to your peers and learn from them, this interaction must be superficial with regards to all work submitted for grading. This means you cannot work in teams, you cannot work side-by-side, you cannot submit someone else's work (partial or complete) as your own.

All the submissions for programming assignments will be checked for plagiarism using an automated tool that checks each submission against all other submissions. The tool is sophisticated enough to detect variable renaming, code transformation (turning a for loop into a while loop, inverting an if statement, moving functions/methods to a different place/file, etc.) and I will investigate personally submissions that are classified as much more similar than average.

Plagiarism on a programming assignment will result in an immediate zero grade for that assignment, a one letter grade drop from your final grade (e.g., from B to C, or from D to F), loss of all bonus points given throughout the semester, and a report filed with the Dean of Students for all parties involved.

Academic dishonesty is unacceptable, and penalties can result in expulsion from the university. All cases are handled via the official student conduct process described at https://dos.uic.edu/conductforstudents.shtml.

In particular, note that you are guilty of academic dishonesty if you extend or receive any kind of unauthorized assistance. You may not solicit code from family, friends, or online forums. Other examples of academic dishonesty include emailing your program to another student, working in a group on a homework assignment, and allowing a tutor, TA, or another individual to write an answer for you. It is also considered academic dishonesty if you click someone else's iClicker with the intent of answering for that student.

If you are struggling with a particular part of your assignments, post a question on Piazza. Other students may be struggling as well and benefit from the discussion around your question. If you see a student asking a question that you know how to answer with a small code example carefully curated from your assignment (e.g., how to use an API to parse a file), it is acceptable to share that code. Use your common sense to not disclose important parts of the assignment. For instance, if you are struggling with Java but not with the concepts of an assignment, showing/sending your code to a fellow student is considered a violation of academic integrity. Posting a question on Piazza about that same issue is rewarded with bonus points.

Use of AI assistants

You are required to be the sole author of all the code you submit for the assignments. Submitting code not authored by you (e.g., copy/pasted from an AI tool) violates the academic integrity policy (described below) and will result in applying all the penalties. When performing plagiarism checks, instructors will include code copied directly from common AI assistants submission to detect such cases.

Use of shared computers

The use of shared computers may lead to inadvertently accessing someone else's code for an assignment, which will be counted as plagiarism. A shared computer is a computer that is accessible potentially by many students. Your own personal computer can become a shared computer if you allow a fellow student to use it in the context of this course. Examples of shared computers include (but are not limited to):

- A public access computer: Library, lab, printer, hotel, etc.
- Using your credentials on another student's computer: Downloading your code to another students computer, logging on your account on Blackboard, Piazza, Github, or Travis on another student's computer.
- Allowing another student to use your computer

If you use your credentials on another student's computer, then fail to log-off securely, and that other student accesses your answers, both are guilty of academic dishonesty. I suggest using incognito/private mode in modern browsers when using your credentials on any computer that may be accessed by someone else. Using shared computers can lead to complications that are considered plagiarism. When using shared computers, play close attention to each submission of each assignment. For instance, the following situations can happen:

- Alice lends Bob her computer so that Bob can check his grade. Later, Alice submits her assignment using Bob's credentials by mistake. Both Alice and Bob are guilty of plagiarism.
- Charlie uses Dana's computer to submit his assignment using InteliJ. Later, Dana is struggling with Github and InteliJ replaces one of Dana's files automatically with files from Charlie's assignment. Dana submits the file without realizing. Both Charlie and Dana are guilty of plagiarism.

Email Expectations

Students are expected to use Piazza for any communication with the instructors. Instructors will not read any email received, and simply reply with a redirection to read this document instead.

Students are responsible for reading all information instructors publish on Piazza. Faculty messages should be regularly monitored and read in a timely fashion.

Instructors may need to communicate directly with students via a private Piazza message. Instructors will assume that students read all private Piazza messages within 24h.

Course Schedule

Week 1. Software and Hardware Basics
Week 2. Realities of Concurrency
Week 3. Mutual Exclusion
Week 4. Concurrent Objects
Week 5. The Java Memory Model
Week 6. Spin-locks
Week 7. Consensus
Week 8. Coarse-grained lists
Week 9. Fine-grained lists

Week 10. Wait-notify and task scheduling
Week 11. Non-blocking progress
Week 12. Queues and the ABA problem
Week 13. Concurrent Hashing
Week 14. Transactional Memory
Week 15. Concurrency in Other Languages
Week 16. Final exam week

Disclaimer

This syllabus is intended to give the student guidance on what may be covered during the semester and will be followed as closely as possible. However, as the instructor, I reserve the right to modify, supplement, and make changes as course needs arise. I will communicate such changes in advance through in-class announcements and in writing via Blackboard Announcements.

Accommodations

Disability Accommodation Procedures

UIC is committed to full inclusion and participation of people with disabilities in all aspects of university life. If you face or anticipate disability-related barriers while at UIC, please connect with the Disability Resource Center (DRC) at https://drc.uic.edu, via email at drc@uic.edu, or call (312) 413-2183 to create a plan for reasonable accommodations. To receive accommodations, you will need to disclose the disability to the DRC, complete an interactive registration process with the DRC, and provide me with a Letter of Accommodation (LOA). Upon receipt of an LOA, I will gladly work with you and the DRC to implement approved accommodations.

Religious Accommodations

Following campus policy, if you wish to observe religious holidays, you must notify me by the tenth day of the semester. If the religious holiday is observed on or before the tenth day of the semester, you must notify me at least five days before you will be absent. Please submit this form by email with the subject heading: "YOUR NAME: Requesting Religious Accommodation."

Classroom Environment

Inclusive Community

UIC values diversity and inclusion. Regardless of age, disability, ethnicity, race, gender, gender identity, sexual orientation, socioeconomic status, geographic background, religion, political ideology, language, or culture, we expect all members of this class to contribute to a respectful, welcoming, and inclusive environment for every other member of our class. If aspects of this course result in barriers to your inclusion, engagement, accurate assessment, or achievement, please notify me as soon as possible.

Name and Pronoun Use

If your name does not match the name on my class roster, please let me know as soon as possible. My pronouns are [she/her; he/him; they/them]. I welcome your pronouns if you would like to share them with me. For more information about pronouns, see: https://www.mypronouns.org/what-and-why.

Community Agreement / Classroom Conduct Policy

- Be present by turning off cell phones and removing yourself from other distractions.
- Be respectful of the learning space and community.
- Use preferred names and gender pronouns.
- Assume goodwill in all interactions, even in disagreement.
- Facilitate dialogue and value the free and safe exchange of ideas.
- Try not to make assumptions, have an open mind, seek to understand, and not judge.
- Debate the concepts, not the person.
- Be willing to work together and share helpful study strategies.
- Be mindful of one another's privacy and do not invite outsiders into our classroom.

Content Notices

Our classroom provides an open space for a critical and civil exchange of ideas, inclusive of a variety of perspectives and positions. Some readings and other content may expose you to ideas, subjects, or views that may challenge you, cause you discomfort, or recall past negative experiences or traumas. I intend to discuss all subjects with dignity and humanity, as well as with rigor and respect for scholarly inquiry. If you would like me to be aware of a specific topic of concern, please email or visit my Student Drop-In Hours.

Resources: Academic Success, Wellness, and Safety

We all need the help and the support of our UIC community. Please visit my office hours for course consultation and other academic or research topics. For additional assistance, please contact your assigned college advisor and visit the support services available to all UIC students.

Academic Success

- UIC tutoring resources
- College of Engineering tutoring program
- Equity and Inclusion in Engineering Program
- UIC Library and UIC Library Research Guides
- Offices supporting the UIC undergraduate experience and academic programs
- Student Guide for Information Technology

Wellness

- Counseling Services: https://counseling.uic.edu/

- Access U&I Care Program for assistance with personal hardships
- Campus Advocacy Network (Title IX): TitleIX@uic.edu and http://can.uic.edu/

Safety

- UIC Safe App (please download for your safety)
- UIC Safety Tips and Resources
- Night Ride
- Emergency: Dial 5-5555 from a campus phone, or (312) 355-5555 from a cell phone

Change log

01/06/2026 Document available

01/12/2026 Added TA office hours