CS 330 Introduction to Real-Time Systems Winter 2025

General Course Info

Time:	5a (MW 1:50pm-3:00pm, F 2:20pm–3:20pm)
Location:	Anderson 329
Website:	https://www.cs.carleton.edu/faculty/tamert/courses/cs330-w25

Instructor Info

Name:	Dr. Tanya Amert
Office:	Olin 301D
Email:	tamert@carleton.edu

Student Hours:	M 3:10pm-4:20pm (6a), W 9:50am-11:00am (2a),
(in Olin 308)	Th 1:30pm-2:30pm, F 12:00pm-1:00pm (3a), or by appt.

Textbook and Resources

Required textbook: *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*, 4th edition, by Giorgio Buttazzo (ISBN: 9783031454103). You should be able to access the PDF online for free through the campus library (https://apps.carleton.edu/campus/library/).

All lecture and homework materials will be available on the course website: https://www.cs.carleton.edu/faculty/tamert/courses/cs330-w25. Papers will be available for download from Moodle, and your programming assignments will be submitted through Gradescope.

We are using Slack for questions and announcements. You can ask about any of the course material on Slack, but you must refrain from posting assignment solutions.

Course Description

How can we prove that dynamic cruise control will brake quickly enough if traffic suddenly stops? How must a system coordinate processes to detect pedestrians and other vehicles to ensure fair sharing of computing resources? In real-time systems, we explore scheduling questions like these, which require provable guarantees of timing constraints for applications including autonomous vehicles.

This course will start by considering such questions for uniprocessor machines, both when programs have static priorities and when priorities can change over time. We will then explore challenges introduced by modern computers with multiple processors. We will consider both theoretical and practical perspectives.

Prerequisites

Computer Science 200/201 and Computer Science 202 (Mathematics 236 will be accepted in lieu of Computer Science 202).

Target Audience

This course surveys the scheduling and synchronization algorithms needed to design a real-time system. We will utilize the proof techniques from CS 202 or MATH 236 to analyze and verify the correctness and runtime complexity of the algorithms discussed. We will also build on some of the data structures introduced in CS 201. In addition, written homework and programming assignments reinforce the concepts through further analysis and implementation.

Given that CS 200/201 is a prerequisite, students are expected to have a working knowledge of a programming language; for this course, we will use Python.

Learning Objectives and Course Content

Throughout this term, we will explore the following scheduling paradigms:

- Uniprocessor scheduling of aperiodic or periodic tasks, executing preemptively or non-preemptively
- Uniprocessor synchronization
- Multiprocessor scheduling of periodic tasks
- Multiprocessor synchronization

We will also read and discuss recent research papers published at Real-Time Systems conferences.

Key Dates

First day of class: Monday, January 6th
Last day of class: Wednesday, March 12th

No class: Monday, February 10th (Midterm Break)

Final presentations: Saturday, March 15th, 3:30pm – 6:00pm

Technology Requirements

In this course, we will be using Python for programming assignments. If you do not have your own laptop, or your laptop ceases to work, let me know. You may also be able to complete these assignments on the department lab machines.

You are, generally, expected to refrain from laptop use during regular class periods (notetaking on a tablet is the exception).

Course Components and Grading

Course Components

Lessons: Class will usually be a mixture of lecture and small-group exercises. Students are expected to have completed a reading assignment prior to each lesson; in class, we may focus on some of the more complex topics from the readings. Some class periods will be devoted to discussions of recent research papers.

Classwork: Many class periods will include in-class exercises ("classwork") designed to gauge pacing and general understanding of the material. You need not prepare outside of class aside from doing the assigned readings.

Assignments: The homework assignments will be a combination of written homework questions to give students practice with the theoretical aspects, as well as programming assignments for practical application of course material. The programming assignments will build up a small codebase for experimenting with different scheduling algorithms and synchronization protocols.

Quizzes: Each week there will be a short quiz focusing on the fundamental concepts we have recently studied. At the end of the term, there will be an opportunity to take alternate versions of two quizzes to potentially replace earlier scores.

Final project: Students will complete a final project, alone or with a partner. The project will consist of a written proposal, a brief proposal presentation, a short written progress update report, a final writeup and optional code, and a presentation during the final exam timeslot.

Paper discussion: On most Fridays, we will be discussing a recent research paper in depth. Small groups (two or three students) will be assigned to prepare the paper discussion; all students are expected to participate in each discussion.

Grading Criteria

Assessment	Count	Percentage
Classwork	~24	5%
Written Assignments	~5	20%
Programming Assignments	~3	15%
Quizzes	8	10%
Paper Discussion Lead	1	14%
Paper Discussion Participant	8	6%
Final Project	1	30%

Assignments (written and programming) will be graded both for correctness and style. For proofs, good style implies that you give enough information to complete the proof, and are not excessively verbose. For programs, this means well-documented (through comments and/or good variable/function names) and easily readable code.

Grade Derivation

Course grades will be computed as a weighted average following the above criteria. Course grades will be converted to letters using the common scale of A: ≥ 93, A-: ≥ 90, B+: ≥ 87, B: ≥ 83, etc. These thresholds may be adjusted downward (i.e., to your benefit) for the entire class, but they will not be increased.

Late Policy

Written assignments must be submitted on paper, at the start of the relevant class period. Programming assignments need to be submitted to Gradescope by the time listed on the assignment webpage. If you do not use a token (explained below), or you do but submit after the updated deadline, late submissions will be penalized 10% within the first hour, or 60% within the first 24 hours.[1] Any work submitted more than 24 hours after its deadline will receive no credit.

Each student starts the term with five tokens. These can be used on any written or programming assignment for any reason, no explanation necessary. One token provides an automatic 24-hour deadline extension. If working with a partner, each person must use their own tokens to extend their individual deadline.

At most two tokens can be used on an assignment. To use a token, you should email me by the assignment's deadline (no explanation, just to say that you're using one).

There is no token use needed to make up missed classwork. You can make up classwork until 4pm on the last day of class.

In the last week of the term, you will have the opportunity to make up two quiz scores by taking alternate versions of those quizzes; your grade for each quiz will be the maximum of the original and any make-up score.

Course Schedule

Note that the schedule is tentative, and subject to change.

Unit 1	Weeks 1-2	Uniprocessor Scheduling of Aperiodic Tasks
Unit 2	Weeks 2-4	Uniprocessor Scheduling of Periodic Tasks
Unit 3	Weeks 4-5	Uniprocessor Synchronization
Unit 4	Weeks 6-9	Multicore Scheduling
Unit 5	Weeks 9-10	Multicore Synchronization

We will not meet for class on Monday February 19th due to Midterm Break.

[1] Mathematically, the late penalty will be a cap on the possible score, e.g., an assignment submitted two hours late and graded to have 85% of the points earned will receive $\max(85, 100 - 60) = \max(85,40) = 40$ points.

Course Policies

Attendance Policy

Students are expected to attend all class periods. If you must miss class, you are responsible for checking the schedule to determine what you missed and studying that material on your own. If you know of an absence in advance, check with me for how you may best prepare.

Quizzes cannot be taken after the day on which they occur. If you know you must miss class, you may be able to take a quiz in advance; discuss with me before your absence. If you're feeling unwell the day of the quiz, we can arrange a separate room for you; if you're too sick to take a quiz, you will need to rely on the opportunity to replace a quiz score during the last week of the term.

Student Hours ("Office Hours")

Student hours, also known as office hours, are a time that you can (and are encouraged to!) stop by to chat with me about anything. This includes questions about material covered in class or in the textbook, questions regarding assignments, quiz preparation, opinions on WoW expansions or D&D editions, the major in general, board games, or any other topic.

You are welcome to attend any scheduled office hours; if you have a schedule conflict, send an email and we'll find a different time.

Classroom Etiquette

In class, you are expected to maintain proper etiquette. This includes arriving on time, not having conversations during lecture, not using your phone in class, etc.

You are expected to take your own notes during class (this can be done on a tablet, but it must behave like a notebook); you should not be taking pictures of the board.

Collaboration Policy

For written assignments, you are encouraged to work together, for example, writing on the board together, but you are expected to write up all of your answers yourself.

For programming assignments, you are encouraged to work with up to one other student. You may look at your partner's code and help with debugging, but *you are expected to type up all of your code yourself*. You may, however, discuss general approaches with others (for example at a white board, or "I had to sort the list of tasks first"), but you should never look at the code of anyone but your partner.

If you worked with someone else, you must each include each other's names at the top of your source files (see assignment instructions for more information). *You*

should never be looking at the code of anyone other than your one partner for a given programming assignment.

Quizzes are always closed-book and closed-note, and must be completed individually.

For the final project, you are strongly encouraged to work with a partner; you will submit your work together as one submission.

Academic Integrity and Generative AI

You are expected to refrain from using any online source for this class. You *should not be using any AI text/code generation tools when completing your assignments, project, or paper discussion preparation*. This includes the GitHub Co-pilot extension in VS Code, which you are expected to disable if you have installed.

You should never be in possession of anyone else's assignment code, and never look at anyone else's assignment code but your partner's.

If any student is suspected to have violated the academic integrity policy, a report will immediately be made to the Academic Standing Committee, as described at https://apps.carleton.edu/campus/doc/integrity. Ask me if you are unsure about what constitutes acceptable collaboration.

Illness/COVID-19 Policy

Some of the policies stated in this syllabus may need to be modified at times due to illness. For example, if I have COVID, we may need to have some class periods over Zoom. If you are unable to attend class, you should talk to a friend to get notes, and let me know so that we can work around your absences to enable you to still achieve mastery over the course content. We will work together to make this term a success, and you will be informed about any necessary changes as soon as possible.

Inclusion

I strive to create an inclusive and respectful classroom that values diversity. Our individual differences enrich and enhance our understanding of one another and of the world around us. This class welcomes the perspectives of all ethnicities, cultures, gender identities, religions, ages, sexual orientations, disabilities, socioeconomic backgrounds, regions, and nationalities.

College Policies

Accommodations for Students with Disabilities

Carleton College is committed to providing equitable access to learning opportunities for all students. The Office of Accessibility Resources (107 College Street) is the campus office that collaborates with students who have disabilities to provide and/or

arrange reasonable accommodations. If you have, or think you may have, a disability (e.g., mental health, attentional, learning, autism spectrum disorders, chronic health, traumatic brain injury and concussions, vision, hearing, mobility, or speech impairments), please contact OAR@carleton.edu to arrange a confidential discussion regarding equitable access and reasonable accommodations

Student Well Being

Your health and well-being should always be your first priority. At Carleton, we have a wide array of resources to support students. It is important to recognize stressors you may be facing, which can be personal, emotional, physical, financial, mental, or academic. Sleep, exercise, and connecting with others can be strategies to help you flourish at Carleton. For more information, check out Student Health and Counseling (SHAC), the Office of Health Promotion, or the Office of the Chaplain.

Reporting Sexual Misconduct

Carleton is committed to fostering an environment free of sexual misconduct. Please be aware all Carleton faculty and staff members, with the exception of Chaplains and SHAC staff, are "responsible employees." Responsible employees are required to share any information they have regarding incidents of sexual misconduct with the Title IX Coordinator. Carleton's goal is to ensure campus community members are aware of all the options available and have access to the resources they need. If you have questions, please contact Carleton's Title IX Coordinator (titleix@carleton.edu) or visit the Title IX website.

Math Tutoring

The Math Skills Center supports all Carleton students in any mathematics course they are taking in which they are experiencing difficulty, either with the mathematical concepts or with the mathematical tools needed to succeed in the course. Their mission is to "level the playing field" by giving students who enter Carleton without strong mathematics backgrounds the tools they need to succeed here at Carleton.

Writing Center

The Writing Center provides a space staffed with peer writing consultants who can work with you during any stage of the writing process (brainstorming to final proofreading). Hours and more information can be found on the writing center website. You can reserve specific times for conferences by using their online appointment system.

Support for Students who use English in Addition to Other Languages

If English is not your primary or home language and you believe you might benefit from working regularly with a writing consultant this term, email Melanie Cashin, Multilingual Writing Coordinator, at mcashin@carleton.edu. She can arrange

once- or twice-a-week meetings between you and a specific writing consultant throughout the term.

Disclaimer

As instructor, I reserve the right to make changes to the syllabus, including assignment due dates, exam dates, and class cancellations or modality changes, for example due to weather or illness. These changes will be announced as early as possible.