Course Syllabus

✎ Edit

# Course Syllabus

## Prerequisites

To be successful in this class, students should meet the following prerequisites:

- **Foundational Computer Systems Courses:** Completion of courses such as Data Structures and Algorithms (CS 1332) and Computer Systems and Networks (CS 2200) is required. Familiarity with operating systems (CS 3210) and introductory database systems (CS 4400) is recommended.
- **Proficiency in C++:** Comfort with C++ programming is essential, as all assignments are based on this language. You should either be proficient in C++ or be prepared to learn it quickly.
- **Systems Programming Experience:** A solid background in systems programming and software engineering fundamentals is necessary to manage the complexities of database system implementation.
- **Complex Problem-Solving Skills:** The course involves deep analytical work and requires strong problem-solving and algorithmic thinking abilities.
- **Willingness to Learn:** An openness to new technologies and advanced database concepts is crucial, as the course covers cutting-edge techniques in database systems.

## Educational Objectives

This is the first of a two-part series on designing and implementing database management systems, with a strong focus on hands-on programming. By the end of this course, you will:

- Master the internal mechanics of relational database systems.
- Apply advanced storage management techniques, including slotted pages and memory management.
- Implement key indexing data structures like B+ trees and hash tables.
- Develop techniques for parsing and executing complex SQL queries.

Throughout this course, we will explore:

- Introduction to Relational Databases
- Storage Management
- Access Paths
- Query Execution

In the second part of the series, we will cover:

- Logging and Recovery Methods
- Concurrency Control
- Query Optimization

## Grading Scheme

The final grade for the course will be *tentatively* based on the following weights:

- 50% Exams (Mid-Term Exam: 22.5%, Final Exam: 27.5%)
- 27.5% BuzzDB Programming Assignments
- 22.5% Exercise Sheets

## Programming Assignments

The programming assignments are designed to complement the lecture topics, using **BuzzDB (https://buzzdb-docs.readthedocs.io/index.html)** , an end-to-end toy relational database management system created specifically for this course. Through these assignments, you will explore and apply concepts covered in lectures in a practical, hands-on way.

## Assignment Deadlines

You can access all assignments by selecting the “Assignments” tab on Canvas. When in doubt about the deadline for an assignment, you can always check Canvas, as all assignments are listed there. To avoid problems, you should make sure to configure Canvas so that all deadlines reflect your local timezone. To do so, you should go to Canvas ➡ Account ➡ Settings ➡ Edit Settings and Time Zone and select your local time zone. Also, please note that **the deadline for assignments is the “Due” date, whereas the “Until” date indicates the late submission deadline** (see the Late-Submission Policy section below).

## Exercise Sheets

The exercise sheets, administered as quizzes on Canvas, are designed to closely mirror the exams you will take during the semester. Both the exercise sheets and the exams will feature multiple-choice questions (MCQs) covering the topics taught in class. So, the exercise sheets are practice for the exams, the style, format, and difficulty are essentially the same.

## Ed Discussion

Ed Discussion is our online forum, which you will use to communicate with us (and vice versa).

- **Accessing Ed Discussion:** To access the forum, click the Ed Discussion link on the Canvas page for this class (on the menu on the left). We will add registered students to Ed Discussion

before the class starts. **If you have registered for (or been added to) the class after classes started, you may get a message that an access code is required** when using the Ed Discussion link until newly registered students are added to the Ed Discussion roster, which happens daily. In this case, please do not contact the instructors to ask for the code and instead allow a day for that process after you have been added to Canvas (and watch your official email for notice that you have been added).

- **Class-related communications: Make sure to use Ed Discussion, rather than email or Canvas messages, for ALL class-related communications**. We can miss email messages, but we are careful about checking messages on Ed Discussion and making sure that your questions are answered in a timely fashion. Most importantly, by asking questions on Ed Discussion, you can benefit from the collective knowledge of your classmates, instructors, and TAs. For this reason, I encourage you to ask questions publicly, rather than privately to the instructors, as that maximizes your chances to get a prompt reply and, most importantly, allows your classmates to see questions, answers, and discussions that can benefit them. In fact, sometimes we wait before responding to posts because we want to encourage student responses and discussions. It is just as important that you also check Ed Discussion postings regularly.
- **Keeping up with Ed Discussion:** Besides benefitting from questions, answers, and discussions, as mentioned above, **Ed Discussion is where we post announcements and other useful information about the class, and we assume that you read these announcements**. Finally, posts are clearly appreciated, but please don't feel like you have to post at all costs. You can get 100% Ed Discussion participation by simply following the discussions on Ed Discussion. (Ed Discussion keeps track of each student's online presence.) In other words, just make sure that you check updated, unresolved, and unread posts on a regular basis, and you will be fine even if you do not post yourself. This also helps to avoid duplicate postings, in which students ask questions that have already been asked (and often answered).
- **Duplicate posts: Please make sure to check the existing posts before asking a question**: duplicated posts waste both our and your time. For similar reasons, you should also **make sure to read the assignments and the syllabus carefully before posting questions about them on Ed Discussion**. Finally, many of the questions we receive on Ed Discussion can be answered with a simple web search, so please make sure to try that before posting on Ed Discussion, which frees up resources for more interesting questions, answers, and discussions.
- **Our response time on Ed Discussion:** We do our best to respond to Ed Discussion messages daily, or at most every couple of days (depending on volume and other factors). If you don't get an answer to a post within 48 hours, feel free to post a follow-up, but please avoid doing so a few hours after posting (i.e., we do our best to be responsive on Ed Discussion, but the forum is not a 24/7 hotline :-). Please also avoid reposting the same message twice to get more attention; you should add a follow-up to your existing post instead. Related to the previous point, I try to answer most of the posts that are not grading challenges myself, but that creates issues during

particularly busy weeks/semesters. (Just to give you an idea, I sometimes spend about an hour every day on Ed Discussion.) In the last few semesters, I may therefore delegate some of the day-to-day Ed Discussion interactions to the TA team. However, I am actively following Ed Discussion and am happy to answer questions directly addressed to me, which I also do during the weekly office hours. Feel also free to reach out to me with direct Ed Discussion messages if you have complaints about some specific interactions. If you do so, please point me to specific posts involved, as that allows me to get context and suitably discuss the issue with the TA team.

- **Declines on Ed Discussion:** In line with what we mentioned above, we may decline (i.e., disregard) on Ed Discussion questions that are clearly answered in the syllabus or in the assignment description.

## Exams

There are **two online exams scheduled**, and these will be monitored using Honorlock. Each exam will include a combination of subjective and objective questions that reflect the topics discussed in the course.

## Textbook

- Silberschatz, Korth, & Sudarshan: **Database System Concepts (https://db-book.com/slides-dir/)** . Seventh Edition. McGraw-Hill, 2019 [ISBN: 9780078022159].

## Accommodations for Students with Disabilities

If you are a student with learning needs that require special accommodation, contact the Office of Disability Services at (404) 894-2563 or **http://disabilityservices.gatech.edu (http://disabilityservices.gatech.edu/)** , as soon as possible, and make an appointment to discuss your special needs and obtain an accommodation letter. Please also send an email to me as soon as possible in order to discuss your learning needs.

## Academic Honesty

All Georgia Tech students are expected to know and abide by the Georgia Tech Academic Honor Code (**https://osi.gatech.edu/students/honor-code (https://osi.gatech.edu/students/honor-code)** ). University Policies will be followed strictly in this course. Please, pay particular attention to academic misconduct. In this class, in particular, although we generally encourage the discussion of the course material, the exercises, assignments, and projects with your classmates, we draw a firm line regarding what sharing and copying are permissible in your coursework. Specifically, you must adhere to the following rules:

- **You may not, under any circumstances, copy any code from any current or former student in the class, or from any public sources addressing the same content as the course**

**assignments/projects.**

- **The only source code you are permitted to borrow are isolated project-agnostic snippets**, meaning source code that serves a specific purpose and that are applicable outside the context of our projects (such as iterating through an array). In these cases, you must bracket the borrowed code with comments identifying the source exactly as follows:

/* BEGIN CODE FROM (*source URL*) */

… borrowed source code …

/* END CODE FROM (*source URL*) *

The source URL must link to a web page containing the borrowed source code snippet. We will not accept lists of sources provided without context.

- Using tools based on machine learning generative models, such as ChatGPT or Github Co-pilot, to generate solutions to assignments or parts thereof is also strictly prohibited.
- You may reuse your own work or code from past private endeavors, such as previous course attempts or different class assignments. However, any reused code or work must strictly conform to the policies of the current semester. Even if past work did not violate a prior course's policies, if it violates this course's current policies (for instance, relating to the use of tools like ChatGPT), it cannot be reused without proper adjustments. Additionally, should any of your past work be publicly available, you must treat and cite it in the same manner as any other public source, as specified above. **Collaborative works from the past, such as work done for a prior class group project, cannot be reused in this course.**
- There will often be watermark-type fields added to projects to date the project to a particular term. If an assignment is turned in with a prior term's watermark, we will assume that cheating has occurred and the penalties for cheating will be enforced.
- **Do not post publicly or share any content (solutions or parts thereof) with other students in the class**.
- When in doubt, don't hesitate to ask: in this case, it is definitely NOT better to beg for forgiveness than to ask for permission.

Please keep in mind that we use plagiarism-detection software to identify similarities among submissions, between submissions and online materials (e.g., repositories of students from previous semesters), and between submissions and materials generated by tools based on machine learning generative models. Because we have access to the same resources to which you have access, our plagiarism detector will most likely identify these cases (as it has done countless times in the past).

**Violations of this policy will be subject to the institute's Academic Integrity procedures, which may include a 0 grade on assignments found to contain violations, additional grade penalties,**

**academic probation or dismissal, and prohibition from withdrawing from the class.** The bottom line is that it is not fun for anybody when a case of academic misconduct is discovered, so please avoid collaborating beyond what is allowed by the Georgia Tech Academic Honor Code.

**Important:** If you are accused of academic misconduct, you are not permitted to withdraw from the class until the accusation is resolved; if you are found to have participated in the misconduct, you will not be allowed to withdraw for the duration of the semester. If you do so anyway, you will be forcibly re-enrolled without any opportunity to make up work you may have missed while illegally withdrawn.

## Communication Policies

Use Ed for asking all course-related questions as they would also benefit other students. You can ask “private” questions as well.

## Late Submission Policies

A daily late penalty of 25% will be applied for each day an assignment is submitted past the deadline. We will permit four penalty-free late days throughout the entire semester.

## Grading Scale

The grading scale for this course includes the following breakdown:

- A (Excellent) - 4 points: Achieved with a score of 90% and above.
- B (Good) - 3 points: Requires a score between 80% and 89%.
- C (Satisfactory) - 2 points: Awarded for scores between 70% and 79%.
- D (Passing) - 1 point: Earned with a score between 60% and 69%.
- F (Failure) - 0 points: Assigned for scores below 60%.

We may adjust the grading thresholds downward at the end of the course.

## Hardware and Software Requirements

- Hardware Requirements
  - **Processor**: A modern CPU with 4–8 cores (e.g., Intel or AMD or Apple).
  - **Memory**: Minimum of 4 GB.
- Software Requirements
  - Browser and connection speed: An up-to-date version of Chrome is required. We use proctoring software Honorlock for our examinations and it currently only supports Chrome. 2+ Mbps is recommended.
  - Compiler:
    - C++ 17-compatible compiler (e.g., g++ or clang++)

    - Check that you can compile code with the `-std=c++17` flag
- Operating system:
    - **Linux**: A modern distribution (e.g., Ubuntu 20.04+) released within the last four years.
    - **macOS**: Version 11 (Big Sur) or later, including M1/M2 devices
    - **Windows**: Version 10 or later (released within the last four years). **(https://visualstudio.microsoft.com/)** C++ compiler via **(https://visualstudio.microsoft.com/)** **Windows Subsystem for Linux (WSL).** **(https://docs.microsoft.com/windows/wsl/)**
-