

GEORGIA TECH · COURSE SYLLABUS

CS 6422: Database System Implementation

Course Syllabus

GT

COURSE GUIDE

Table of Contents

Prerequisites	4
Educational Objectives	4
Course Resources	5
Ed Discussion	5
BuzzDB Documentation	6
Grading Scheme	6
Programming Assignments	6
Assignment Release	6
Assignment Deadlines	7
Gradescope Submission	7
Gradescope Leaderboards	7
Local Testing and Autograder Expectations	7
Assignment Debugging Questions	7
Exercise Sheets	8
Exams	8
Honorlock	8
Scratch Paper, Calculator, and Cheat Sheet	9
Assessment Procedure Issues	9
Textbook	9
Academic Honesty	9
Code Attribution	10
LLM and AI Tool Use	10
Academic Integrity Review	11

Accommodations for Students with Disabilities	11
Late Submission Policies	11
Grading Scale	12
Grading, Feedback, and Regrades	12
Hardware and Software Requirements	13
Hardware	13
Browser and Connection	13
Compiler	13
Operating System	13
Recommended Development Setup	14
Optional CLion and Docker Setup	14

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 programming 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 important, as the course covers advanced 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 such as 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, the course sequence covers:

- Logging and recovery methods
 - Concurrency control
-

- Query optimization

Course Resources

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, we 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, Ed Discussion is where we post announcements and other useful information about the class, and we assume that you read these announcements. Posts are welcome when they add a question, answer, or useful clarification, but you do not need to post just for the sake of posting. Make sure that you check updated, unresolved, and unread posts on a regular basis. 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. The TA team will answer most routine Ed Discussion questions, with instructors monitoring the forum and stepping in when needed, especially for policy questions, assignment clarifications, recurring concerns, or posts addressed directly to the instructors. For concerns about a specific Ed Discussion interaction, please use a private Ed message and include links to the relevant posts so staff can review the context.

- **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.

BuzzDB Documentation

Programming assignments use BuzzDB, an end-to-end toy relational database management system created for this course. Use the [BuzzDB documentation](#) for assignment details and setup guidance.

Grading Scheme

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

- **50% Exams**
- Midterm exam: 22.5%
- Final exam: 27.5%
- **27.5% BuzzDB programming assignments**
- **22.5% Exercise sheets**

The most recent official syllabus or instructor announcement is the source of truth for grade breakdowns. Canvas assignment-group weights may lag behind staff updates. If Canvas, the syllabus, and another course source disagree, please post on Ed with screenshots or links to the conflicting sources so staff can confirm the correct policy.

Programming Assignments

Programming assignments are designed to complement the lecture topics using BuzzDB. Through these assignments, you will apply lecture concepts in a practical, hands-on way.

Assignment Release

Use the latest official assignment instructions and staff announcements. Assignment files are separate from the public BuzzDB repository. The public lecture repository is designed for following course material, but assignment files are released through Canvas when the project opens. Ed may be used for announcements, clarifications, and updates.

Assignment Deadlines

Programming assignment deadlines are listed in Canvas under the **Assignments** tab.

Gradescope Submission

Submit only the specified source file listed in the assignment instructions, such as `buzzdb_lab1.cpp`. Upload it directly to Gradescope. Do not compress it into a `.zip` file unless the assignment explicitly asks for a zip, and do not change the filename.

Gradescope is the main way programming assignments are evaluated. Your submission must implement the required behavior and respect the allowed code boundaries, interfaces, and collaboration rules. Staff may manually review submissions when needed, especially if a submission appears to bypass required behavior or violate the spirit of the assignment.

Gradescope Leaderboards

Some assignments may include a leaderboard to compare performance with classmates. Leaderboard participation is optional and purely for learning. Do not use your real name on a leaderboard.

Local Testing and Autograder Expectations

Local or visible tests are not a complete specification. Gradescope may run tests in a different environment, with stricter compiler flags, larger inputs, edge cases, persistence checks, operator chaining, concurrency behavior, or cases not covered by local tests.

If your code compiles locally but fails on Gradescope, compile locally using the same style of command used by Gradescope:

```
g++ -fdiagnostics-color -std=c++17 -O3 -Wall -Werror -Wextra <file_name.cpp> -o <output_name.out>
```

Warnings are treated as errors. If your local compilation command omits `-Wall`, `-Werror`, or `-Wextra`, you may miss issues that Gradescope sees. Before posting on Ed about a local machine-vs-Gradescope mismatch:

- Confirm that you did not change function signatures, class layouts, expected filenames, or autograder-facing code.
- Port only your intended TODO changes.
- Compile with the Gradescope-style command.
- Test inside the provided environment when possible.

Assignment Debugging Questions

Use the assignment-specific Ed category or subcategory when available. It helps the on-call TA find and answer questions faster. For autograder or debugging questions, include:

Assignment:

Question category: compile failure / timeout / autograder test / local-vs-Gradescope / output formatting

Environment: Docker / Ubuntu / Mac / WSL / other

Compile command used locally:

Gradescope test number:

Exact Gradescope error:

Does it reproduce locally?

What I already checked:

Smallest relevant code region or invariant:

Exercise Sheets

Exercise sheets are administered as quizzes on Canvas, and these will be monitored using Honorlock. They are designed to closely mirror the exams. During exercise sheets, work without LLM assistance and follow the allowed-resource rules for that assessment.

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.

All assigned course material is fair game unless the staff explicitly narrows the scope. This includes lectures or modules, assigned textbook readings, assigned papers, and concepts reinforced through programming assignments. Later assessments naturally build on earlier concepts, but students should rely on the dedicated exam logistics thread when it is released. During exams, work without LLM assistance and follow the allowed-resource rules for that exam.

Honorlock

Honorlock requires an up-to-date version of Chrome. If Honorlock asks for an access code, students normally should not need to obtain one manually. Try the standard troubleshooting steps:

- Use Chrome.
- Uninstall Honorlock.
- Clear cookies.

- Reinstall Honorlock.
- Log in again and retry.

If Honorlock blocks access, denies access, or a technical issue interrupts an assessment, contact Honorlock support immediately, document what happened, and make a private Ed post. Include timestamps, screenshots if available, and any support transcript.

Scratch Paper, Calculator, and Cheat Sheet

Follow the assessment instructions exactly. If scratch paper, a calculator, or a cheat sheet is allowed, show the required items clearly during the room scan and again at the end if instructed.

Assessment Procedure Issues

For assessment procedure issues, use a private Ed post and include:

Assessment:

Issue type: Honorlock / access code / timer / interruption / extension

Approximate time of issue:

What happened:

What I did immediately:

Honorlock/support case or transcript, if any:

Attachments: scratch paper photos, screenshots, support messages

Question for staff:

Textbook

- Silberschatz, Korth, and Sudarshan. **Database System Concepts**. Seventh Edition. McGraw-Hill, 2019. ISBN: 9780078022159.
- Companion materials: <https://db-book.com/slides-dir/>

Assigned textbook readings may be tested even if the exact topic was not emphasized in lecture. The course uses lectures, readings, papers, and projects as different angles on the same database systems concepts.

Academic Honesty

All Georgia Tech students are expected to know and abide by the [Georgia Tech Academic Honor Code](#). University policies will be followed strictly in this course.

Although discussion of course material, exercises, assignments, and projects is generally encouraged, there is a firm line between permitted discussion and impermissible sharing or copying. You must adhere to the following rules:

- You may not copy code from any current or former student in the class.
- You may not copy code from public sources that address the same content as the course assignments or projects.
- You may not post publicly or share solutions, partial solutions, or assignment content with other students in the class.
- Collaborative work from past classes or group projects cannot be reused in this course.
- Submitting files, code, or documents from a prior term, including materials with prior-term identifiers or watermarks, may be treated as evidence of academic misconduct.

Code Attribution

The only public source code you are permitted to borrow is isolated, project-agnostic snippets, meaning code that serves a specific purpose and applies outside the context of the course projects. Examples include generic snippets such as iterating through an array.

When permitted, bracket borrowed source 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. Lists of sources provided without context are not sufficient.

You may adapt code shown in lecture slides or BuzzDB for your assignment when allowed by the assignment instructions. Add a clear attribution comment, such as `Adapted from source X`.

LLM and AI Tool Use

You may use LLMs as a learning aid, similar to using documentation, a tutor, or a search engine. Appropriate uses include asking conceptual questions, learning syntax, getting explanations of compiler errors, brainstorming debugging strategies, or checking your understanding of a topic.

You are still responsible for everything you submit. Do not submit AI-generated work that you do not understand, cannot explain, or have not verified. Do not use LLMs to bypass the learning goals of an assignment, solve graded work for you, or violate course collaboration or academic integrity policies.

If AI assistance substantially shaped your submitted solution, disclose it in a comment or attribution note, just as you would acknowledge other meaningful help.

Do not paste private course materials, exam questions, private Ed posts, classmates' work, or confidential information into external AI tools.

LLM assistance is not permitted during exams or exercise sheets.

Academic Integrity Review

Course staff may use plagiarism-detection software to identify similarities among submissions, between submissions and online materials, and between submissions and materials generated by machine-learning tools.

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.

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 misconduct, you will not be allowed to withdraw for the duration of the semester. If you withdraw anyway, you will be forcibly re-enrolled without any opportunity to make up work you may have missed while withdrawn.

When in doubt, ask for permission before acting.

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/> as soon as possible and make an appointment to discuss your special needs and obtain an accommodation letter. Please also contact the course staff as soon as possible to discuss your learning needs.

Late Submission Policies

Four penalty-free late days are permitted throughout the semester across programming assignments. They do not apply to exams or exercise sheets. After those late days are used, a daily late penalty of 25% will be applied for each day a programming assignment is submitted past the deadline.

Extensions for exams or exercise sheets require staff approval and, for serious or repeated cases, appropriate documentation from the Dean's office. Staff may limit extensions for fairness to classmates.

If you have travel or personal constraints, plan around the published assessment availability window.

Grading Scale

The grading scale for this course includes the following breakdown:

- **A:** 90% and above
- **B:** 80% to 89%
- **C:** 70% to 79%
- **D:** 60% to 69%
- **F:** Below 60%

Grading thresholds may be adjusted downward at the end of the course.

Final cutoffs, rounding, and any curve will be posted after final exam grades and remaining adjustments are complete.

Grading, Feedback, and Regrades

Staff will announce approximate grade-posting timing when possible, especially before important decision points such as the withdrawal deadline, exams, or final grade submission.

When staff announce that project grades are finalized, late penalties and known adjustments have been applied. If something appears wrong, make a private Ed thread with the specific discrepancy.

Do not publish exercise sheet or exam questions or answers in public Ed threads, especially while extensions are still active. If feedback is enabled, review it through the assessment platform. If you need clarification, open a private Ed thread and include the relevant question or answer text if allowed by staff instructions.

For feedback or regrade questions, include:

Assessment/assignment:
Question number or copied randomized question:
My answer:
Marked-correct answer:
Why I think this needs clarification:
What I am asking staff to check:

For missing grades, include:

Assessment/assignment:

What Canvas/Gradescope shows:

Why I believe this is incorrect:

Approximate completion/submission time:

Evidence: screenshots, submission link, logs, confirmation email

Question for staff:

If your final letter grade appears wrong, make a private Ed thread with your calculated score, Canvas score, expected letter grade, and the specific discrepancy.

Hardware and Software Requirements

Hardware

- Processor: A modern CPU with 4 to 8 cores, such as Intel, AMD, or Apple silicon.
- Memory: Minimum of 4 GB.

Browser and Connection

- An up-to-date version of Chrome is required for Honorlock.
- A connection speed of 2+ Mbps is recommended.

Compiler

- Use a C++17-compatible compiler, such as `g++` or `clang++`.
- Check that you can compile code with the `-std=c++17` flag.
- Assignments and the Gradescope environment are configured for C++17. Code that depends on C++20 or C++23 features may compile locally but fail on Gradescope.

Operating System

- Linux: A modern distribution, such as Ubuntu 20.04 or newer, released within the last four years.
- macOS: Version 11, Big Sur, or later, including Apple silicon devices.
- Windows: Version 10 or later. Windows students should use Windows Subsystem for Linux (WSL) or the provided course development environment when applicable.

Recommended Development Setup

VSCode is recommended for editing C++ code. The VSCode C++ documentation is available at <https://code.visualstudio.com/docs/languages/cpp>.

Use the course [Makefile](#) to simplify building and testing instead of manually typing long compiler commands when the assignment includes the Makefile.

Common Makefile commands:

```
make          # build the project in debug mode
make release  # build the project in release mode
make test     # run all tests
make test 6   # run a specific test, such as test 6
make clean    # clean build artifacts
```

Optional CLion and Docker Setup

Georgia Tech students can use CLion through the JetBrains student program: <https://www.jetbrains.com/academy/student-pack/>. CLion's quick-start guide is available at <https://www.jetbrains.com/help/clion/clion-quick-start-guide.html>.

If an assignment provides Docker files, you may use Docker to run the programming assignment inside a consistent container. The course setup files include a [Dockerfile](#) and [docker-compose.yaml](#). With CLion's remote toolchain feature, you can still get IDE support such as code completion, debugging, and build/run commands while the assignment runs inside the container.

See the Georgia Tech student software list for additional software available to Georgia Tech students: https://docs.google.com/spreadsheets/d/1Gk3IPDd7_WyKeSfyuxXWrh4Hkk80As82CnaOO2m750M/edit#gid=0.