COMP 530: Database System Implementation

Description

After you complete this class, you will be able to answer the question: How does one design and build a database system? During the semester, we will discuss database management system architecture, buffer management, query processing and optimization, transaction processing, concurrency control and recovery, data storage, indexing structures, and related topics. Most significantly, students will build a database system from the ground up, that will involve writing at least a couple of thousand lines of C++ code.

Instructor: Chris Jermaine (cmj4@rice.edu). Office hours are right after class Wednesday for one hour (in the Brochstein Pavilion) and Thursday 9:30 to 10:30 (also in the Brochstein Pavilion).

Why take this class? Few people will ever implement a database system from scratch, and so you might be wondering: Why am I here? Perhaps the biggest reason to take this class is that it is one of the best opportunities—maybe the best opportunity—at Rice to get experience with hard core systems programming. At the end of the semester, you'll have a complete system totaling around 10,000 lines of code that are all working together to solve a single problem (though you won't write all 10,000 lines; I'll provide a lot of that for you). It is certainly the only opportunity you'll have at Rice to get experience with hard core C++ systems programming.

4 credit hours vs. 3 credit hours: **Important!** If you are a student who has no knowledge of databases I'll give you a crash course on relational databases so that you can still take the class. If this is you, you'll need to enroll in the 4-credit version of COMP 530. For those who take this option, there will be some video lectures, plus an additional programming assignment where you will get some practice writing code in SQL. You **cannot** choose this option if you have taken COMP 330/430/533 or the equivalent.

If you have knowledge of databases (that is, you have taken COMP 330/430/533 or the equivalent) but you still wish to take a 4-credit hour version of COMP 530, there will be an additional, optional assignment at the end of the class that you can complete. This will be assigned the last week of classes and due at the end of finals. For those that either take the 3-credit hour option, or who have not previously taken a database course, the final assignment will be due at the end of classes.

Note that you need to decide **now** whether you are taking the class for 3 or 4 credit hours, and then you need to sign up appropriately. Changing the credit hours after the add/drop deadline is not possible other than under exceptional circumstances, and requires a petition process which may or may not be successful.

Finally, I will also say that you should take the 4 credit hours only if you really need to take it (that is, if you have not taken a DB class, or you need a fourth credit hour in order to meet graduation requirements). Each year, there are several people who decided to take an extra credit hour because it seemed like no big deal the first week of the semester. Then they regret that decision when they are debugging multi-threaded code during finals.

Important note: COMP 530 is not simply the graduate version of COMP 430/533. If you just want to know how to use a database system, then take COMP 430/533; COMP 530 is the wrong class. However, if you want to know how to *build* a database, or you want experience designing and implementing a large software system using a "real" system programming language (C++), or you just like building software, this is the class for you!

Prerequisite: Basically, students must be competent at systems programming, and have knowledge of databases.

On the *systems side*, students must have reasonable systems programming skills, be somewhat comfortable using the C language, and the desire to build a large software system. For undergraduates, that means (at least) that you should have taken COMP 321. Note that while we will be using C++, if a programmer is comfortable using C and Java, it is possible to quickly pick up C++.

On the *database side*, you must have a basic knowledge of relational databases. For undergraduates, that means you must have taken 430. For graduate students, it means that you should have taken an undergraduate database course, or it means that you should enroll in the 4-credit version of this course.

Course Difficulty: This is a challenging and time-consuming course. After all, you'll be writing a database system from the ground up. In my experience last year, students who had taken COMP 421/521 before COMP 530 found the course to be much easier than those who had not. I would say that even graduate students who had taken the equivalent of COMP 421/521 elsewhere (outside of Rice) found that they were not necessarily prepared for the course. While COMP 421/521 is not a prerequisite for COMP 530, you should seriously consider taking it before you take 530. Also, I'll say emphatically that this course is **not** the place to learn about C and pointers. If you are not totally comfortable with pointers and self-managed memory in C, A1 is going to be a rude awakening!

Textbook

My lectures will closely follow the material in Database System Implementation by Garcia-Molina, Ullman, and Widom. The material in that book is also available in Database Systems: The Complete Book, by the same authors. Either one of these

books will be valuable, but not necessary.

Lectures, Meeting Times and Locations

Class will be held Monday, Wednesday, Friday from 1:00 to 1:50 in Ryon 201. Aside from the evening lectures for students taking the extra credit hour, all lectures will exclusively be given using the blackboard, old school. Most students enjoy this more than slides. But the downside is that if you miss class, you'll need to obtain the notes from a friend.

Registration

You're responsible for registering for COMP 530 with the university registrar. For undergrads, prerequisites will be strictly enforced. For graduate students, I'll be more open to letting people in.

Communication

The class will have a Piazza forum for all day-to-day communication:

https://piazza.com/rice/spring2020/comp530/home

It is expected that if you have a technical question on an assignment or an upcoming exam, you will post it to the forum rather than sending an email to the instructors. This guarantees a fast response and means that everyone can benefit from the question and the answer. In general, only inquiries of private or personal nature should be made directly to the instructor ("I need to go out of town on Oct 22nd, can I have an extra day..."). Everything else should be posted on Piazza. You'll get faster feedback from the group than you can get from your instructors.

If you have any communication of a more personal nature and wish to contact the instructor of the class, please send email to Chris, **and include the word "530" in the subject line**. Please realize that O get a lot of random email, so if you do not include 530 in the subject line, your email will likely be ignored.

Assignment handouts and turnins, as well as your grades, will be on Canvas. Everything else will be on Piazza.

Grading and Evaluation

If you are taking the class for 3 credits, your grade is based upon a warm-up programming assignment (4% of your grade) assigned the first week of classes, a set of eight programming assignments (64% of your grade; each is worth 8% of your grade), and a set of ten, short, in-class quizzes (32% of your grade). Each quiz will take approximately 20 minutes and the date will be announced beforehand. You can drop the two lowest scores that you receive on the quizzes; all others are worth 4% of

your grade each.

If you are taking the class for 4 credits, there is an additional assignment. In this case, each of the eight “regular” assignments (aside from A0) is worth 6% of your grade That additional assignment is worth 16% of your grade.

Your numeric grades will be published to you in Canvas.

Final grades are based on the numeric grades, where 90-100 is an A, 80-89 is a B, and so forth. We reserve the right to apply a “curve” to change this, but only for the better. That is, if you’ve gotten 90%, you’re guaranteed at minimum an A- for your final grade, but you might do better.

Assignments

This is an assignment-oriented class. There will be eight programming assignments, all completed completed in teams of two (or alone, if you really prefer). Both students always get exactly the same score on each assignment. Students my switch or leave partners during the first two days that any assignment is out, but no switching is allowed outside of that two-day window.

The assignment dates and due dates for the assignments will be:

A0 C++ Warm-Up: out Monday, Jan 13th, in Monday, Jan 20th
A1 Buffer and file management: out Monday, Jan 20th, in Thursday, Jan 30th
SQL assignment: out Monday, Jan 20th, in Thursday, Feb 13th (only for 4 credit students who do not have database knowledge; this assignment is completed individually)
A2 Record manager: out Thursday, Jan 30th, in Thursday, Feb 6th
A3 Sorted file implementation: out Thursday, Feb 6th, in Tuesday, Feb 18th.
A4 B+-tree implementation: out Tuesday, Feb 18th, Sunday, March 1st
A5 SQL type checking: out Sunday, March 1st, In Sunday, March 8th
A6 Relational operators: out Sunday, March 8th, in Friday, March 27th
A7 Optimization: out Friday, March 27th, in Friday, April 10th
A8 Putting it all together: out Friday, April 10th, in Friday, April 24th
A9 Final project: out Friday, April 19th, in time of final exam (only for 4 credit students who are not completing the “database basics” assignment)

Lateness and Missed Assignments

In general, I am quite generous with extensions on assignments, provided you ask one week (or more) in advance of the due date. Within one week of the due date, extensions are not granted. So if you are going to be very busy and will need an extension, make sure to ask before the one week deadline.

Assignments must be turned in by 11:55PM (5 minutes before midnight) on the day

that they are due. You can turn in an assignment up to 24 hours late, in which case you receive a 10% penalty (that is, 10 points are subtracted from an assignment that is worth 100 points), or up to 48 hours late, in which case you receive a 20% penalty. Assignments turned in after that are not accepted. Please note that your turnin time is whatever Canvas says, and your turnin is whatever you turn into Canvas, **no exceptions**. Be safe; submit early and often!

A missed quiz results in a zero, again: **no exceptions**. I realize that people travel and need to miss class, but that's why I'm dropping the two lowest scores.

We kept on saying **no exceptions**, but there are exceptions in very extreme circumstances, with proper documentation. For example, if you obtain a doctor/dentist note stating that you were so ill at the due date/time that you could not reasonably be expected to meet the deadline, it is possible to get an extension.

Regrade Requests

These must be made within **one week** of an assignment/midterm being returned, during Chris' office hours, to Chris in person. Sending an email does not constitute a regrade request. When you talk to Chris, he'll help you understand whether you've got a legitimate request. If you do, then you'll write that request down formally, print it on paper, and hand it to Chris. Chris will batch these, and them periodically and issue final grade adjustments in bulk for everybody.

Academic Misconduct

In a programming class, there is sometimes a very fine line between "cheating" and acceptable and beneficial interaction between peers. Thus, it is very important that you fully understand what is and what is not allowed in terms of collaboration with your classmates. Our goal here is to be 100% precise, so that there can be no confusion.

The rule on collaboration and communication with your classmates is very simple: you cannot transmit or receive code from or to anyone in the class in any way---visually (by showing someone your code), electronically (by emailing, posting, or otherwise sending someone your code), verbally (by reading code to someone) or in any other way we have not yet imagined. Any other collaboration is acceptable.

The rule on collaboration and communication with people who are not your classmates (or your TAs or instructor) is also very simple: it is not allowed in any way, period. This disallows (for example) posting any questions of any nature to programming forums such as StackOverflow.

As far as going to the web and using Google, we will apply the "two line rule". Go to any web page you like and do any search that you like. But you cannot take more than two lines of code from an external resource and actually include it in your assignment in any form. Note that changing variable names or otherwise transforming or

obfuscating code you found on the web does not render the "two line rule" inapplicable. It is still a violation to obtain more than two lines of code from an external resource and turn it in, whatever you do to those two lines after you first obtain them. Furthermore, you should **cite your sources**. Add a comment to your code that includes the URL(s) that you consulted when constructing your solution. This turns out to be very helpful when you're looking at something you wrote a while ago and you need to remind yourself what you were thinking.

Finally, I'll point out that codes from last year (including my own codes) are likely floating around out there. It is a violation of the honor code to look at any of those codes.

Any violations of these rules will be reported to the Honor Council. Just don't do it!

Students with Disabilities

Students with disabilities should contact the course instructor and Disability Support Services regarding any accommodations that they may need.