

CSSE230 Winter 2022-2023 (a.k.a. 202320)

What is this course about?

What will I learn?

Your goal: After successfully completing this course, you will be able to **independently analyze, develop** an clear, and efficient algorithms and data structures. You will start to think like a Computer Scientist.

What will I do?

Your work required:

1. To learn to analyze algorithms (exact and big O runtime of code that uses loops, nested loops, and recursion) you will **complete 1 homework set each week**. Mostly these will be written problems, but occasionally they will be programming problems.
2. To independently develop and debug correct, clear, and efficient software, you will **complete 1 major project**. You will learn to plan your design on paper and to use the debugger to trace your code.

Do I have what it takes to do this?

The formal prerequisites are MA 112 and a grade of C or better in CSSE 220, so we expect that you are comfortable programming in Java, know how to program and debug object-oriented Java code well, and that you have experience coding linked lists, recursion, and algorithms. Needed attitudes are (1) a willingness to work hard, (2) patience to plan your code before writing it, (3) willingness to work cooperatively and responsibly with partner(s) on pair and team assignments, and (4) willingness to do analysis.

Here is a more detailed [list of prerequisite attributes](#), for those who like lists.

What kind of stuff will I learn?

Why ArrayLists **double** in internal capacity when they fill up and we add another element.

How fast looping **and recursive** code runs.

Why balanced Binary Search Trees allow you to lookup AND insert items, both in $O(\log n)$ time!

What are the two underlying techniques to store ANY collection of data?

How to choose data structures:

Is a **linked list** better than an array list? It depends!

Is a **balanced binary search tree** better than a sorted array? It depends (for the same reason!)

When is a **binary heap** better than a balanced BST? For a common, but very specific use case!

Why not use **hash sets** for everything?

How to implement all of the above data structures.

And much, much more.

What habits of the mind will I learn?

If we are successful in teaching you, ten years from now, you will know:

...that you often need to wrack your brain planning your code before you ever type a line of it. So **you'll code**

...that you have no idea if your code does what you think it does unless you step through it. So **you'll use the**

...that it often pays to write **and rewrite** code so it is **elegant**.

Who, when, and where? Help!

Class Meeting Times and Places

- *Section 02 (Krohn)*: 10:00-11:50, M O269; WF O267
- *Section 03 (Jelen)*: 1:00-2:50, M O269; WF O267

Instructor Information

Rachel Krohn

Email: krohn@rose-hulman.edu

Office address: Moench F226

Office hours: See the resources Moodle page.

Ben Jelen

Email: jelen@rose-hulman.edu

Office address: Moench D208

Office hours: See the resources Moodle page.

Course Assistants

Graders for written assignments: Evan Slater, Dominic Csomos, Blaise Swartwood.

Grading programming assignments: Brock Buczkowski and JL Koenig.

In-class TAs: Jared Kagay (Section 2) and Allyn Loyd (Section 3).

Lab Assistant: Ben Joens. See the [CSSE230-only TA Schedules](#) for when they are available.

Many Other Sources of Help

- Learning center tutors include many willing to help with CSSE230. See the [LC Tutor Schedules](#).
- Besides the instructors and tutors, **other students** in the course can often be a great source of help. Ask them to explain things to you.

- Don't try to be the Lone Ranger in this course, especially if you do not find the course easy. If you find it takes 30 minutes without making any progress, it's probably time to seek help! Software development is a team sport that a fresh set of eyes can often spot a problem right away.
- **But you aren't ready for an exam until you can write the code yourself.** If you got lots of help, write your own exam. It may take you more time, but it will help internalize what you've learned.

Textbook? No.

There is no required textbook for this offering of CSSE230. For most students, our course materials, along with Wikipedia, provide enough reference material. If you're looking for more supplemental resources, please ask your instructor.

Where is the course online?

We will use Moodle to post grades and materials that require restricted access, like lecture videos, quizzes, and assignments. Moodle is the hub of the course.

Starting code for most programming projects will be provided using Git repositories (details in the first program).

What are the homework policies?

Your solutions to **weekly programming problems** should be well-designed and well-documented. Some will be individual. Each submitted program file should include (in comments at the top of your files) your name(s) and should use reasonable and consistent **Javadoc** comments, style, and indentation. **Longer methods should explain why** you wrote the code the way you did. Your programs should not contain lines that are exceedingly unreadable (e.g., printouts). Grades for programming problems will be based on **correctness, style, and efficiency** of your code to your git repository.

We will assign weekly **homework problems (written and short coding exercises)** and a few **in-class exercises**, mathematical analyses, or algorithm-design exercises. We expect you to think through them carefully (if you can't write it neatly, type it). On some problems, not only the correctness but also the quality of your solution matters. Some of the problems will be straightforward practice with concepts from the course; others will require creative problem-solving. Submit your solution by using [Gradescope](#) within Moodle.

There will often be **daily quizzes**, which differ significantly from traditional quizzes. (Note: the quizzes are narrow in scope, since in the hybrid course format, you will complete them **before** class as you are watching the video. The questions should be contained in the lecture. The quizzes should help you to focus on some of the lecture material to stay on track during discussion time, and to have some notes that you can use for review later. They will be graded.)

Late Assignment Policy

All assignments must be turned in before the due time if you want credit for them .

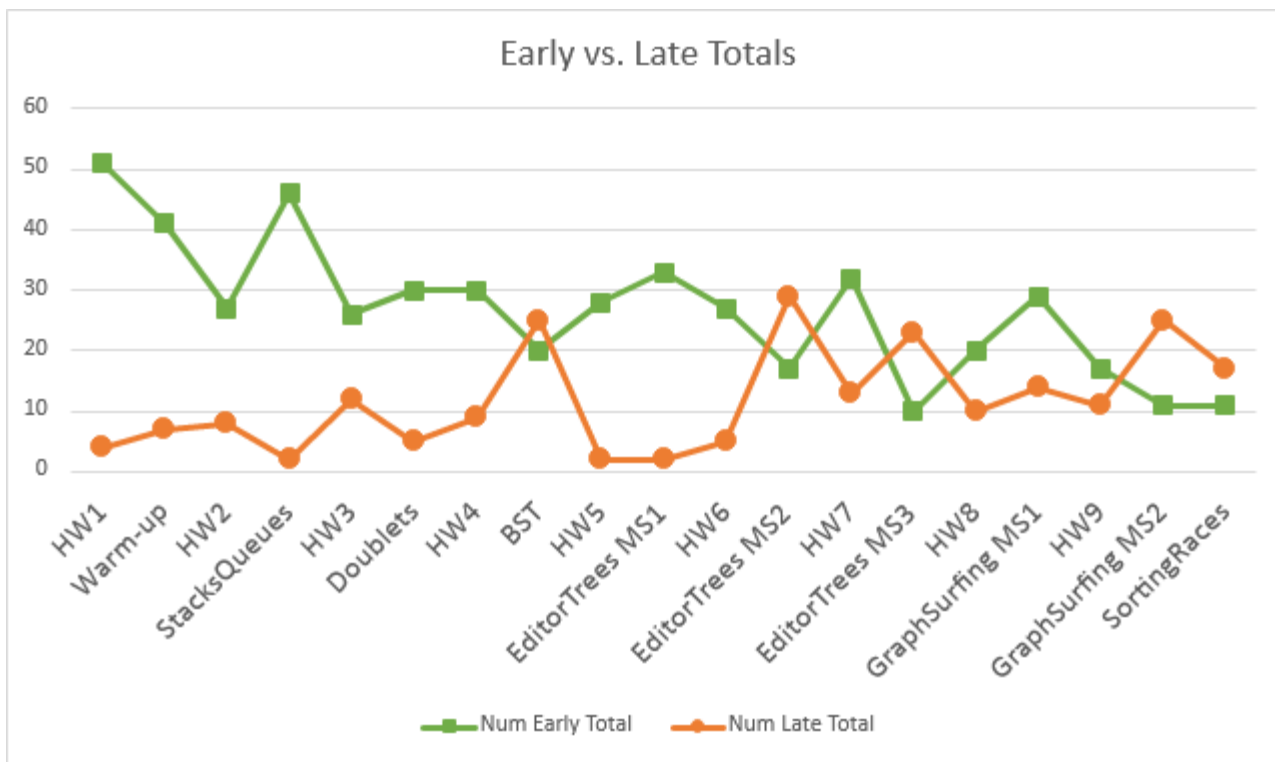
However, we all have days when we are extremely busy, or times when a program takes longer to complete than expected. So we give each student a **"late day bank account"** that starts with **three** late days. Note: this late day policy applies to programming projects. It cannot be used for in-class quizzes and activities, which must be turned in on time for credit.

1. Using (withdrawing) a late day allows you to turn in any assignment up to 24 hours after the time it is due.

that time frame, if it falls on a non-class day.

2. You may earn (deposit) a late day by turning in an assignment at least 24 hours early (We will sometimes have no limit to the number of days you can save up. Extra late days at the end of the term are never redeemable for a small, small number of extra-credit points. 8-) If you find a mistake on homework submit on time.
3. Your late day balance can dip below 0, but it must be ≥ 0 by the end of the quarter. If your late day balance dips below 0, talk to your instructor to bring it back to 0, which typically involves zeroing out assignments until you reach 0 or one any time before the break, consider that a sign that you need to “press harder on the accelerator”.
4. At most one late day may be used or earned for any given assignment. Talk to your instructor in advance with unusual circumstances requiring more than 1 day.


Here's a handy chart of recent quarters (Spring 2021, Winter 2022) detailing when people earned and used late days. You'll thank yourself later!



Late Day Procedures: You do not have to notify us when you earn or use a late day. Just track your balance early days based on the time of your submission to a Moodle assignment page or your latest commit time of a

Some particular assignments may be designated as “no late days“ assignments. This might happen because:

- there is an exam the next day or the day after, so we want to post solutions right away; or
- what we will do during the next class meeting depends on this assignment.

†**Occasionally**  , we will allow extra time for everyone to complete a particular assignment without “extending” the deadline. If you are working on an assignment during a grace period, you should do so with the recognition that you must finish it and move on to the next assignment. If we decide to give a grace period for an assignment, we will ex

Exams

There will be three midterm exams and a final exam, all but Exam 2 having a paper part and a computer part. (Exam 2 has a computer part only.) Exams 1 and 3 will be evening exams, taking the place of a day of class. **Why evening exams?** We schedule evening exams to give students to take the exam at the same time. Also, because it is not easy to judge the time required for the projects, we allow a few additional minutes at the end if they are needed.

How will my grade be figured?

Weight Criteria

5%	Daily quizzes
5%	Citizenship
35%	Written assignments and programming projects
13%	Exam 1
7%	Exam 2 (programming only)
15%	Exam 3
20%	Final Exam

Final grades are also contingent on the following:

- You **must** have a passing (60+) average on the exams to pass the course.
- You must demonstrate that you can **individually** write and debug simple Java programs at the level of proficiency expected. In particular, the programming parts, will be the usual way to do this. The instructor reserves the right to change the weight of a major category (homework, exams, project) if it differs significantly from the overall average.
- **Attendance is required**, as detailed below in the note on how citizenship affects your grade.

The above is a guideline that we typically follow. Please understand that it is not a promise. We will do our best to follow the definition of the various grades, as described in the [Academic Rules and Procedures](#). As you read it, note that “thorough competence to do excellent work” appears in the description of the “B” grade (the standard for “A” is “superior”). “B” and “B+” will not be given for mere compliance with the minimum essential standards of the course.

Citizenship Counts!

Your default citizenship grade is 80% (4 of 5 points) for average engagement in the CSSE 230 learning community (person) attendance, engagement, adherence to deadlines, voluntary positive participation in class and online partnership in pair and group assignments, timely completion of various surveys, and peer evaluation of other members for group projects. If you come to class and sit quietly and cooperatively, this will be your grade. The grade can be lowered if you do not volunteer regularly in class. The easiest way to lower this grade is to miss class or come late, without an excuse.

Java Version

You should be using Java 15 to compile your assignments to ensure the test cases are running correctly. We've had some issues with JUnit tests.

To address this, ensure you check out [Updating Compiler to Use Java 15](#).

What is the course format?

We will follow a partially flipped-classroom approach: video lectures before class, and homework help and pre-consensus of recent student feedback points to this model as the most efficient usage of students in- and out-

Course content will be delivered in virtual form through videos and daily quizzes, which should be completed in a session of class. Completion of these activities will be tracked through Panopto logs and submission of quizzes. Policies for the synchronous sessions:

- To foster our course's sense of community, and facilitate efficient and effective work, students are required to attend each in-person session, as directed by your instructor. Many students choose to work in class for the full pair assignment.
- You will turn in your lecture quiz at the beginning of the session.
- This synchronous time will be used exclusively for CSSE230 work time: working on written or programmatic assignments with a partner, and with ready access to the instructor's help. No non-CSSE230 distractions or multi-tasking! It is viewed as fusing work time with office hours.
- In the interest of everyone's health, students feeling unwell or with health concerns should contact their instructor virtually.
- We aim to fully leverage the in-class time set aside for this course. Toward this end, two unexcused absences are allowed. **Three or more unexcused absences may result in failure of the course.** Habitually arriving late will

Is it OK for my friends to help me with my homework?

It depends how they "help" you...

Recall the [Institute policy on academic misconduct](#)  :

"Rose-Hulman expects its students to be responsible adults and to behave at all times with **honor and integrity**

Exams and homework will be done on an individual basis except where explicitly noted. The simple rule of thumb is:

Never give or use someone else's code or written answers.

Such exchanges are definitely cheating and not cooperation.

We encourage you to discuss the problems and general approaches to solving them with other students. However, code, it must be your own work (or the work of your group if it is a group assignment). If you are having trouble with code works or pinning down a run-time or logic error in your program, by all means talk to someone about it. (This is not cheating.)

If you use someone else's ideas in your solution, you must:

- give credit to that person in the comments of your program, and
- be sure that you understand it as well as if it were your own.

If you are ever in doubt about whether some specific situation violates the policy, the best approach is to discuss it with your instructor.

This is a very serious matter that we do not take lightly. Nor should you.

You should never look at another student's code to get ideas of how to write your own code. Beginning the process with an electronic copy of work done by other students is never appropriate.

Working on written problems with other students is strongly encouraged. However, once you have solved a problem solution individually, without referring to the common solution, to make sure that all of you understand it. Again appropriate.


Plagiarism (where a student solution to an exam or assignment was copied from another student's solution that is posted anywhere) will result in a score of -100% for the assignment or exam. Egregious conduct in this course. Furthermore, such cases **will also be reported to the Department Head and Dean of Students**, as added to the student's record and so discourage repeat offenses. More importantly, such dishonesty steals your opportunity to learn. So don't cheat!

Making our classroom welcoming

We want you and the other students to feel welcomed in our classroom

If at any point, ***you are not comfortable in the classroom, for ANY reason, or you observe any behavior:*** (by course assistants or your instructors) that ***may make the classroom climate feel less welcoming for students:*** please let us know.

Ways to do so include:

- Ask to meet with your instructor privately. Or, explain your concern to your professor after class or by email if the issue does not require a conversation.
- Talk with any instructor in the department with whom you feel comfortable.
- All your professors will do their utmost to ensure your confidentiality, but sometimes you might feel the room is momentarily frustrated. Take a deep breath, count to 10, and if you still want to express a concern, by a private message.
- Know your campus resources beyond the CSSE department as well. These include your faculty advisor and [Student Affairs](#)  .
- Or, if you want to express concerns in a completely confidential way, the [Student Counseling Center](#) (see their website for how to contact them.)

You can do your part to ensure a welcoming, professional classroom

- Speak to your classmates, course assistants and instructors with courtesy and professionalism. The classroom is a place to learn and grow, so avoid off-color jokes, rude language, or just about anything that might offend someone.
- Be supportive of your classmates. Offer help where appropriate. Help your classmates feel confident and capable.
- Project respect to classmates, course assistants and instructors. (And tell us, as described above, if you observe any behavior that may make the classroom climate feel less welcoming for students or others in the classroom.)
- Avoid words and behaviors that might be perceived as confrontational or aggressive. Strive to avoid negative phrases like "you're wrong" or "you need to...". Instead, use neutral "I/us" phrases like "I think that that approach is not right" or "I'm having trouble with this, can you help?"

- Whenever possible, arrive to class on time.
- Set your phone to a “silent” mode, turn off notifications on your computer that might distract classmates professional images on your computer screen.
- Don’t do anything that will detract from your learning or that of people around you. Such things include not taking adequate care of your personal hygiene.
- Restrict all your conversations in class to things related to the class, except for times when your instructor allows a between-periods break, for example).
- If you are experiencing issues that may make you less able to be respectful to others in the classroom, please seek help from the [Student Counseling Center](#) (see below). They are an excellent resource available to students for free.

Working with special needs

Rose-Hulman, and the instructors of this course in particular, are committed to working with students with disabilities. We understand that “invisible” disabilities (learning and attention deficit disorders, chronic fatigue) can significantly affect a student’s academic performance.

We strongly encourage students to document special academic circumstances with the staff at the Office of Student Accessibility as soon as possible so that we can work together to provide recommended academic accommodations while preparing for class. It is the student’s responsibility to request any approved, documented academic accommodations (such as extra time on exams).

Another pair of resources available to students for free:

- The office of [Health Services](#) .
- The [Student Counseling Center](#) . From their website:

The Student Counseling Center provides confidential, culturally sensitive support for a variety of people at Rose-Hulman. We provide brief individual counseling, group counseling, couples counseling, assessments, and resources to enhance the students’ ability to fully benefit from academic and social life at Rose-Hulman. This includes the development of personal awareness, building life skills, and overcoming personal concerns.

When you are feeling stress or experiencing any sort of difficult issue, our services can help you find solutions.

To make an appointment, call them at (812) 877-8537, or visit their office in the Union, rooms 245-250.

Official Course Info

Course Catalog Description

This course reinforces and extends students’ understanding of current practices of producing object-oriented software. It disciplines the design process to include formal analysis of space/time efficiency and formal proofs of correctness of concepts from CSSE 220, including implementations of abstract data types by linear and non-linear data structures and randomized algorithms. Students design and implement software individually, in small groups, and in a chaotic environment.

CSSE Department’s Official Learning Outcomes

Students who successfully complete this course should be able to:

1. Describe classical data structures (list, stack, queue, tree, priority queue, hash table, graph, set, dictionary) implementation choices for each.
2. Explain classical sorting, graph and tree-balancing algorithms.
3. Develop empirical and mathematical analyses of the asymptotic worst, best and average case run times of
4. Justify the choice of an algorithm based on the analysis of several algorithms appropriate for a problem.
5. Design and implement object-oriented programs competently and independently.
6. Implement various data structures, and apply them to medium-sized programming exercises.
7. Work with a team of 2-3 students to implement a complex data structure, using basic software engineering and unit testing, and demonstrating effective team decision making, division of labor and conflict resolution.

Last modified: Sunday, 5 March 2023, 4:57 PM