Comp Sci 214 — Data Structures — Winter 2023

1 Course Description

Comp Sci 214 teaches the design, implementation, analysis, and proper application of abstract data types, data structures, and their associated algorithms. We will explore a wide variety of data structures both conceptually and concretely via implementation. The class involves a significant hands-on programming component; you will both implement and use several data structures yourself.

2 Course Staff

Instructor          Prof. Vincent St-Amour (stamourv@cs.northwestern.edu)  
                    (Call me Vincent)
Teaching Assistants Leif Rasmussen (leifr@u.northwestern.edu)  
                    Tochukwu Eze (TochukwuEze2025@u.northwestern.edu)
Peer mentors       Desmond Adanuty, Anoushka Kapur, Amy Liao, Jackson Miller, Ethan Pineda,  
                    Mattie Poelsterl, Ella Shin, Akash Shroff, Shreya Sridhar, Li Kang Tan, and Michelle  
                    Zhang

3 Prerequisites

• Comp Sci 111 AND (Comp Sci 211 OR Comp Sci 150)
• Comfort with programming
• Basic discrete math. See Huck Bennett’s excellent writeup on Canvas.

4 Learning Objectives

• Functional correctness: can you apply data structures and algorithms concepts to produce implementations that display correct behavior and produce correct results?

• Non-functional correctness: can you produce programs that are efficient, robust, and maintainable, and demonstrate good programming habits and hygiene?

• Theory: can you understand, apply, and contextualize definitions and techniques related to the theoretical aspects of data structures and algorithms: computational complexity, invariants, etc.

• Evaluation: can you make informed decisions when picking data structures and algorithms and evaluating alternatives?

• Integration: can you combine multiple data structures and algorithms to decompose and solve complex problems?
5 Communication Channels

In this class, we will rely on a variety of communication channels, each serving a different purpose.

5.1 Lectures

Lectures will be synchronous and in person. We will make a best effort attempt at recording them. The recordings will be available from Canvas under the ‘Panopto’ tab. Please see the recording policy at the end of this document.

We expect you to come to lecture and participate.

To ensure the health and safety of our community, DO NOT come to lecture if you have reason to believe you may be sick with COVID-19 (or anything else, really). Please watch the recordings instead and contact the instructor. Please see the COVID-19-related policies at the end of this document.

Questions

I love questions! Keep ’em coming. During lecture, please ask questions in one of two ways:

- Raise your hand and ask verbally.
- If you prefer to ask anonymously and/or textually, reply to the Piazza (see below) post of the current lecture. A member of the course staff will then ask your question in your stead.

5.2 Piazza

We will use Piazza for discussion, Q&A, and announcements. You can find the class’s Piazza instance here: https://piazza.com/northwestern/winter2023/comp_sci214

We expect you to subscribe to the 214 Piazza board ASAP and monitor it regularly.

Piazza should be the first place you go when you have a question. The course staff will monitor Piazza (as will many of your fellow students), which makes it the best option for getting answers quickly. Your fellow students may also benefit from the discussion. Any questions about the specifics of your solutions must be marked as private, however.

For questions that require longer discussions/explanations, please attend office hours (see below) instead. If all else fails or for sensitive matters, send the instructor an email, either for discussion or to schedule an appointment.

5.3 Canvas

We will use Canvas to distribute materials, assignments, and feedback. All (non-exam) assignments will also be submitted via Canvas.

We expect you to check Canvas regularly.

Please do not use Canvas messages/comments to communicate with us; they are very easy to miss. Please use Piazza instead.

5.4 Office Hours

We will hold a combination of online and in-person office hours, circumstances permitting. We will maintain an up-to-date schedule (with locations) on the ‘Office Hours’ page on Canvas. We will monitor attendance patterns and adjust the schedule to the best of our ability to limit crowding.

Online office hours will be held using the gather.town platform. The room we will use is at: https://app.gather.town/app/fQLpVpi2VqWouD4N/Comp-Sci-214-Office-Hours

See the ‘Office Hours’ page on Canvas for the password to the room.
To ensure we can help students in a first-come first-served order, and to help us group up people with similar questions, we ask that you sign up on this form when you arrive to office hours, or when you have a follow-up question: https://forms.gle/lpC2AXdkLeaDpkw99

You can look at the state of the queue to give you an idea of how close you are to being next in line: https://tinyurl.com/2f4jar72

Expectations

Office hours are a great place to get advice and assistance on all aspects of the class (and even CS in general!) In particular, they are not limited to debugging assistance; conceptual questions are welcome too!

Keep in mind, however, that our course staff is here to help you learn, not to give you solutions or to solve problems for you. An important part of the learning process is learning to figure things out and solve problems on your own; having members of the course staff do that for you would be doing you a disservice!

This means you should expect our course staff to give you advice and strategies or point you to relevant resources (e.g., documentation, slides, videos), but not to hold your hand (so to speak) while you work through problems. This will be especially true later in the quarter as assignments get more challenging and office hours get busier; to make sure everyone can get help, they won’t have time to do that.

To get the most out of office hours, come prepared. Please come with specific questions or issues: i.e., not “I don’t know what to do.” or “Where do I start?”. And be ready to tell us what you have tried to resolve them: e.g., you wrote some test cases to narrow down where a bug could be, you’ve added some printing to understand what your code was doing. This will make it much easier for our course staff to figure out what’s going on, and they’ll be grateful.

Ultimately, though, none of us are psychic: it’s absolutely possible there may be an issue with your code which we cannot figure out. The person who is in the best position to understand it is the person who wrote the code and has been staring at it for hours: i.e., you. Our job is to give you the tools you need to get to that understanding.

6 Resources

Textbook

For some of the topics we will discuss, we will follow the draft of a textbook I am writing for this class; the tentative schedule later in this document details how lectures and textbook chapters line up.

For the most part lectures are intended to be standalone, with the textbook as a supplement for students who want to reinforce their learning or go deeper into a topic. Some topics, however, will not be covered directly in lecture; you will therefore need to read the relevant textbook chapter to be able to understand what follows in class. These topics will be announced in advance.

In addition, if you are looking for a reference, Cormen, Leiserson, Rivest, and Stein’s Introduction to Algorithms is the standard one. It is very comprehensive, but not always easy to approach. Its emphasis is also more on the theoretical side of things than this class’s. Nonetheless, it is a useful book to have on hand for computer scientists.

Supplementary Videos

A variety of supplementary videos about various 214-related topics are available on Canvas (on the page of that name). These have been produced by (former) members of the course staff, and provide additional information and advice that we don’t have time to cover in lectures. We recommend watching them early on; they offer valuable advice that can be quite helpful when working on assignments.
7 Assessment

This class uses a form of Specifications Grading (Nilson 2015). In particular, it features mainly qualitative assessment, and does not rely on points, percentages, and weighted averages to determine final grades.

This is likely quite different from what you have seen in your other classes, and will take some getting used to on your part. This system, however, has a number of benefits for students—see section 7.6.

7.1 Programming Assignments

This class features five programming homeworks and a final project that has a programming component, which chiefly contribute to our functional correctness learning objective. All programming assignments and related sub-assignments (see below) must be done individually; see our Academic Integrity policy later.

When evaluating your programming submissions, we will give you feedback regarding where they fell short as well as assign the following possible outcomes:

• **Got it**: your submission implements the specification correctly and in full.
• **Almost there**: your submission implements the specification with minor mistakes or omissions.
• **On the way**: your submission has elements of a correct solution, but requires non-trivial further work to be correct and/or complete.
• **Not yet**: your submission requires significant further work.
• **Cannot assess**: we could not evaluate your submission as is; repairs are required before we can do so.

Specifics of what is required to achieve each outcome are included in the handout for each assignment; our expectations are stated up front, no surprises.

Resubmissions

Making mistakes and correcting them is a natural part of the learning process. An outcome of On the way or Not yet is NOT a failure: it just means you still have things to learn from this assignment. Our assignments are intentionally challenging and our grading is intentionally strict in order to maximize your learning; it’s perfectly normal to not get them entirely correct on the first try.

To give you the opportunity to incorporate our feedback and get credit for your improved learning, we will allow you to resubmit the code portion of each homework one week after the initial deadline.

To determine final grades, we will use the best of the two outcomes your submissions have achieved. Given the scale and importance of the project, we will give you two opportunities to resubmit it instead of just—one week and two weeks after the initial deadline—and consider the best of the three outcomes.

Resubmissions are optional: if you’re happy about your first submission, you can skip the second.

Self-Evaluations

Each of the five programming homeworks will be followed by a self-evaluation, due one week after the initial deadline. Self-evaluations will consist of a few questions about the code you turned in.

The purpose of these self-evaluations is to evaluate your submissions on non-functional correctness: desirable aspects of programs that go beyond strictly producing correct answers. Aspects like efficiency, testing, robustness, or factoring. In actual programming (i.e., not in classes), these are just as important (if not more so) than raw functional correctness. This class will give you opportunities to cultivate these skills and habits as you work on assignments.

---

1See schedule at the end for exact dates.
2Specifically, the code you turned in for the first submission, not the resubmission, and not work in progress towards the resubmission. You can download your original submission from Canvas if you overwrote/misplaced it.
Project Report

Instead of a self-evaluation, the final project will include a written report where you will explain your design and provide rationales for your decisions. More details are included in the final project handout.

The final project and its associated report contribute to our evaluation and integration learning objectives.

7.2 Worksheets

To evaluate specific aspects of our theory learning objective, we will have two worksheet-style assignments.

To ensure everyone has a solid theoretical foundation, these worksheets will be done as Canvas quizzes with unlimited retries; we encourage you to re-do them until you fully understand.

7.3 Exams

We will have two in-person, on-paper midterm exams which will chiefly assess our theory learning objective. No notes or electronics (laptops, calculators, tablets, phones, smart watches, tamagotchis, etc.) will be allowed during exams. Specifics and logistical details will be announced leading to each exam.

7.4 Final Grades

Final letter grades will be determined as follows.

To help you keep track of your progress, please see the self-serve final grade worksheet on Canvas.

Base Grade

Your base grade will be determined based on your performance on homeworks and the final project:

<table>
<thead>
<tr>
<th></th>
<th>A- Excellent</th>
<th>B Good</th>
<th>C+ Satisfactory</th>
<th>D Poor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homeworks # Got it</td>
<td>all 5</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td># Almost there*</td>
<td>—</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td># On the way*</td>
<td>—</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td># not counted</td>
<td>—</td>
<td>—</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Final project Minimum outcome</td>
<td>Got it</td>
<td>Almost there</td>
<td>On the way</td>
<td>On the way</td>
</tr>
</tbody>
</table>

* or better

For example, a student whose homework submissions earned 4 Got it but one On the way and whose project submissions earned an Almost there would earn a base grade of B.

As another example, a student whose homework submissions earned 2 Got it, 2 On the way, 1 Not yet, and whose project submissions earned an On the way would earn a base grade of D.

Failing to meet the criteria for a D will result in a base grade of F.

Warning: these criteria put a lot of importance on the final project, which is intentional. We believe that the learning experiences from the project are especially valuable. We therefore consider it very important, and we want you to do so as well.

Modifiers

This base grade will then get adjusted with modifiers based on your performance in other aspects of the class. Each of these can either increase your grade by a partial letter grade, leave your grade unchanged, or decrease your grade by a partial letter grade. There are a total of six possible modifiers:

---

3 And past students agree!
Increase | Unchanged | Decrease
--- | --- | ---
Worksheets | Both 100% | Both ≥80% | Either <80%
Self-evaluations | # 5/5s | 4 | — | —
 # ≥ 3/5s | — | ≥3 | — | ≤ 2
 # ≤ 2/5s | ≤ 1 | ≤ 2 | ≥ 3 | —
Exam #1 | TBA | TBA | TBA | TBA
Exam #2 | TBA | TBA | TBA | TBA
Project program | Above expectations* | Meets expectations | N/A
Project report | Outstanding | Minor errors/omissions | Major errors/omissions

* Meaning: higher outcome than required for your base grade.

Both worksheets are considered together to determine a single modifier; both must meet the expected level. Similarly, all five self-evaluations are considered together to determine the self-evaluation modifier. Each exam, however, has its own modifier.

For example, suppose our hypothetical student who earned a base grade of B got everything right on both worksheets (Increase), submitted self-evaluations that earned three 5/5s and two 2/5s (Unchanged), did poorly on exam 1 (Decrease) but well on exam 2 (Unchanged), and wrote an excellent project report (Increase). This would work out to an increase of one partial letter grade, for a final grade of B+.

Alternatively, suppose our other hypothetical student who earned a base grade of D did well on exam 1 (Unchanged) and great on exam 2 (Increase), earned a Got it on the project (above expectations for a D base grade, so Increase) while having Increases for both worksheets and the project report, and Unchanged for self-evaluations, this would result in an increase of four partial letter grades. Starting from a base grade of D, our hypothetical student would earn a final grade of B-.

For reference, the possible letter grades at Northwestern are: F, D, C-, C, C+, B-, B, B+, A-, and A. Going up one partial letter grade means going forward one step in this sequence, and going down one letter grade means going backwards one step.

The letter grade sequence “saturates” on either end: applying four + modifiers to a base of B results in an A (there is no A+), and applying two - modifiers to a base of D results in an F (there are no D-, F+, or F-).

### 7.5 Late Policy

Unless otherwise indicated assignments are due by 11:59pm on their due date.

To accommodate everyday slippage and minor life hiccups, each student starts the class with three late tokens. Each of these late tokens can be exchanged for a two-day no questions asked extension to any (non-exam) assignment, including resubmissions. Only one token can be used per submission; i.e., two days is the maximum extension possible. Late tokens will be used automatically when you submit late; you don’t need to reach out to us and ask for permission.

In the event a student runs out of tokens, I’ll be in touch. Being occasionally late is perfectly fine. Being consistently late, on the other hand, may be a sign of deeper underlying issues: let’s talk so we can figure out the best way to help you.

Barring extreme circumstances, we will not grant further ad-hoc extensions or accomodations. If you do end up in a situation that would warrant additional flexibility, you must contact your dean of students

---

4Good job, hypothetical student!
5Solid recovery, other hypothetical student!
6Joe Holtgreive (jjh@northwestern.edu) for McCormick students, Liz Fekete Trubey (eft@northwestern.edu) for Weinberg students, and others for students in other schools.
7.6 Why this system?

This assessment system carries many benefits to you as a student.

- **Learning and grades align.** Learning is our true goal here; grades are (at best) a distraction. But given the impact grades can (unfortunately) have beyond the classroom, ignoring them altogether may not be an option for everyone. As a compromise, this system ties grades directly to concrete achievements and milestones in your learning, rather than on an amorphous pool of fungible points of assorted provenances.

- **You are in control.** You decide what grade you want to aim for, and the expectations you’ll need to meet for that goal are stated up front and transparently. Grades are determined solely by your achievements, not by those of your classmates; no percentage cutoffs, curves, or other such nonsense. No more "hoping" for a grade, and no more surprises when letter grades come: you can always tell exactly how things are going, and know what you need to do if you want to change it.

- **High standards, low stakes.** You’re all super smart. Yes, that means you too. You all have the potential to learn a lot in this class and grow as computer scientists. To inspire you to learn as much as you can, we will hold you to high standards; one doesn’t learn much by consistently producing meh work. At the same time, we recognize that high standards can carry a lot of pressure; which is why we’re also including mechanisms to lower stakes whenever possible (see next item).

- **There is room for mistakes.** Making mistakes is part of the learning process, and you should get credit for learning from them. Resubmissions give you an opportunity to do so. And for places where resubmissions don’t make sense (e.g., exams, self-evaluations), some slack is built-in to the grading standards to allow for small slips, and the consequences of even larger slips are bounded.

- **Flexibility built-in.** Life happens; we recognize that. But not all students feel comfortable asking for flexibility when necessary, either as a matter of personality or of upbringing. For this reason, "hiding" flexibility behind explicit requests is not equitable. Instead, this system comes with built-in flexibility in a number of places—resubmissions, late tokens, many possible modifiers, etc.—all of them with no questions asked and no need to ask for it. That way everyone benefits, regardless of comfort level.

**Caveats**

That being said, this is a large class—which introduces issues of scale and equity—and we’re limited by the length of the term—which introduces issues of scheduling. This means we’ve unfortunately had to make some compromises when designing our assessment strategy.

In an ideal world, we would assess your learning directly using mechanisms like oral exams and code walkthroughs, with infinite retries and no hard deadlines. But we are not in an ideal world, so we’re forced to assess your learning indirectly through your work using automated testing, written exams, and other mechanisms geared towards efficiency. These mechanisms, while sadly necessary in our context, may not reflect your learning 100% accurately.

Therefore, it is your responsibility to make sure the work you submit is as close a reflection of your actual learning as possible, so we can get the best picture we can of where you’re at as a student. And in turn, so you can get as much recognition as possible for your learning. Concretely, this means:

- For programming assignments, being thorough in your testing to avoid silly mistakes, and taking full advantage of our feedback when working on resubmissions.

- For "one-shot" assessments like exams or self-evaluations, be very careful to read instructions completely and thoroughly, and ask for clarifications if need be. We can’t assess your learning if you answer the wrong question.
8 Software

Code examples and programming assignments will use the DSSL2 (Data Structures Student Language version 2) language. It runs on top the Racket environment. You will need to install version 8.7 from:

download.racket-lang.org

If you have an old version from a previous course, some code may not work.

Then, to install DSSL2 proper, from DrRacket open the File menu and select Install Package..., then type dssl2 as the source. Then click Install. When it’s done, the Install button will change to Update, indicating that the package is installed.

To familiarize yourself with the language, you should consult the DSSL2 reference:

https://docs.racket-lang.org/dssl2/

8.1 Development Environment

The DrRacket IDE (which comes with Racket, and which you may have used in 111) has the most complete DSSL2 integration of any environment; that’s the one I encourage you to use. If you’re already familiar with other environments, varying levels of DSSL2 support are also available for:

- **VSCode**: your colleague Joshua Irvin wrote a DSSL2 plugin for VSCode[^1]. I have not tried it, though, so *caveat emptor*.

- **Vim**: Vim users are welcome to try your colleague Marko Vejnovich’s DSSL2 plugin[^2]. I have not used it myself either.

- **Emacs**: Emacs’s Python mode offers adequate, but far from ideal DSSL2 support.

You’re welcome to use these alternative environments if you wish, but neither I nor the course staff will be able to offer you support. You’ll be on your own.

8.2 Hardware

By default, we assume that you will be using your personal computer to work on assignments. If need be, you should also have physical access to the Wilkinson Lab, as well as login access to the computers there.

If you cannot access these computers, please reset your password at:

https://selfserv.eecs.northwestern.edu/temp_password/

If that does not work, please contact root@eecs.northwestern.edu. Similarly, contact root if you do not have physical access to the labs.

9 Academic Integrity

The trust employers, colleagues, and the public put in a Northwestern degree is built on the premise that our grades are an accurate reflection of student learning and effort. Any attempt at subverting the relationship between learning and grades is a threat to this hard-earned trust and, left unaddressed, hurts all Northwestern students. As such, we take any attempts at violating academic integrity very seriously.

In this class, we expect all students to know, understand, and abide by the McCormick Academic Integrity policy[^3], as well as the course-specific policies below. Failure to abide by these policies is a serious offense which carries serious consequences.

Fundamentally, anything you turn in for this class must be your own individual work, and be the product of your own thinking and reflection. There are two main ways in which your work may fall short of this expectation: accessing unauthorized sources and engaging in excessive collaboration.

[^1]: https://github.com/Gamefreak130/dssl2_vscode_extension
[^2]: https://github.com/markovejnovic/vim-dssl2/
[^3]: www.mccormick.northwestern.edu/students/undergraduate/academic-integrity.html
9.1 Unauthorized Sources

Sources which you are allowed to use while working on assignments:

- Materials posted to this course’s Canvas or Piazza this quarter.
- Interactions with members of the course staff, be they in office hours, on Piazza, etc.
- Feedback you receive from us on your work, for example in homework grading reports.
- That’s it.

In contrast, sources which you are not allowed to use or even consult while taking this class:

- Submissions or work in progress from other students, past or present, in full or in part. This includes test cases; those are part of submissions.
- Solutions or solution fragments you may find online or elsewhere.
- Course materials from previous quarters, regardless of how you obtained them.
- Outside tutors or "work-for-hire" services.
- Code generation tools (AI-based or otherwise), such as Github CoPilot, ChatGPT, etc.
- This list is not exhaustive; ask the instructor before using any source not explicitly listed above.

Using, or even having access to, unauthorized sources constitutes a violation of this class’s academic integrity policy, regardless of the extent to which the source was actually used, or even if it was used at all. Similarly sharing, posting, or otherwise making available your own solutions (in full or in part) or any course materials (homeworks, exams, solutions, test cases, etc.) is also a violation of our academic integrity policy. This extends even after the quarter ends; course material remains private information which you may not share or reproduce.

9.2 Excessive Collaboration

Collaboration is a good thing, and something you’ll be expected to do in the workplace. However, because this is an introductory class, we also need to ensure that every student has a solid foundation, which will allow them to be effective and productive collaborators throughout their career. For this reason, students must tread very carefully when collaborating in this class, to ensure they do not (accidentally or willfully) cross the line into excessive collaboration, with all the negative effects on learning this implies.

When working with current or former students in this class, you are limited to arms-length collaboration. When collaborating at arms length:

- You may not read, write, look at, record, or in any way transcribe solution elements from someone else. In turn, your collaborator may not read, dictate, or otherwise share elements from their own (or anyone else’s) solution with you, nor can they directly read, write, or look at your own solution either.
- You may not have your own solution on your screen or otherwise in your field of vision while collaborating, and neither must your collaborator.
- You must cite any collaborators whose ideas affected your work, and explain the effect the collaboration had on your solution. Bear in mind, however, that citing a collaboration does not absolve you from the aforementioned limits on arms-length collaboration.

To avoid getting close to the line, we strongly recommend you keep any discussions with current or former students either at a conceptual level, or about non-graded aspects of the class (e.g., examples from course materials).

You are of course always welcome to seek help from any member of the course staff on assignments; the above limitations do not apply to such interactions.

As always when in doubt, ask the instructor before doing something you’re not sure about.
9.3 Suspected Violations

Any suspected violation will be reported to the McCormick Office of the Dean. The Dean will conduct an investigation to determine whether a violation did take place or not, which will involve meeting with you. Should you find yourself in this situation, be forthright and honest; lying or withholding information from the Dean is a further offense, and can make a bad situation much worse.

10 Welcoming Environment

I consider this classroom to be a place where you will be treated with respect. I welcome all students, and expect all of you to do the same. Together, we can create an environment where everyone feels welcome and can engage fully in our community.

Each student has something of value to contribute, especially in engineering disciplines where empathy, communication, and teamwork elevate our contributions to society; and lack thereof can lead to disaster. Individual differences can deepen our understanding of one another, the world around us, and our lifelong role as engineers.

(Credit: Adapted from statements by the ASEE and Prof. Emma DeCosta)

11 Miscellaneous Statements

The registrar has a lot of things they want me to tell you.

11.1 Accessibility Statement

Northwestern University is committed to providing the most accessible learning environment as possible for students with disabilities. Should you anticipate or experience disability-related barriers in the academic setting, please contact AccessibleNU to move forward with the university’s established accommodation process (e: accessiblenu@northwestern.edu; p: 847-467-5530). If you already have established accommodations with AccessibleNU, please let me know as soon as possible, preferably within the first two weeks of the term, so we can work together to implement your disability accommodations. Disability information, including academic accommodations, is confidential under the Family Educational Rights and Privacy Act.

11.2 COVID-19 Expectations for Students

Students, faculty and staff must comply with University expectations regarding appropriate classroom behavior, including those outlined below and in the COVID-19 Expectations for Students. With respect to classroom procedures, this includes:

- Policies regarding masking, social distancing and other public health measures evolve as the situation changes. Students are responsible for understanding and complying with current University, state and city requirements.

- In some classes, masking and/or social distancing may be required as a result of an Americans with Disabilities Act (ADA) accommodation for the instructor or a student in the class even when not generally required on campus. In such cases, the instructor will notify the class.

If a student fails to comply with the COVID-19 Expectations for Students or other University expectations related to COVID-19, the instructor may ask the student to leave the class. The instructor is asked to report the incident to the Office of Community Standards for additional follow-up.
11.3 Exceptions to Class Modality

Class sessions for this course will occur in person. Individual students will not be granted permission to attend remotely except as the result of an Americans with Disabilities Act (ADA) accommodation as determined by AccessibleNU.

*Maintaining the health of the community remains our priority. If you are ill for any reason, do not attend class.*

Students who are ill, must quarantine, or experience a personal emergency should contact the instructor as soon as possible to arrange to complete coursework.

Should public health recommendations prevent us from holding class in person on a given day, the instructor or the university will notify students.

11.4 Recording Policy

This class or portions of this class will be recorded by the instructor for educational purpose and available to the class during the quarter. Your instructor will communicate how you can access the recordings. Portions of the course that contain images, questions or commentary/discussion by students will be edited out of any recordings that are saved beyond the current term.

Unauthorized student recording of classroom or other academic activities (including advising sessions or office hours) is prohibited. Unauthorized recording is unethical and may also be a violation of University policy and state law. Students requesting the use of assistive technology as an accommodation should contact AccessibleNU. Unauthorized use of classroom recordings – including distributing or posting them – is also prohibited. Under the University’s Copyright Policy faculty own the copyright to instructional materials – including those resources created specifically for the purposes of instruction, such as syllabi, lectures and lecture notes, and presentations. Students cannot copy, reproduce, display, or distribute these materials. Students who engage in unauthorized recording, unauthorized use of a recording, or unauthorized distribution of instructional materials will be referred to the appropriate University office for follow-up.

11.5 Support for Wellness and Mental Health

Northwestern University is committed to supporting the wellness of our students. Student Affairs has multiple resources to support student wellness and mental health. If you are feeling distressed or overwhelmed, please reach out for help. Students can access confidential resources through the Counseling and Psychological Services (CAPS), Religious and Spiritual Life (RSL) and the Center for Awareness, Response and Education (CARE). Additional information on all of the resources mentioned above can be found here:

https://www.northwestern.edu/counseling/
https://www.northwestern.edu/religious-life/
https://www.northwestern.edu/care/
# Tentative Schedule

Dates and content are subject to change. Any changes will be announced on Piazza.

<table>
<thead>
<tr>
<th>Date</th>
<th>Homeworks</th>
<th>Self-Evals and Resubmissions</th>
<th>Lecture Topic (and Reading)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thu 1/5</td>
<td>1 out</td>
<td></td>
<td>Intro, DSSL2 Basics</td>
</tr>
<tr>
<td>Tue 1/10</td>
<td></td>
<td></td>
<td>DSSL2 Q&amp;A, Linked Lists, Abstract Data Types, Stacks, Queues</td>
</tr>
<tr>
<td>Thu 1/12</td>
<td>1 due, 2 out</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mon 1/16</td>
<td></td>
<td>1 out</td>
<td>Asymptotic Complexity</td>
</tr>
<tr>
<td>Tue 1/17</td>
<td></td>
<td>1 due</td>
<td>Dictionary ADT (Chapters 7 and 8)</td>
</tr>
<tr>
<td>Thu 1/19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mon 1/23</td>
<td></td>
<td>2 out</td>
<td>Hash Tables (Chapter 9)</td>
</tr>
<tr>
<td>Tue 1/24</td>
<td></td>
<td></td>
<td>Graph ADTs and Representations (Chapters 10, 11, and 12)</td>
</tr>
<tr>
<td>Thu 1/26</td>
<td></td>
<td>2 due</td>
<td></td>
</tr>
<tr>
<td>Tue 1/31</td>
<td></td>
<td>4 out</td>
<td>Graph Search (Chapter 13)</td>
</tr>
<tr>
<td>Thu 2/2</td>
<td></td>
<td>3 due</td>
<td>Single-Source Shortest Path (Chapter 14)</td>
</tr>
<tr>
<td>Mon 2/6</td>
<td></td>
<td>3 out</td>
<td>First Midterm Exam</td>
</tr>
<tr>
<td>Tue 2/7</td>
<td></td>
<td>3 due</td>
<td>Priority Queue ADT (Chapter 15)</td>
</tr>
<tr>
<td>Thu 2/9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mon 2/13</td>
<td></td>
<td>4 out</td>
<td>Data Design</td>
</tr>
<tr>
<td>Tue 2/14</td>
<td></td>
<td>4 due</td>
<td>Data Design, continued</td>
</tr>
<tr>
<td>Thu 2/16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mon 2/20</td>
<td></td>
<td>5 out</td>
<td>Minimum Spanning Trees, Disjoint Sets ADT</td>
</tr>
<tr>
<td>Tue 2/21</td>
<td></td>
<td>5 due</td>
<td>Self-Balancing Trees</td>
</tr>
<tr>
<td>Thu 2/23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tue 2/28</td>
<td>project program due</td>
<td></td>
<td>Amortized Analysis</td>
</tr>
<tr>
<td>Thu 3/2</td>
<td>WARNING: project has Tue deadlines!</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tue 3/7</td>
<td></td>
<td>project program due</td>
<td>TBD</td>
</tr>
<tr>
<td>Thu 3/9</td>
<td></td>
<td>project report due</td>
<td>Second Midterm Exam</td>
</tr>
<tr>
<td>Tue 3/14</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>