Syllabus - CS 0359 Foundations of Algorithms and Data Structures

Explores foundational computer science concepts essential for graduate level coursework. Examines discrete mathematics principles, including probabilities and combinations, along with proof techniques such as induction. Addresses various algorithmic approaches including quadratic and divide-and-conquer sorting methods. Investigates data structures such as sequences, trees, graphs, and hashtables and introduces research methodologies relevant to computer science and related disciplines.

Performance-Based Admissions

CS 0359 is a self-paced module that focuses on demonstrating competency in introductory algorithms and data structures. By completing this course, you demonstrate that through a combination of previous experience *and* completing a detailed portfolio, you have the skill set needed to be successful in an MSCS program with Khoury College of Computer Sciences. It is important to note that completing the course does not guarantee admission (as other factors are often taken into account), but it provides evidence of prior learning and knowledge to improve your admission prospects.

Prerequisites

While there are no direct course prerequisites, it is assumed that students entering the course have the following backgrounds:

- Prior programming experience through bootcamps, self-taught knowledge, or on-the-job experience
- Previous bachelor's degree in any field
- Strong work ethic and motivation for self-paced learning using a variety of resources

Self-Paced / Competency Pathway vs. Align Pathway

This course focuses on students developing a portfolio to demonstrate learning and understanding of key topics. Resources are provided to help students gain the understanding necessary for demonstrating their knowledge, and students are encouraged to find additional resources as needed. This course is designed for people who already have practical knowledge, but lack the theoretical and foundational knowledge needed to be successful in the MSCS.

It is worth noting that if you don't have a programming background and/or you are looking for a more guided sequence with traditional assignments and courses, you may want to explore the Align Program. This is a high-contact, high-support program focused on guiding non-computer scientists to be successful in the MSCS.

Outcomes and Topics

By the end of the course, students should demonstrate an understanding of both theory and implementation of the following topics:

- Understand, discuss, and apply mathematical foundations for computer science
  - Number representation (binary, conversion, modular arithmetic)
  - Collections and sets
  - Methods of proof
  - Combinatorics and permutations
  - Probability
  - Series
  - Big O and growth of functions
  - Graph theory
- Sorting data with quadratic and divide-and-conquer sorts
- Analyze and implement fundamental data structures
  - Vectors
  - Linked lists
  - Binary trees
  - Heaps
  - Hash tables
  - Graphs
- Describe the foundations of of greedy algorithms
  - Implement Dijkstra's shortest path algorithm
- Describe trade-offs between data structures given context
- Implement programs utilizing the best data structure for the given context
- Analyze empirically and theoretically new data structures and algorithms

Course Schedule

As the course is self-paced, the primary focus is on the final portfolio and technical interview. However, due to Northeastern deadlines, the course *must* be completed at least 15 days before one of the admission deadlines (ideally sooner). This allows students the opportunity to apply for the program and ensures systems are updated to reflect course completion. If you miss the deadline, you will roll to the next deadline.

Target Semester	Khoury Admissions Deadline	Technical Interview **Completion** Deadline	Portfolio Submission Deadline
Fall	May 1	April 15	April 1
Spring (international students)	October 1	September 15	September 1

Target Semester	Khoury Admissions Deadline	Technical Interview **Completion** Deadline	Portfolio Submission Deadline
Spring (domestic students)	December 1	November 15	November 1

For many students starting CS 0359 in the fall, this will mean targeting a start semester of the following fall—so Fall 2025 starts will be targeting a Fall 2026 start date.

> [!IMPORTANT] You should not wait until the deadlines to submit, especially when your portfolio. If your portfolio gets returned with suggested updates and changes, this may mean you will miss the next deadline. Early completion is recommended.

Khoury Student Expectations

1. Respect should be shown in all communications and interactions with faculty, staff, industry, peers, and all others on campus. This includes respecting the preferred methods and response times of faculty and staff.
2. Students are to actively participate in course activities and discussion.
3. Any issues that arise should be communicated to the appropriate faculty or staff member proactively.
4. All course interaction including instruction, TA advising, and course activities are to be done in English.
5. Students should come with the goal of learning and have a "growth mindset".

Assessment

The course will be assessed with two major components

1. A portfolio
2. A technical interview

Both are detailed heavily in the canvas shell. Students should expect to spend multiple weeks putting together the portfolio, as it has five different parts from building a medium sized program to writing a technical paper. Students will be asked to program in a variety of languages to demonstrate competency in both systems level languages (C or Rust) or high level languages (C++, Java, Kotlin, Python, Rust, etc). They will be asked to write at a college level with a written report for each component.

Once the student completes the portfolio they may be asked to resubmit with changes, or they will be invited to setup a technical interview. The technical

interview will involve interviewing aspects of their portfolio and working on technical interview problems.

Students must complete **both** the portfolio and technical interview with the equivalent of a B or higher grade to pass the course. We will follow a mastery grading model, in which students will be given opportunities to retake various parts after feedback and a slight weighting period between attempts.

Course Materials

There are no required materials for this course beyond the materials provided in the course shell.

However, there are multiple resources on the course Resources Repository that you may find useful. We highly encourage you to look through them before the start of the course.

Github Classroom

Github Classroom is used in this course to manage programming assignments. While the details of each assignment will be made available on Canvas, students will be required to manage their assignments via commits to Github Classroom. Github is a website that used git, a professional version control system used by developers from all over the world to discover, share, and build software. Github Classroom requires a Github.com account.

MS Teams

Working in environments such as slack, discord, or MS teams is expected for most modern day programmers. It is expected students have MS Teams installed on their local system (don't just access through the browser). Please note, different instructors have different preferences for contact, and those preferences should be followed.

General Policies

Digital Etiquette (netiquette)

For much of what we do, even as face-to-face interactions become possible again, a primary form of communication in the modern world, especially in CS, is the written word. Because this means you are missing body language cues and immediate feedback from your "listener", it is very important to understand some common rules for good online etiquette: (adapted slightly from 7 Rules for Online Etiquette)

1. **Be respectful.** While it is easier to say hurtful or disrespectful things without standing face-to-face with someone, it is important to remember that your classmates and instructors are real people who are affected by

the words you say and write. It is essential to keep in mind the feelings and opinions of others. If you wouldn't say it to someone's face, you shouldn't say it online either.

2. **Be aware of strong language, all caps, and exclamation points.** It is easy for written text to be misread and misunderstood. Have you ever sent a text message with good intention but your recipient thought you were being rude? If so, then you've experienced this firsthand. By being aware of strong language, you can identify potential confusions before sending messages. Tip: Read everything out loud before you send it.
3. **Be careful with humor and sarcasm.** Certainly, you shouldn't avoid being funny. We love to see your personality shine through in our classes. Many of the instructional staff are exceptionally funny too. But make certain that it is clear you are being funny and not being rude. Emoticons and smileys can be helpful when conveying humor or sarcasm so that it is read correctly.
4. **Yes, grammar and spelling matter.** While texting, "textspeak" can b gr8 4 ur friends. In an educational setting (even online), however, keep it a bit more formal. Your written communication should be professional and reflect proper writing style. Save written shortcuts and less than stellar grammar for Snapchat if you must, but follow grammar rules for school.
5. **Don't post or share (even privately) inappropriate material.** Enough said here, nothing is truly private online.

Be forgiving. Remember that not everyone will know these rules before posting. Try to be understanding of others when they struggle with written communication. It is very different than simply talking to a person face-to-face.

Academic Integrity and Scholarship

Everybody is aware of vast amount of knowledge that is available on the Internet. While every assignment in this class is designed to be solvable using the methods we have discussed in class, your Professor recognizes that every student will end up using the Internet at some point or another. As a student, your job is to be honest and forthright with your efforts. It is of utmost importance to your learning that you never just cut-and-paste a solution to a homework problem; instead make the effort to understand the solution well enough to put it into your own words and be sure to cite your sources. Citations should include references (paper, website, or other) for any site that you used to research a solution. Ideally, proper APA or ACM format should be used. For websites this includes name of website, title of the article, the url, and the date of retrieval. If you find yourself spending any significant amount of time searching the web, you should speak with a TA or the Instructor, because it is a sign that something is not working for you in this class. If you ever feel a need to "copy code" from a web site, please think again; any code snippet should by very short and prefaced by inline comments stating where it came from and giving credit to the actual author.

> It is better to read and understand the code, close your web browser, reconstruct the ideas from memory, and then write your own version. Even then, give credit to the original author for helping you think it through.

You are expected to read, understand, and follow the University's policies on Academic Integrity. With the exception of explicit, group projects, such as Pair Programming (which we will teach), each student is expected to do his or her own work. Violations of academic integrity will result in a **negative grade** on the corresponding assignment, along with harsher penalties for more widespread problems, including automatic failing of a course. Academic integrity violations will be reported to OSCRR and the Khoury Academic Integrity Committee.

Copying code is cheating and lacks integrity. CMU provides some nice examples to follow:

For personal assistance, here are some of the things that are appropriate:

- Clarifying ambiguities or vague points in class handouts, textbooks, or lectures
- Discussing or explaining the general class material
- Providing assistance with the programming language, in using the system facilities, or with editing and debugging tools
- Discussing the code that we give out on the assignment
- Discussing the assignments to better understand them
- Getting help from anyone concerning programming issues which are clearly more general than the specific project (e.g., what does a particular error message mean?)
- Suggesting solution strategies
- In general, oral collaboration is OK.

Here are some things that are inappropriate:

- Copying files or parts of files (such as source code, written text, or unit tests) from another person or source
- Copying (or retyping) files or parts of files with minor modifications such as style changes or minor logic modifications
- Allowing someone else to copy your code or written assignment, either in draft or final form
- Getting help that you do not fully understand, and from someone whom you do not acknowledge on your solution
- Writing, using, or submitting a program that attempts to alter or erase grading information or otherwise compromise security
- Copying someone else's files containing draft solutions, even if the file permissions are incorrectly set to allow it
- Lying to course staff
- Reading the current solution (handed out) if you will be handing in the current assignment late
- Copying prose or programs directly

- Giving copies of work to others
- Coaching others step-by-step

If you do any of these, your should also acknowledge it in what you turn in; but expect to have a conversation with an instructor about it and, at least, suffer some penalty in the grade. If we discover you have done this and not acknowledged it, the penalty will be much more severe. In other words, dishonesty is much worse than stupidity.

Here are some gray areas:

- Reading someone's code for clarity or bugs, after you have completed your own
- Helping with debugging
- Looking at someone's prose or program but thinking about them and writing your own
- Following someone's advice or instructions without understanding them
- Many others

These, too, should be acknowledged.

A few resources

- Cheating versus Collaboration
- CMU Policy to acknowledge their examples above.
- Northeastern University Citation and Academic Integrity Checklist - this does apply to your code!
- OSCCR - Office of Student Conduct and Conflict Resolution
- Northeastern Academic Integrity Policy

The university's academic integrity policy discusses actions regarded as violations and consequences for students http://www.northeastern.edu/osccr/academic-integrity.

Generative AI (Claude, ChatGPTm and Copilot)

> [!CAUTION] Generative AI is a powerful tool, it's use is allowed, but use it wisely. See below for this course's policy, and not every course has a different policy based on the learning goals.

For this class, you are allowed the use of claude or similar product for the use of small code snippets, building rote methods, and tests. You are not allowed to use it write your entire assignment, and you will find that it will often try to redesign against the expected design. As such, we only recommend its use to help you better understand and learn syntax. If at any point we feel you are using generative AI to build your entire assignment or write your reports for you, this is considered plagiarism and will be treated as an academic integrity violation. Simply, it is a tool to facilitate your learning and efficiency in writing code. It should not be a substitute for understanding the concepts.

Disabilities; Reasonable Accommodations

Our goal is to ensure that every student should be able to participate in this course. Students with disabilities who wish to receive academic services and/or accommodations should visit the Disability Access Services (DAS) (previously known as DRS) or call (844) 688-6287. If you have already done so, please provide your letter from the DAS to the instructor – early in the semester – to arrange for those accommodations. If you do require any special accommodations, let the instructor know immediately so that appropriate details can be worked out.

Student Feedback

Your opinions are very important to the TAs and faculty. In addition to the university required student evaluation form, at the end of the semester, we will be asking for your feedback at least once about during the semester. However, if you have suggestions or concerns about the course, don't wait until you are asked. You can contact us any time! As a simple example, if my lectures are too fast, or too slow, or if I need to speak more loudly, let me know!

Academic Standing Process: Offering Additional Support

Given the unique academic structure of the Align program and reality that our students are coming from many diverse backgrounds with varying levels of math proficiency and CS exposure, it is important that policies and procedures are set in place that can support our students while maintaining academic standards set forth by the College. Also, during the first two semesters, the pace of the course is typically faster and truncated, which may present some of its own challenges.

Thus, the advising team and faculty work together as a Student Success Team to discuss students who could benefit from more support at the midpoint and end of the semester.

If there are concerns about a student's academic performance at the midpoint, an Academic Advisor will reach out to schedule a check-in and possibly recommend resources like dedicated tutoring or frequent check-ins. Although these are recommendations, note that it is a student's choice to accept or decline, as the goal is to empower students to make that decision based upon what's best for them.

The College has defined that a B grade or higher is required for students to move on, and we have found that the midpoint check-in serves as an effective way to intervene before it's too late.

As an additional measure, the Student Success Team will discuss the academic outcomes of students earning less than a B grade after final grades have been submitted and determine whether they can proceed to the subsequent course(s).

If you have any questions regarding the Academic Standing process, please contact your Academic Advisor.

Title IX

Title IX of the USA Education Amendments of 1972 protects individuals from sex or gender-based discrimination, including discrimination based on gender-identity, in educational programs and activities that receive federal financial assistance.

Northeastern's Title IX Policy prohibits Prohibited Offenses, which are defined as sexual harassment, sexual assault, relationship or domestic violence, and stalking. The Title IX Policy applies to the entire community, including male, female, transgender students, faculty and staff.

Faculty members are considered "responsible employees" at Northeastern University, meaning they are required to report all allegations of sex or gender-based discrimination to the Title IX Coordinator.

The university offers confidential resources for medical treatment, emotional support and counseling through Confidential Employees. Confidential Employees are not required to disclose information about Prohibited Offenses to the Title IX Coordinator without prior consent of the student. Confidential Resources on campus include University Health and Counseling Services (UHCS) staff, Sexual Violence Resource Center (SVRC), Office of Prevention and Education and the Center for Spiritual Dialogue and Service (CSDS). [From Title IV Policy, Section III.C]

Alleged violations can be reported to the Title IX Coordinator within The Office for University Equity and Compliance at: titleix@northeastern.edu and/or through NUPD (844) 688-6287.

Reporting Prohibited Offenses to NUPD does NOT commit the victim/affected party to future legal action.

In case of an emergency, please call 911.

Please visit The Office for University Equity and Compliance for the full Title IX Policy, a complete list of reporting options and resources both on and off-campus.