

Syllabus

Textbook

R. Sedgewick and K. Wayne, Computer Science: An Interdisciplinary Approach, Addison–Wesley Professional, 2016. ISBN 978-0134076423. We will be referencing this text all semester. The lectures are based on its contents. The hardcopy version of the textbook is available from: [Labyrinth Books](#).

Course Organization

This course includes lectures, class meetings, labs, precepts, programming assignments, a final project, and four (4) exams, as outlined below.

Lectures. The first “lecture” is a live kickoff meeting in McCosh 50 on Wednesday, September 1 at 12:30. After this, remaining lectures are provided as studio-produced videos. You are expected to watch lecture before lab and precept so that you are well-prepared. See the [Schedule](#) page.

Class Meetings. Class meetings are scheduled on various Mondays and Wednesdays throughout the semester, at 12:30. Some of these meetings are used for exams (mandatory, rooms to be announced) while others will be used for practice exams and to provide guidance on assignments (optional, McCosh 50).

Labs. Labs meet Tuesdays and Thursdays. Each lab is a self-paced section, and involves working with a partner on interactive exercises and problem sets. Each lab is supervised by an undergraduate facilitator to answer your questions. You are required to submit your labs prior to precept.

Precepts. Precepts meet Fridays. Working within a small group, your preceptor will review key learning objectives for the week’s lectures and lab exercises. The precept will involve question and answer, group discussion, etc.

Programming Assignments. There are ten (10) programming assignments, all equally weighted. You may drop one one programming assignment. However you may not drop the first programming assignment.

- **Late Work.** Assignments are due on Sunday nights at 11:59pm. Submissions made at 12:00am will be marked late. If any part of an assignment is submitted late, the entire assignment will be marked late. To submit an assignment late, mark the checkbox on our online submission system (i.e., TigerFile) that indicates that your work is incomplete. If the deadline has passed and the checkbox is not marked, we will grade submitted work as-is.
- **Penalties for late work.** Late assignments are assessed a penalty equal to 20% of the possible points on the assignment per day or partial day late. We realize that students may have to travel, observe holidays, etc. So, the penalties for your first four late days are automatically waived. Additional late penalties will be waived only in the case of unforeseeable circumstances such as medical emergencies. Please consult with your Director of Studies or Dean, who may provide a recommendation to us. Notes from McCosh cannot be accepted for late work.

Final project. The final project is a large, open-ended programming project, approximately the length of two programming assignments. We encourage you to work with a partner. It is due on Dean’s Date at 5:00pm. You may not use late days on the final project. We will not accept any work after Dean’s Date without a Dean’s recommendation.

Exams. There are two two-part exams during the semester. The exams have a written portion and a programming portion. There is no final exam. No makeup exams will be considered without a Dean’s recommendation and our approval in advance.

Regrades. If you feel a mistake has been made grading your work, please use the regrade request features of codePost (programming assignments, programming exams) or GradeScope (written exams). Regrade requests must be submitted within three (3) days of your work being returned.

Final Grades

Your final grade will be calculated based on a weighted sum of all the labs, assignments, final project and exams. The final letter grade cutoffs will be determined at the end of the semester based on overall class performance, but not curved to any specific distribution. Staff discretion **may** be applied to adjust borderline cases - for example, answering student questions on Ed Discussion can be used to bump borderline grades.

Grade Contributions	Weight
Programming Assignments	40%
Final Project	10%
Lab Exercises	5%
Programming Exam 1	7.5%
Written Exam 1	15%
Programming Exam 2	7.5%
Written Exam 2	15%

Again, there is no curve: you must earn, at a minimum, 93.0% of the available points for an A, 90.0% for an A–, 87.0% for a B+, 83.0% for a B, 80.0% for a B-, 77.0% for a C+, 73.0% for a C, 70.0% for a C-, or 60.0% for a D. You will fail the course if you earn below 60.0% of the available points.

Collaboration Policy

This course permits (and encourages) many forms of collaboration, including with course staff, classmates, and lab TAs. However, you must be careful to collaborate only as authorized below. Here is an executive summary:

	your partner	course staff	COS 126 grads	classmates	other
discuss concepts with ...	yes	yes	yes	yes	yes
acknowledge collaboration with ...	yes	yes	yes	yes	yes
show your code/solutions to ...	yes	yes	yes	no	no
view the code/solutions of ...	yes	no	no	no	no
copy the code/solutions from ...	yes	no	no	no	no

Your solutions. You must individually compose all of your solutions. The term solutions refers to any of the products created when completing a programming assignment, such as source code (including comments) and the readme.txt file. It includes both finished and unfinished products, regardless of correctness or completeness.

- You must never expose solutions to anyone who is taking COS 126 now or who might take COS 126 in the future. Do not post solutions in public code repositories.
- You must never receive or view someone else’s solutions to a programming assignment (or variant of an assignment).

All the rules above continue to apply after assignments are graded and after the end of the semester.

Collaboration with course staff. You are welcome to discuss your solutions with course staff members (instructor and preceptors) in office hours or via private Ed Discussion posts. Do not post or email us your code; instead, submit it via TigerFile - the course submission system.

Collaboration with classmates. We encourage you to discuss common concerns with classmates either private via personal interactions or publicly in the Ed Discussion course forum. These discussions must be kept at a general level, without exposing your solutions.

For example, you may discuss:	But, you may not:
How to interpret an error message. For example, “What does StackOverflowError mean?”	Look at another classmate’s solutions (or show another classmate your solutions).
How to use a feature in IntelliJ. For example, “How do I start the embedded terminal?”	Lead a classmate step-by-step through any part of the assignment (or allow a classmate to lead you step-by-step through any part of the assignment).
Clarifications for the lecture videos, lecture slides, textbook, old exam questions, or the assignment specifications.	

Collaboration with students who have already taken COS 126. Anyone who successfully completed COS 126 in a previous term, including lab TAs and peer tutors, may assist you in the following ways:

- They may look at your code.
- They must not write, type, or dictate code.
- They may help debug, but they must not give step-by-step directions on how to complete the assignment.
- They should give thought-provoking suggestions rather than dictating what to change.
- While helping you, they must not look at any other solutions, including their own.

Collaboration with a partner. On certain assignments the collaboration policy is relaxed to allow working with a partner, subject to the following rules:

- Choose a partner whose skill level is close to your own.
- Complete all work with your partner, sharing the same screen. (Zoom supports this!) This includes debugging, testing, commenting, writing the `readme.txt`, and submitting the files.
- Go to office hours or the lab with your partner; one partner may not edit the code without the other partner present.
- Use the following pair programming protocol: one partner, the driver, types the code and the other partner, the navigator, reviews the code, identifies bugs, and asks questions. Swap roles every thirty (30) minutes.
- Do not post your code where it is visible to anyone but you and your partner. Code can be shared using Google Drive, Dropbox, etc.
- To dissolve a partnership, you must contact the course administrator.

Collaboration acknowledgment. You must acknowledge all collaboration (except for partner names on group assignments) in the `readme.txt` file for that week’s assignment. Please include the names of lab TAs, preceptors, etc.

Plagiarism

Plagiarism and abetting plagiarism are serious academic infractions. Programming is a creative work and the academic regulations that apply to plagiarizing prose also apply to plagiarizing code. Princeton’s [Rights, Rules, Responsibilities](#) defines plagiarism as “the use of any outside source without proper acknowledgment.” It ranges from “verbatim copying” (e.g., cutting-and-pasting code) to “thorough paraphrasing” (e.g., changing variable names or rearranging code).

Outside sources and citations. Copying or adapting code that is not yours is permitted only if it comes from the course materials (i.e., the course textbook, companion booksite, programming assignment specifications, checklists, lecture slides, lecture videos, labs and precepts). If you do so, you must cite any code that you copy or adapt (with the exception of code that is included with the assignment).

Plagiarism detection. We use sophisticated software tools to detect plagiarism. For example, we compare each submission (including interim submissions) against a code database, which includes all submissions from this offering and previous offerings of the course. While we take no pleasure in bringing cases to the Committee on Discipline; it is our duty to keep the playing field level for the overwhelming majority of students who work very hard in this course and follow the rules.

Penalties. We refer alleged academic violations (including plagiarism and abetting plagiarism) to the Committee on Discipline. If found responsible, the typical penalty is an F as a course grade plus whatever penalty that the CoD imposes. (The typical CoD penalty for plagiarism is suspension from the University for at least one semester.) Violators of course policies that are not adjudicated by the CoD will receive penalties based on the severity of the violation, ranging from a warning (for violations that are both unintentional and innocuous) to an F in the course (for violations that are both intentional and serious).

Published with [Wowchemy](#) — the free, [open source](#) website builder that empowers creators.