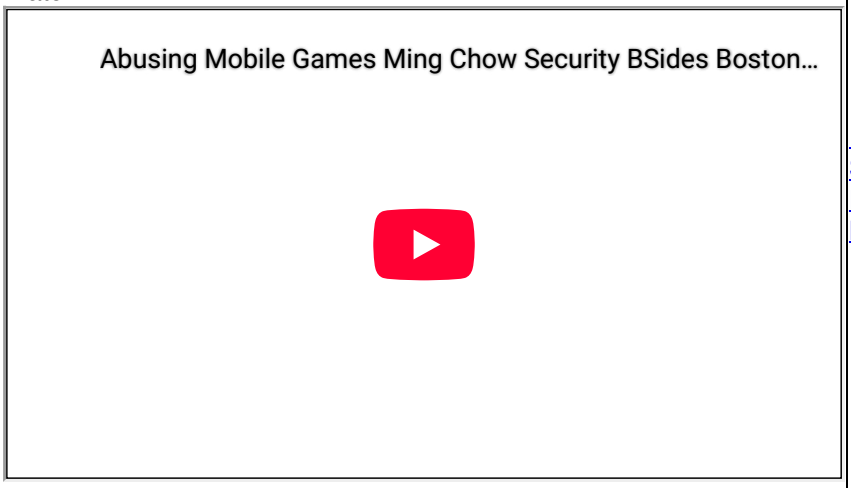
Thursday, October 15th		
Tuesday, October 20th	Project 1 Due	
Thursday, October 22nd		• Tufts Hackathon: Friday, October 23rd - Saturday, October 24th
Tuesday, October 27th	• Feedback on Game Project 1 • Game Testing • Personal Engagement Projects	Project 2 Assigned
Thursday, October 29th		
Tuesday, November 3rd		
Thursday, November 5th		
Thursday, November 12th		
Tuesday, November 17th	• Ethics, MMORPGs, and Securing Online Games • Abusing Mobile Games (my presentation at the BSides Boston Security Conference 2014) • Watch: 	IEEE Security & Privacy Securing Online Games Readings. Due on Tuesday, November 24th in class
Thursday, November 19th		
Tuesday, November 24th		
Tuesday, December 1st		
Thursday, December 3rd		

Tuesday, December 8th		
Thursday, December 10th		

Grading

- Group Project 1: 20%
- Group Project 2: 55%
- Labs and exercises: 15%
- Class participation: 10%
- Personal Engagement Project: 10%

Course Policies

Assignment Late Policy

Assignments (labs, homework) that is submitted electronically (most homework) are due at 11:59 PM on a Tuesday or Thursday. We will grant an automatic extension of ten minutes at no cost to you. If you plan on submitting your work at midnight or at six, you will have nine minutes for last-minute changes.

An assignment is expected to be submitted on time. However, we recognize that the exigencies of college life occasionally interfere with on-time submission. If you have difficulty getting the assignment in on time, you have two options:

1. For ordinary difficulties, each student is automatically issued three (3) "extension tokens." By expending an extension token, you can get an automatic 24-hour extension on all deadlines associated with a single assignment. To use an extension token, you must e-mail me at **`mchow@cs.tufts.edu`**. Thus must be sent before the assignment is due. At most two extension tokens may be expended on any single assignment. When you are out of tokens, late assignments will no longer be accepted: it will be returned ungraded, and you will receive no credit for the work.
2. If a serious illness affects your ability to complete the assignment on time, your first step is to report the illness using the "Illness Notification Form" that is available in WebCenter for Students. We will make suitable arrangements. For extraordinary difficulties, such as bereavement, family emergencies, or other extraordinary unpleasant events, your first step should be to make contact with your associate dean for undergraduate education. You must take this step before the assignment is due. Ask your dean to drop me an email or give me a call, and we will make special arrangements that are suited to your circumstances.

Please understand that extension tokens are meant to be used. That is, you will not receive any special bonus at the end of the course if you do not use any of your extension tokens.

Labs

A lab is due one week from the day it is assigned.

Solutions to Assignments and Examinations

Solutions to assignments and examinations will not be posted for this course.

COMP 150-SEN

Software Engineering Foundations, Spring 2019

Staff

Name	Office	E-mail	Office Hours (also available by appointment)
Jeff Foster	211 Halligan	jfoster@cs.tufts.edu	MW 1:15-2:00pm
Tomoki Shibata	Wed: Halligan extension 007; Fri: Halligan 209	tshibata@cs.tufts.edu	W 4:15-5:15pm, F 9:30-10:30am

Information

Location	Bromfield-Pearson Room 002
Time	MW 3:00-4:15
Midterm	March 13, in class
Final	May 3, 3:30-5:30pm
Textbooks	There are no required textbooks

Description

The scale of modern software systems is truly amazing. We regularly use software that is hundreds-of-thousands to millions of lines of code, and that software, while certainly not bug-free, mostly does what it is supposed to. Achieving this kind of scale has been the result of steady progress in *software engineering* over the last several decades.

In this class, we will study the foundations of software engineering, focusing on the core principles and ideas that enable us to build large-scale software systems. Our focus will be on the coding side of software engineering. Equally important, but deferred to a different course, is the people/team side of software engineering. We will explore ideas such as abstraction, modularity, architecture, specification, testing, and debugging, among others.

The course will be conducted in Java, and will include a short introduction to Java at the beginning. During the course, students will complete a number of programming assignments, and the course will also include some reading assignments.

Prerequities: COMP 40, graduate standing, or instructor consent. If you are not sure whether you meet the prerequisites for the course, please contact the instructor.

Schedule

Week	Monday	Wednesday
Jan 14	*(No class)*	*No class*
Jan 21	*MLK Day* No class	Introduction Java Reading Assignment #1
Jan 28	Java Project 1	Java LinkedList.java List.java

Feb 4	Java ArrayList.java	Java
Feb 11	Design Patterns Reading Assignment #2	Design Patterns Project 2
Feb 18	*President's Day* No class	(Feb 20) Modularity Discussion Design Patterns (Feb 21) Design Patterns ooc.c Reflection
Feb 25	Software Architecture Reading Assignment #3	Testing
Mar 4	Fuzz Testing Discussion Testing	Testing
Mar 11	Midterm Review	Midterm
Mar 18	*Spring Recess* No class	*Spring Recess* No class
Mar 25	Refactoring Project 3	Program Verification
Apr 1	*Guest Lecture:* Milod Kazerounian, Program Verification (cont'd)	*Guest Lecture:* Diogenes Nunez, Garbage Collection Reading Assignment #4
Apr 8	Security java-sec-example.tar.gz	No Silver Bullet discussion Project 4
Apr 15	*Patriots' Day* No class	Debugging How Failures Come to Be The Scientific Method Reproducing Problems
Apr 22	Simplifying Problems Fixing the Defect Reading Assignment #5	ACM Code of Ethics
Apr 29	What Makes a Great Software Engineer Final Exam Review	*(No class)*

Reading Assignments

1. (Due Jan 30) Tony Hoare, Null References: The Billion Dollar Mistake
   - It's okay to read the show notes instead of watch the video.
   - Q1: What was the most interesting thing you learned from the presentation?
   - Q2: Do you think null pointers really were a billion dollar mistake?
2. (Due Feb 20) D.L. Parnas, On the Criteria To Be Used in Decomposing Systems into Modules
   - Q1: Briefly describe one important lesson about modularity that the paper describes and that you think is still relevant today.
   - Q2: Computing has advanced significantly since this paper was written. Briefly describe one challenge (not necessarily with modularity) or perspective that, while realistic in 1972, does not apply today.
3. (Due Mar 4) Miller, Fredriksen, and So, An Empirical Study of the Reliability of UNIX Utilities
   - Q1: Briefly describe one strength and one weakness of fuzz testing.
   - Q2: Two modern fuzz testers are american fuzzy lop and libfuzzer. Pick one, look through the web page(s) for it, and briefly describe one way it improves on the original fuzz testing work of Miller et al.
4. (Due Apr 10) Brooks, No Silver Bullet: Essence and Accidents of Software Engineering
   - Q1: Briefly discuss one contrast Brooks draws between Software Engineering and other areas or science or engineering.
   - Q2: Brooks discusses several "Hopes for the silver", but his comments are about the state of the art in 1987. Describe one "hope for the silver" that you see today, but that Brooks did not see back then. Or, if you are a pessimist, explain why there is no hope.
5. (Due Apr 29) Li, Ko, and Zhu, What makes a great software engineer?
   - Q1: What was the most surprising characteristic to you of a great software engineering, as described by the paper?
   - Q2: What are some things you might want to do on your own, after this course ends, to become a great software engineer?

Projects

- Project 1 - Java ADTs and the Adapter Pattern
- Project 2 - Design Patterns in Java
- Project 3 - A Unit Testing Framework
- Project 4 - Java on Rails

Exams

- Sample questions for the midterm (solutions)
- Midterm sample solutions
- Final sample solutions

Syllabus

> Syllabus subject to change until the start of the semester

Prerequisites

COMP 40, graduate standing, or instructor consent. If you are not sure whether you meet the prerequisites for the course, please contact the instructor.

List of Topics (Tentative)

Below is an approximate list of topics for class. The exact topics will be determined based on the pace of the class.

- Java programming, including classes, objects, inheritance, interfaces, delegation, Java generics, and the Java runtime
- Abstract data types, modularity, information hiding
- Design patterns, including for concurrency
- Software architecture
- Program specification and verification
- Object-oriented refactoring
- Testing
- Debugging
- Program synthesis
- Special topics (TBA)

Office Hours and Web Forum

Office hours for the instructional staff will be posted on the course web page a few days into the semester.

While we will provide assistance with assignments during office hours, you are responsible for developing and debugging your own programs. Do not rely on the instructional staff to make your project work.

Important announcements will be made in class or on Piazza. Please make it a habit to check Piazza daily, and/or sign up to receive email when updates are posted to Piazza. You may also use the class web forum to ask general questions of interest to the class as a whole, e.g., administrative issues or project clarification questions. Please do not post any information that would violate the University's Academic Integrity Policy.

Grading

You are responsible for all material discussed in class and posted on the class web page, including announcements, deadlines, policies, etc. Your final course grade will be determined according to the following percentages:

Projects/homework	50%
Readings	9%
Midterm	20%
Final	20%
Meet your professor	1%

Any request for reconsideration of any grading on coursework **must** be submitted within **one week** of when it is returned. Exam regrading requests must be made in writing. Any coursework submitted for reconsideration may be regraded in its entirety, which could result in a lower score if warranted.

Final course grades will be curved as necessary, based on each student's total numeric score for all coursework at the end of the semester. **Important:** Completing the programming assignments is an essential part of the course. Therefore, **we may fail any student who does not make a good-faith attempt on all course projects**, regardless of the student's performance or scores on the other coursework.

Programming Projects

Projects must be submitted electronically following the instructions given in class. Projects **may not** be submitted by any other means (e.g., please do not email your projects to us). It is **your responsibility** to test your program and verify that it works properly before submitting. All projects are due at 11:59pm on the day indicated on the project assignment, according to the submission server's internal clock.

Projects may be submitted up to 24 hours late for a 10% penalty. For example, a project that would earn 90 points for an on-time submission will earn 81 (which is 90 times 0.90) if submitted late. Note that your project score as it appears on the project submission server will not include any late penalties. Any penalties will be incorporated into the final project grade on the grade server.

Exam Scheduling

The class includes a midterm and a final exam. **Tentative** dates for the exams will be posted on the class web site. The exact dates will be confirmed later. We will let you know the exact dates well in advance.

Academic Resources

The Academic Resource Center offers a range of services for students.

Accommodations for Students with Disabilities

Tufts University values the diversity of our students, staff, and faculty, recognizing the important contribution each student makes to our unique community. Tufts is committed to providing equal access and support to all qualified students through the provision of reasonable accommodations so that each student may fully participate in the Tufts experience. If you have a disability that requires reasonable accommodations, please contact the Student Accessibility Services office to make an appointment with an SAS representative to determine appropriate accommodations. Please be aware that accommodations cannot be enacted retroactively, making timeliness a critical aspect for their provision.

Please also contact the instructor to discuss any necessary accommodations.

Excused Absences

You are expected to attend class regularly, complete course assignments on time, and take exams at the scheduled times. If you are unable to fulfill these requirements due to absence for a good reason, the instructor will excuse the absence and provide accommodation. Events that justify an excused absence include:

- Religious observances
- Mandatory military obligation
- Illness of the student or illness of an immediate family member
- Participation in university activities at the request of university authorities
- Compelling circumstances beyond the student's control (e.g., death in the family, required court appearance, etc.)

It is **your responsibility** to inform the instructor **in advance** of intended religious observances. Notice must be provided **immediately** upon an exam date being announced or confirmed for an absence to be excused.

The policies for excused absences **do not** apply to project assignments. Projects will be assigned with sufficient time to allow students to carry out the work even with other responsibilities. In cases of extremely serious documented illness of lengthy duration or other protracted, severe emergency situations, the instructor may consider extensions depending on the specific cirucmstances.

Absences stemming from job interviews, traffic or transportation problems, personal travel, and similar will not be excused.

Academic Integrity

The university's Academic Integrity Policy will be strictly enforced.

Unless otherwise specified, programming projects are to be written **individually**. Therefore, cooperation or use of unauthorized materials on projects is a violation of the Academic Integrity Policy. **Project solutions may not be posted online.** Any evidence of this, or of unacceptable use of computer accounts, use of unauthorized materials or cooperation on exams or quizzes, or other possible academic integrity violations will be reported.

For learning the course concepts, students are welcome to study together or to receive help from anyone else. You may discuss with others the project requirements, the features of the programming languages used, what was discussed in class and in the class web forum, and general syntax errors. Examples of questions that would be allowed are "Does a Java class definition end in a semicolon?" or "What does a 'class not found' error indicate?", because they convey no information about the contents of a project.

When it comes to actually writing a project assignment, other than help from the instructional staff a project must solely and entirely be your own work. Working with another student or individual, or using anyone else's work in any way except as noted in this paragraph, is a violation of the Academic Integrity Policy. You may not discuss design of any part of a project with anyone except the instructor or teaching assistants. Examples of questions you may not ask others might be "How did you implement this part of the project?" or "Please look at my code and help me find my stupid syntax error!". You may not use any disallowed source of information in creating either their project design or code. When writing projects you are free to use ideas or short fragments of code from published textbooks or publicly available information, but the specific source must be cited in a comment in the relevant section of the program.

Violations of the Code of Academic Integrity may include, but are not limited to:

1. Failing to do all or any of the work on a project by yourself, other than assistance from the instructional staff.
2. Using any ideas or any part of another person's project, or copying any other individual's work in any way.
3. Giving any parts or ideas from your project, including test data, to another student.
4. Allowing any other students access to your program on any computer system.
5. Transferring any part of a project to or from another student or individual by any means, electronic or otherwise.

If you have any question about a particular situation or source then consult with the instructor in advance. Should you have difficulty with a programming assignment you should **see the instructional staff in office hours**, and not solicit help from anyone else in violation of these rules.

Right to Change Information

Although every effort has been made to be complete and accurate, unforeseen circumstances arising during the semester could require the adjustment of any material given here. Consequently, given due notice to students, the instructor reserves the right to change any information on this syllabus or in other course materials.

Resources

- Java Basics
  - Java Development Kit (JDK) Download
  - JDK 11 Documentation
  - JDK 11 API
- Java Textbooks (read via Safari Books Online)
  - Head First Java
  - The Java Programming Language
  - Effective Java
  - Java Cookbook
- Related Papers
  - Garlan and Shaw, An Introduction to Software Architecture
  - Klees et al., Evaluating Fuzz Testing
- Other
  - Explaning Code using ASCII Art
  - Why Programs Fail (Slides)

COMP 260
Advanced Algorithms
SPRING 2025

Instructor: Lenore Cowen
Joyce Cummings Center; 627-5134; cowen AT cs.tufts.edu ;
Office Hours: Typically 2-3pm on Monday and Wednesday
Lectures: Mondays/Wednesdays 3pm-4:15pm in JCC 302

See the private page here for the code to add yourself to Gradescope to submit your homework assignments.

Description: If you loved your algorithms class and can't wait for more, this is the class for you. In this pleasant and fun class, we will look at some more modern algorithms, some beautiful algorithms gems, and some areas of current research in algorithms. Topics will include using randomness in the design and analysis of algorithms, approximation algorithms, and online algorithms.

The website for this class is at http://www.cs.tufts.edu/comp/260

Prerequisites: Comp 160 or permission of the instructor.

There is no text for this course.

The course was last taught in 2023.

Here is a Template for scribe notes with instructions for first-time Latex users We can also supply this as an overleaf project-- just ask.

You can find the scribe notes for the class here: http://www.cs.tufts.edu/comp/260/private/

- Week 0/1: January 15 and 22 Review of P and NP, Bipartite perfect matching and Stable Matching
- Week 2: January 27 and 29 Approximation algorithms for Metric TSP , HW 1 due
- Week 3: Feb 3 and Feb 5 Knapsack I and II
- Week 4: Feb 10 and 12 Ford-Fulkerson Algorithm for Max Flow/Min Cut; HW1 due
- Week 5: Feb 19 and 20 Intro to Randomaized Algorithms, Min Cut, MaxCut
- Week 6: Feb 24 and 26 k-Center. Begin scheduling
- Week 7: March 3 and March 5 Set Cover, HW2 due
- Week 8: March 10 and March 12:
- No classes: March 17 and March 19: Spring break
- Week 9: March 31 and April 2: Intro to Online Algorithms
- Week 10: April 7 and April 9: Finish Online lecture and k-Center
- Week 11: April 14 and April 16: Max Cut Revisited and Approx 3 Coloring

- Week 12: April 21 and April 23: Patriot's Day and TBA
- Week 13: April 28 Last Class

Homework Assignments: The code to add yourself to Gradecope for this class to submit electronically is handed out in class or available on the private portion of this website.

HW1 (due Thurs, Jan 30 at 10pm) and here is the Latex source

HW2 (due Thurs, March 6 at 10pm) and here is the Latex source where you will also need This figure to compile the latex source.

The Bleeding Edge:

In this section we link to current research papers that are related to the topics of the lectures of this course.

- Lecture 1: More on stable marriage. See recent work on Generalized Median Stable Matchings of Christine Cheng's on Her webpage

- Lecture 3: While there has been no improvement on Christofides algorithm for metric TSP in terms of approximation factor, there has been more recent approximation algorithms for metric TSP s-t path (i.e. you don't close the cycle) and some very recent work on nearly achieving the same approximation guarantee of Christofides algorithm with a better running time:

- Lecture 4: A PTAS for the Multiple Knapsack problem (a generalization of Knapsak) was found by Chekuri and Khanna in 2006: C. Chekuri, S. Khanna, "A polynomial time approximation scheme for the multiple knapsack problem", SIAM Journal on Computing 35(3), 713-728.

-

COMP170 - Spring 2025

ow you will find our **proposed** schedule for the semester. Topics and assignments are subject to change, and may dated as the semester progresses.

eek 1: 1/12/2025

Wednesday	**Recitation 0:** handout | solution
Thursday	**Lecture:** Introduction Reading: Gödel's Proof, Chapters 1-4 **HW0 Out:** pdf

eek 2: 1/19/2025

Monday	No School
Tuesday	**Lecture:** Paradoxes Reading: Gödel's Proof, Chapter 5
Wednesday	Fake Monday
Thursday	**Lecture:** Gödel Reading: Gödel's Proof, Chapter 7

eek 3: 1/26/2025

Tuesday	**Lecture:** Turing Reading: The Annotated Turing, Chapter 5 **HW0 Due** by the start of class **HW1 Out:** pdf | tex
Wednesday	**Recitation 1:** handout | solution
Thursday	**Lecture:** The Universal Turing Machine Reading: The Annotated Turing, Chapter 9 **QUIZ 1**

eek 4: 2/2/2025

Tuesday	**Lecture:** The Un-Halting Problem Reading: The Annotated Turing, Chapter 9 **HW1 Due** by the start of class **HW2 Out:** pdf | tex
Wednesday	**Recitation 2:** handout | solution

Thursday	**Lecture:** Deciders Reading: The Annotated Turing, Chapter 10

eek 5: 2/9/2025

Tuesday	**Lecture:** The Halting Problem Reading: Sipser, Chapter 4 **HW2 Due** by the start of class **HW3 Out:** pdf | tex
Wednesday	**Recitation 3:** handout | solution
Thursday	**Lecture:** Rice's Theorem Reading: Sipser, Chapter 5 **QUIZ 2**

eek 6: 2/16/2025

Monday	No School
Tuesday	**Lecture:** Mapping Reductions Reading: Sipser, Chapter 5.3 **HW3 Due** by the start of class **HW4 Out:** pdf | tex
Wednesday	**Recitation 4:** handout | solution
Thursday	Fake Monday

eek 7: 2/23/2025

Tuesday	**Lecture:** Finite State Automata Reading: Sipser, Chapter 1.1 **HW4 Due** by the start of class **HW5 Out:** pdf | tex
Wednesday	**Recitation 5:** handout | solution
Thursday	**Lecture:** The Pumping Lemma Reading: Sipser, Chapter 1.4 **QUIZ 3**

eek 8: 3/2/2025

Tuesday	**Lecture:** Nondeterministic Finite State Automata Reading: Sipser, Chapter 1.2 **HW5 Due** by the start of class **HW6 Out**
Wednesday	**Recitation 6:** handout | solution

Thursday	**Lecture:** Context Free Grammars Reading: Sipser, Chapter 2

eek 9: 3/9/2025

Tuesday	**Lecture:** CFGs Continued **HW6 Due** by the start of class **HW7 Out**
Wednesday	**Recitation 7:** handout | solution
Thursday	**Lecture:** The Onion **QUIZ 4**

eek 10: 3/16/2025

Monday	No School
Tuesday	No School
Wednesday	No School
Thursday	No School
Friday	No School

eek 11: 3/23/2025

Tuesday	**Lecture:** P vs. NP Reading: Sipser, Chapter 7.1-7.3 **HW7 Due** by the start of class **HW8 Out**
Wednesday	**Recitation 8:** handout | solution
Thursday	**Lecture:** The Cook Levin Theorem Reading: Sipser, Chapter 7.4

eek 12: 3/30/2025

Tuesday	**Lecture:** Graphs Reading: Sipser, Chapter 7.5 **HW8 Due** by the start of class **HW9 Out**
Wednesday	**Recitation 9:** handout | solution
Thursday	**Lecture:** Closure **QUIZ 5**

eek 13: 4/6/2025

Tuesday	**Lecture:** Subset Sum **HW9 Due** by the start of class **HW10 Out**
Wednesday	**Recitation 10:** handout | solution
Thursday	**Lecture:** TBD

eek 14: 4/13/2025

Tuesday	**Lecture:** P-Space Reading: Sipser, Chapter 8.1 **Extra Credit Homework Out** **HW10 Due** by the start of class
Wednesday	
Thursday	**Lecture:** Shor's Algorithm Reading: Algorithms (Dasgupta, Papadimitriou, Vazirani), Chapter 10 **QUIZ 6**

eek 15: 4/20/2025

Monday	No School
Tuesday	**Lecture:** TBD **Extra Credit Homework Due**
Wednesday	**Recitation 11:** handout | solution
Thursday	**Lecture:** Wrap Up

eek 16: 4/27/2025

Tuesday	
Wednesday	
Thursday	

eek 17: 5/4/2025

Monday	Final Exam: 3:30-5:30pm
Tuesday	
Wednesday	
Thursday	

```
Field1,Field2,Field5_text,Field6_text,Field7,Field8_text
ES 2-01,"MWF 10:30-11:45
Anderson Hall 112",Introduction to Computing in
Engineering,E. Danahy,E. Danahy,
CS 4-01,,Teaching Computer Science,M. Monroe,M. Monroe,
CS 5-01,"T 4:30-6:30p
Eaton Hall 201",Preparing for Career Success,,K. Donoghue,
CS 10-01,"MW 1:30-2:45
Room To Be Announced",Computer Science for All,,E. Cornwall,
CS 11-M1,"T 5:30p-7:00p
Online (synchronous)",Introduction to Computer
Science,,P. Biswas,CS 11-M1
CS 11-01,"TR 10:30-11:45
Room To Be Announced",Introduction to Computer
Science,,E. Cornwall,CS 11-01
CS 11-02,"TR 12:00-1:15
Room To Be Announced",Introduction to Computer
Science,,E. Cornwall,CS 11-02
CS 14-01,,Emerging Scholars in Computer
Science,R. Townsend,R. Townsend,
CS 15-M1,"R 5:30p-7:00p
Online (synchronous)",Data Structures,,C. Magnano,CS 15-M1
CS 15-01,"MW 10:30-11:45
Room To Be Announced",Data Structures,,M. Kazerounian,CS 15-01
CS 15-02,"MW 3:00-4:15
Room To Be Announced",Data Structures,,M. Kazerounian,CS 15-02
CS 20-01,"TR 12:00-1:15
Room To Be Announced",Web Programming,,L. DiOrio,CS 20-01
CS 21-01,"TR 4:30-5:45p
Room To Be Announced",Concurrent Programming,M. Sheldon,M. Sheldon,
CS 23-01,"T 6:00p-9:00p
Room To Be Announced",Game Design,,J. Wiser,CS 23-01
CS 27-01,"TR 10:30-11:45
Room To Be Announced",How Systems Fail,,M. Kazerounian,
CS 28-01,"TR 1:30-2:45
Room To Be Announced",Cyber Security and Cyber
Warfare,M. Chow,"M. Chow,  J. Taliaferro",
CS 30-M1,"W 6:00p-7:30p
Online (synchronous)",Programming for Data Science,A. Couch,A. Couch,
CS 40-01,"TR 1:30-2:45
Room To Be Announced",Machine Structure & Assembly-Language
Programming,N. Mendelsohn,"N. Mendelsohn,  M. Sheldon",CS 40-01
CS 61-M1,"T 7:00p-8:30p
Online (synchronous)",Discrete Mathematics,,M. Jahn,CS 61-M1
CS 61-01,"MWF 9:30-10:20
Room To Be Announced",Discrete Mathematics,,M. Allen,CS 61-01
CS 61-02,"MWF 10:30-11:20
Room To Be Announced",Discrete Mathematics,,M. Allen,CS 61-02
CS 93-01,,Directed Study,,Members of the Department,
CS 98-01,"F 12:00-2:45
```

```
Room To Be Announced",Senior Capstone Project II,,D. Lillethun,
CS 99-01,,,M. Chow,M. Chow,
CS 105-01,"MW 3:00-4:15
Room To Be Announced",Programming
Languages,R. Townsend,R. Townsend,CS 105-01
CS 111-M1,"T 7:00p-8:30p
Online (synchronous)",Operating Systems,,P. Biswas,
CS 111-01,"TR 4:30-5:45p
Joyce Cummings Center 170",Operating Systems,,D. Lillethun,
CS 114-01,"TR 12:00-1:15
Room To Be Announced",Network Security,D. Votipka,D. Votipka,
CS 115-M1,"T 7:00p-8:30p
Online (synchronous)",Database Systems,,C. Doucette,CS 115-M1
CS 115-01,"M 7:00p-8:30p
Room To Be Announced",Database Systems,,C. Doucette,CS 115-01
CS 116-M1,"W 5:30p-7:00p
Online (synchronous)",Introduction to Security,M. Chow,M. Chow,CS 116-
M1
CS 116-01,"TR 4:30-5:45p
Room To Be Announced",Introduction to
Security,M. Chow,M. Chow,CS 116-01
CS 119-M1,"M 5:30p-7:00p
Online (synchronous)",Big Data,,J. Singh,CS 119-M1
CS 119-01,"MW 10:30-11:45
Room To Be Announced",Big Data,,J. Singh,CS 119-01
CS 120-M1,"T 5:30p-7:00p
Online (synchronous)",Web Programming and Engineering,,L. DiOrio,
CS 121-M1,"W 7:30p-9:00p
Online (synchronous)",Software Engineering,,Staff,CS 121-M1
CS 122-01,"MW 1:30-2:45
Room To Be Announced",Parallel Computing,,J. Grodstein,
CS 125-01,"MW 1:30-2:45
Joyce Cummings Center 265",Numerical Analysis,,S. Lee,
CS 131-M1,"M 7:30p-9:00p
Online (synchronous)",Artificial Intelligence,,F. Santini,CS 131-M1
CS 131-01,"MW 6:00p-7:15p
Room To Be Announced",Artificial Intelligence,,F. Santini,CS 131-01
CS 132-01,"MW 4:30-5:45p
Room To Be Announced",Computer Vision,,R. Shilkrot,
CS 135-M1,"F 5:30p-7:00p
Online (synchronous)",Introduction to Machine
Learning,,C. Magnano,CS 135-M1
CS 135-01,"TR 12:00-1:15
Room To Be Announced",Introduction to Machine
Learning,,M. Hughes,CS 135-01
CS 138-M1,"R 7:00p-8:30p
Online (synchronous)",Reinforcement Learning,,Y. Shukla,
CS 138-01,"MW 3:00-4:15
Room To Be Announced",Reinforcement Learning,J. Sinapov,J. Sinapov,
CS 142-01,"MW 4:30-5:45p
```

```
Room To Be Announced",Network Science,L. Cowen,L. Cowen,
CS 144-01,"T 1:30-4:00
Room To Be Announced",Iterative Methods in Machine
Learning,U. Khan,U. Khan,
CS 150-01,"TR 9:00-10:15
Room To Be Announced",HCI for Disability,E. Short,E. Short,
CS 150-02,"MW 10:30-11:45
Room To Be Announced",,R. Townsend,R. Townsend,
CS 150-04,"MW 10:30-11:45
Room To Be Announced",,,F. Dogar,
CS 150-05,"TF 12:00-1:15
Room To Be Announced",,R. Mirsky,R. Mirsky,
CS 150-06,,MITRE eCTF,M. Chow,"S. Bell,  M. Chow",
CS 150-07,"T 5:15p-8:15p
Room To Be Announced",Entrepreneurship for Computer Scientists,,,
CS 150-08,"TR 12:00-1:15
Room To Be Announced",Topics in Computational
Complexity,,V. Podolskii,
CS 151-01,"TR 10:30-11:45
Room To Be Announced","Addressing Cyber Threats,
Vulnerabilities",,L. Weissinger,
CS 151-02,"M 5:30p-7:00p
Joyce Cummings Center 402",Cybersecurity Clinic,M. Chow,M. Chow,
CS 151-03,"TR 4:30-5:45p
Room To Be Announced","Privacy, Security, and Data",,J. Bater,
CS 151-04,"TR 1:30-2:45
Room To Be Announced",Anonymous Communications Theory,,M. Ando,
CS 151-05,"MW 1:30-2:45
Room To Be Announced",Quantum Computer Science,,S. Mehraban,
CS 151-06,"MW 4:30-5:45p
Room To Be Announced",Debugging Cloud Computing,,R. Sambasivan,
CS 151-07,"TR 9:00-10:15
Room To Be Announced",Sustainable Computing,,H. Sundar,
CS 160-M1,"W 7:00p-9:00p
Online (synchronous)",Algorithms,A. Winslow,A. Winslow,CS 160-M1
CS 160-01,"MW 9:00-10:15
Room To Be Announced",Algorithms,,K. Edwards,CS 160-01
CS 167-01,"TR 10:30-11:45
Room To Be Announced",Computational
Biology,D. Slonim,D. Slonim,CS 167-01
CS 170-M1,"R 5:30p-7:00p
Online (synchronous)",Computation Theory,A. Winslow,A. Winslow,CS 170-
M1
CS 170-01,"TR 1:30-2:45
Room To Be Announced",Computation Theory,M. Monroe,M. Monroe,CS 170-01
CS 171-01,"MW 1:30-2:45
Room To Be Announced",Human-Computer
Interaction,R. Jacob,R. Jacob,CS 171-01
CS 178-01,"TR 1:30-2:45
Room To Be Announced",Visual Analytics,R. Chang,R. Chang,CS 178-01
```

CS 183-01,"MTWRF 12:00a-12:00a
Room To Be Announced",Privacy in the Digital Age,,,
CS 185-01,"R 6:30p-9:00p
Room To Be Announced",Computing for Developing Regions,,F. Dogar,
CS 191-01,,Research,,Members of the Department,
CS 193-01,,Directed Study,,Members of the Department,
CS 193-02,,Geometric Folding Algorithms,D. Souvaine,D. Souvaine,
CS 193-03,,MS CoreComp,,K. Edwards,
CS 197-01,,Honors Thesis,,Members of the Department,
CS 203-01,"TR 10:30-11:45
Room To Be Announced",How Systems Fail,,M. Kazerounian,
CS 239-01,"MW 9:00-10:15
Room To Be Announced","Ethics for AI, Robotics, and Human Robot
Interaction",V. Sarathy,V. Sarathy,
CS 260-1,"MW 3:00-4:15
Room To Be Announced",Advanced Algorithms,L. Cowen,L. Cowen,CS 260-1
CS 288-M1,"R 6:00p-7:00p
Online (synchronous)",Master of Science Capstone Project I,,M. Allen,
CS 289-M1,"R 7:00p-8:00p
Online (synchronous)",Master of Science Capstone Project II,,M. Allen,
CS 291-01,"seminar: R 3:00-4:15
Joyce Cummings Center 270",Seminar in Computer
Science,D. Souvaine,"J. Redmond, D. Souvaine",
CS 291-02,"seminar: F 1:30-2:30
Robinson Hall 253",Seminar in Computer Science,J. Sinapov,J. Sinapov,
CS 293-01,,Graduate Special Topics / Master's
Project,,Members of the Department,
CS 296-01,,Master's Thesis,,Members of the Department,
CS 297-01,,Graduate Research,,Members of the Department,
CS 299-01,,Internship in Computer Science,M. Chow,M. Chow,
CS 401-PT,,,,Members of the Department,
CS 402-FT,,,,Members of the Department,
CS 404-01,,,,A. Abdulrazzaq,
CS 405-TA,,,,Members of the Department,
CS 406-RA,,,,Members of the Department,
CS 501-PT,,,,Members of the Department,
CS 502-FT,,,,Members of the Department,
DS 93-01,,Directed Study,,Members of the Department,
DS 98-01,"TR 1:30-2:45
Online (synchronous)",Senior Capstone Project in Data Science
II,A. Couch,A. Couch,
DS 143-01,"T 1:20-4:20
Bromfield-Pearson 005",Data Science for Sustainability,,D. Sunter,
DS 143-02,"M 6:00p-9:00p
Anderson Hall 108",Data Science for Sustainability,,D. Sunter,
DS 153-02,"MW 6:00p-7:15p
Eaton Hall 201",Computer Vision,,R. Shilkrot,DS 153-02
DS 153-03,"MW 10:30-11:45
Collaborative Learning and Innovation Complex 401",,,F. Dogar,
DS 153-04,"TF 12:00-1:15

Tisch Library 316",,R. Mirsky,R. Mirsky,
DS 193-01,,Directed Study,,Members of the Department,
DS 288-M1,"T 7:00p-9:00p
Online (synchronous)",Master of Science Capstone Project I,,M. Allen,
DS 289-M1,"T 8:00p-9:00p
Online (synchronous)",Master of Science Capstone Project II,,M. Allen,
DS 293-01,,Masters Project in Data Science,,Members of the Department,
DS 299-01,,Internship In Data Science,A. Couch,A. Couch,
DS 404-01,,,,A. Abdulrazzaq,

CS 15 - Spring 2025

Welcome to CS 15!

CS 15 is all about building your programming toolkit. The tools are data structures, the means by which we store, organize, and access data. We will implement and utilize the key data structures and algorithms that every programmer must know, learn about the useful abstractions that they provide, and study the costs in time and space associated with them.

The structure and assignments of this course are based largely on prior versions of the course taught by Marty Allen, Mark Sheldon, Mati Korman, Chris Gregg, Ben Hescott, and Bruce Molay. Thank you!

Course Information

Course Policies: All students are expected to know and adhere to our course policies. Please take the time to read our admin page in full.

Prerequisites: CS 11 or consent of instructor.

Textbook: No purchase required. We will regularly post online readings from *Data Structures and Algorithm Analysis* by Clifford A. Shaffer.

Instructor: Milod Kazerounian
Email: milod.kazerounian@tufts.edu
Office: JCC 469
Office Hours: Tuesdays from 1-2pm in JCC 469, Thursdays from 2-3pm on zoom

Our graduate TA is Max Liu. Our teaching fellows are Arya Prasad, Hameedah Lawal, Kathy Quintanilla, and Liam Drew. We also have a horde of teaching assistants who will hold office hours in the third floor common space in the Joyce Cummings Center. Schedules for office hours will be regularly updated on our course forum, Piazza.

Lecture Section 1: (Mon/Wed) 10:30 – 11:45am
Location: Barnum Hall LL08
Final exam: May 2nd from 12-2pm

Lecture Section 2: (Mon/Wed) 3:00 – 4:15pm
Location: Robinson 253
Final exam: May 2nd from 3:30-5:30pm

For Spring 2025, both lectures will be automatically recorded by Tufts IT and posted on the CS 15 Canvas site under Echo 360. If you miss a lecture, you can find it there.

We do not use Canvas for anything else in CS 15!

Because all lectures are recorded, we are all on video with audio. The videos are for use by this class only and must not be downloaded or distributed.

CS 40 — Spring 2025

- Home
- Calendar
- Assignments
- Admin
- Resources

Announcements

This calendar represents the current plan, but it is subject to change. Please refresh frequently and look for announcements on Piazza.

All submissions are due by **11:59 pm (23:59)** at the end of the specified due date. Design documents should be submitted before **6 pm,** because submissions as of that time will be graded and have their feedback returned first. This means you can apply that feedback to the final submission! Design submissions made after 6 pm and before 11:59 pm will not be charged an extension token, but the feedback may not arrive in time to be incorporated into your final submission.

Read the specs carefully to understand the submission expectations for each assignment.

Mon	Tue	Wed	Thu	Fri
			16Jan Intro to CS 40 and Transitioning to C Read: C++ to C slides Self-study module: the transition from C++ to C (due Friday) CS 40 Admin & Policies Hanson: Ch. 1 & 2; Intro and First Section (titled "Interface") of Ch. 3-4, 7-9, 11; Section 8.2 Piazza HW 1: filesofpix	

20Jan
Martin Luther King Day
	21Jan
	Abstraction, Modularity, Interfaces, and Pointers
	
	Read:
	Introduction to Exceptions
	C Idoms for CS 40
		22Jan *(Monday schedule)*
		
		Add deadline
			23Jan
			More on Abstract Data Types
			
			Read:
			UArray
			Void and Function Pointers
			Hanson's table interface, analyzed
			`cbr_example.c`
			`list_map_slide_2024.c`
			`list_map_slide_accum_2024.c`
				24Jan
				Lab 2: Unboxed arrays and 2D Arrays (HW2 Partner)

		15Jan Classes begin		17Jan Lab 1: Rawness (HW1 Partner)

	Invariants		`list_map_stats.c`	
			HW 1 design due (6 pm)	
27Jan	28Jan What is a Bit?	29Jan HW 1 due HW 2: iii	30Jan Building C Programs Read: B&O: Ch. 1 through 1.3; Sec. 1.7.4, 3.2.2 (use of objdump); Ch. 7 through 7.3 What happens when you build a program with `gcc`?	31Jan Lab 3: Problem-solving and black (HW2 Partner) Fall incompletes deadline
03Feb HW 2 design due (6 pm)	04Feb Process Memory Read: B&O: 1.7.1, 1.7.3. 9.1, 9.2 `memory.c`	05Feb	06Feb The Memory Hierarchy and Intro to Caching Read: Powerpoint with animations for slides B&O: Ch. 1.5, 1.6, Introduction to Ch. 6, 6.1.1 and 6.1.2 (just the first paragraph of each), 6.2-6.3	07Feb Lab 4: Work on HW2 (HW2 Partner)
10Feb HW 2 due HW 3: locality	11Feb More on Caching Read: Slides Powerpoint with animations B&O: The remainder of Ch. 6 Optional: A true story about a cache flex	12Feb	13Feb Object Oriented Programming	14Feb Lab 5: Striding through memory (HW3 Partner)
17Feb Presidents' Day HW 3 design due (6 pm)	18Feb Numbers and Machine Arithmetic Read: Integer slides Powerpoint with animations for Integer slides Floating point slides Floating point powerpoint with animations	19Feb Drop deadline	20Feb *(Monday schedule)* HW 3 due HW 4: arith Alan Turing proposes Turing test 20 Feb 1947	21Feb Lab 6: Diff (HW4 Partner)

	B&O: 2.2 – 2.4, Useful reference on numeric data (from Liam) fact.c fact.erl			
24Feb	25Feb Shifting, Masking and Endianness Read: Shifting and Masking slides Powerpoint with animations for Shifting and Masking slides Bit and Little Endianness slides Big and Little Endianness powerpoint with animations B&O: Start of chapter 2 – 2.1 endian.c	26Feb HW 4 design due (6 pm)	27Feb Introduction to Assembly Language Read: `hello_world_x86.s` B&O: 3.1-3.7 (warning: examples are not AMD64), 3.8-3.11 (skim), 3.13 This AMD64 Cheat Sheet from Brown is by far my favorite! For fun: IBM System 370 Reference Card	28Feb Lab 7: Bits and Shifts (No Partner)
03Mar	04Mar Assembly Continued Read: Assembler demos you can try Optional: Real programmers use machine code First Cray-1 shipped to LANL (1977) 	05Mar	06Mar Assembly Finale Read: Factorial Assembly Code HW 4 due HW 5: bomb	07Mar Lab 8: The GNU Debugger (HW5 Partner) Following are due on or happen on Saturday 08Mar: HW 4arith challenge Following are due on or happen on Sunday 09Mar: HW 4arith challenge due

	11Mar Midterm Review Read: Midterm Prep Refactoring and practice		13Mar Midterm Read: Midterm solutions	14Mar Lab 9: Closing the loop on reverse engineering (HW5 Partner)
10Mar		12Mar		
17Mar Spring break	18Mar Spring break	19Mar Spring break	20Mar Spring break	21Mar Spring break
24Mar Begin advising period	25Mar Introducing the CS 40 Universal Machine	26Mar	27Mar Implementing the Universal Machine HW 5 due HW 6: um	28Mar Lab 10: Unit testing for the UM (HW6 Partner)
31Mar	01Apr Performance Introduction Read: B&O: 1.9.1, Intro to Ch. 5, 5.7 (many details about how processor architecture affects performance) Be a CA (TA)!	02Apr HW 6 design due (6 pm) Withdraw deadline PF deadline	03Apr Performance Tools Read: B&O: Sec. 5.14 First cell phone call 3 Apr 1973	04Apr Lab 11: Profiling Tools (HW7 Partner)
07Apr IBM announces System 360 computer family 7 Apr 1964	08Apr Performance Tuning Read: B&O: Sec. 5.1, 5.4-5.6 inlining.c specialization.c	09Apr	10Apr The UM Macro Assembler Read: UM Macro Assembler Language Specification UMASM Lecture Notes HW 6 due HW 7: profiling End advising period	11Apr Lab 12: Extending the UM (No Partner) Begin fall registration
14Apr	15Apr Programming with UMASM	16Apr End fall registration First Lisp paper, by John	17Apr UMASM Programming Review and Advanced Examples Read: UMASM examples	18Apr Lab 13: Scripting/Q&A + HW 7 work

		McCarthy 16 Apr 1960		
21Apr Patriot's Day (Observed) HW 7 due HW 8: asmcoding	22Apr No lecture: work on HW	23Apr	24Apr CS 40 Wrap-up Final exam info Read: Final exam prep The UM in Python	25Apr Lab 14: Work on HW8 (HW8 Partner)
28Apr HW 8 due Classes end Grad student withdraw deadline	29Apr *Make up day: meet only if we've missed classes* Reading period	30Apr Reading period Claude Shannon's birthday 30 Apr 1916	01May Reading period	02May Begin final exams
05May Final exam *Tentatively 1530–1730 (i. e., 3:30–5:30pm)* Location: TBA	06May	07May	08May	09May End final exams
12May Degree candidates' grades due 0900 Begin Senior Week	13May	14May	15May	16May Remaining grades due 0900 End Senior Week Following are due on or happen on Sunday 18May: Commencement

CS 40 — Spring 2025

Course Summary

Welcome to CS 40 - Machine Structure and Assembly Language Programming! CS 40 will give you two important experiences:

1. You will turn the corner from introductory programming to programming as a professional.
2. You will study in detail the structure of modern computers, and that in turn will add greatly to your understanding of how computer hardware, operating systems, compilers, language runtimes, and other tools work together to influence the performance of your programs.

These two goals are very closely related: all good computer professionals know how to apply deep knowledge of computing systems to their programming projects and to their research in computer science. Whether you are a newly confirmed CS major, someone who is considering a CS major but is unsure, or a student from another field who is interested in a deep look at some of the most important technologies of our time, CS 40 should give you an intense, realistic and (we hope!) exciting look at how modern computers work.

New CS 40 Students

Many students are eager to get a head start on the CS 40 material so that they can arrive as prepared as possible to the first day of class. Here is what you can do to prepare:

1. When CS 40 begins, you will be making an immediate transition from C++ to C. Be ready for this. Our resources page provides some great tutorials on this transition.
2. Read our admin page. This page is a comprehensive review of our course policies. You will be expected to adhere to these policies, so a good start is to know what they are. Pay particular attention to our course coding standands and our pair programming conduct guidelines.
3. Visit the course Piazza forum. (Your instructor will add you to the Piazza roster at the start of the term.)
4. Make sure that you have access to our course textbooks, which are listed below. In particular, the Hanson book describes the Swiss army knife of tools that will be provided to you for use throughout the semester.
5. Familiarize yourself with our Halligan Helper system. This is how students can request help from TAs during office hours.

Textbooks

The following two textbooks are required:

Textbook: *C Interfaces and Implementations* by David R. Hanson
ISBN: 9780201498417
Online access: via Tufts Library
There is also a PDF Quick Reference.

Textbook: *Computer Systems: A Programmer's Perspective*, 3rd edition by Bryant and O'Hallaron

ISBN: 9780134092669
Important: Make sure you avoid the "International" or "Global" version of this textbook, which is in paperback or the Amazon ebook. While substantially cheaper, many of the chapter practice problems have been rewritten by the publisher and contain many errors and inconsistencies.

Additional books that you may fine useful are listed on our resources page.

Schedule

Lectures: Tuesdays / Thursdays 1:30 – 2:45 PM EST
Location: SEC Robinson 253

Labs: Friday 10:30 – 11:45 AM, 12:00 – 1:15 PM, and 1:30 – 2:45 PM EST
Location: Joyce Cummings Center 235 and 240

Midterm Exam: See the course calendar
Final Exam: See the course calendar

Staff

Instructor: Mark Sheldon
Email: msheldon@cs.tufts.edu
Office hours: Will be posted at https://www.cs.tufts.edu/~msheldon about the second week of the semester.
Instructor: Noah Mendelsohn
Email: noah@cs.tufts.edu
Office hours: Will be posted at https://www.cs.tufts.edu/~noah/office_hours.php about the second week of the semester.

Our teaching fellows are Dan Bergen, Valerie Zhang, Sarah Svahn, and Julia Zelevinski. We also have an army of undergraduate teaching assistants who will hold office hours. Schedules for the undergraduate assistants will be posted and updated on Piazza.

Please refer to our admin page for guidelines on how to contact our course staff with questions and feedback.

Home

CS 61: Discrete Mathematics - Schedule

Main page • Syllabus

DATE	ACTIVITY	TOPIC	READINGS
WEEK 1			
Wed. Jan. 15	Lecture 1A	Intro to discrete mathematics	Syllabus
Fri. Jan. 17	Lecture 1B	Fundamental reasoning	Scheinerman (SCH), sections 1–4
WEEK 2			
Mon. Jan. 20		**No class (MLK day)**	
Wed. Jan. 22	Lecture 2A	Proofs	SCH, section 5
Fri. Jan. 24	Lecture 2B	Counterexamples; Boolean algebra	SCH, sections 6–7
WEEK 3			
Mon. Jan. 27	**Homework 1**	Submit on Gradescope by 11:59pm	
Mon. Jan. 27	Lecture 3A	More on Boolean algebra	SCH, section 7
Wed. Jan. 29	Lecture 3B	Lists; factorials	SCH, section 8
Fri. Jan. 31	Lecture 3C	Sets	SCH, sections 9–10
WEEK 4			
Mon. Feb. 03	**Homework 2**	Submit on Gradescope by 11:59pm	
Mon. Feb. 03	Lecture 4A	Quantifiers	SCH, section 11
Wed. Feb. 05	Lecture 4B	Set operations	SCH, section 12
Fri. Feb. 07	Lecture 4C	Combinatorial proofs	SCH, section 13
WEEK 5			
Mon. Feb. 10	**Homework 3**	Submit on Gradescope by 11:59pm	
Mon. Feb. 10	Lecture 5A	Sets	SCH, section 12
Wed. Feb. 12	Review day	Review for exam	
Fri. Feb. 14	**Midterm 01**		Exam is held during usual class meeting, 170 JCC
WEEK 6			
Mon. Feb. 17		**No class (Presidents' Day)**	
Wed. Feb. 19	Lecture 6A	Sets and combinatorial proofs	SCH, sections 12–13

Thu. Feb. 20 (**Tufts Monday**)	Lecture 6B	Relations	SCH, section 14
Fri. Feb. 21	Lecture 6C	Equivalence relations	SCH, section 15
WEEK 7			
Mon. Feb. 24	**Homework 4**	Submit on Gradescope by 11:59pm	
Mon. Feb. 24	Lecture 7A	Partitions	SCH, section 16
Wed. Feb. 26	Lecture 7B	Binomial coefficients	SCH, section 17
Fri. Feb. 28	Lecture 7C	Binomial coefficients, cont'd.	
WEEK 8			
Mon. Mar. 03	**Homework 5**	Submit on Gradescope by 11:59pm	
Mon. Mar. 03	Lecture 8A	Proofs by contradiction and contrapositive	SCH, section 20
Wed. Mar. 05	Lecture 8B	Proof by induction	SCH, section 22 (up to, not including, start of Strong Induction, p. 141)
Fri. Mar. 07	Lecture 8C	Strong induction	SCH, section 22 (141–145)
WEEK 9			
Mon. Mar. 10	**Homework 6**	Submit on Gradescope by 11:59pm	
Mon. Mar. 10	Lecture 9A	Functions	SCH, section 24
Wed. Mar. 12	Lecture 9B	Functions, cont'd.; the Pigeonhole Principle	SCH, section 25
Fri. Mar. 14	Lecture 9C	Function composition	SCH, section 26
WEEK 10			
Mon. Mar. 24	Lecture 10A	Big-O notation and functions	SCH, section 29 (up to page 207 only)
Wed. Mar. 26	Review day	Review for exam	
Fri. Mar. 28	**Midterm 02**		Exam is held during usual class meeting, 170 JCC
WEEK 11			
Mon. Mar. 31	**Homework 7**	Submit on Gradescope by 11:59pm	
Mon. Mar. 31	Lecture 11A		
Wed. Apr. 02	Lecture 11B		
Fri. Apr. 04	Lecture 11C		
WEEK 12			

Mon. Apr. 07	**Homework 8**	Submit on Gradescope by 11:59pm;	
Mon. Apr. 07	Lecture 12A		
Wed. Apr. 09	Lecture 12B		
Fri. Apr. 11	Lecture 12C		
WEEK 13			
Mon. Apr. 14	**Homework 9**	Submit on Gradescope by 11:59pm;	
Mon. Apr. 14	Lecture 13A		
Wed. Apr. 16	Lecture 13B		
Fri. Apr. 18	Lecture 13C		
WEEK 14			
Mon. Apr. 21	**Homework 10**		Submit on Gradescope by 11:59pm
Mon. Apr. 21	Lecture 14A		
Wed. Apr. 23	Review Day	Review for Exam	
Fri. Apr. 25	**Midterm 03**		
WEEK 15			
Mon. Apr. 28	Last day	Wrap-up and review	
FINALS WEEK			
Fri. May 02	**Final (section 02)**		Exam is held 12:00–2:00 PM, 170 JCC
Tue. May 06	**Final (section 01)**		Exam is held 7:00–9:00 PM, 170 JCC

CS 61: Discrete Mathematics - Syllabus

Main page • Schedule

Description and Objective: An introduction to foundational mathematical concepts and techniques: Sets, relations and functions, logic and methods of proof, combinatorics, graphs and digraphs.

Objectives for the Course

By the end of the semester, a successful student will be able to do all of the following things:

- Write mathematical proofs at an introductory level
  - Frame your arguments in a clear, logical, and convincing fashion
  - Employ standard methods such as proof by induction and proof by contradiction
- Develop linguistic precision
  - Employing technical concepts in correct mathematical language, with correct syntax
  - Translate back and forth between mathematical/logical language and everyday natural language
- Understand and employ basic mathematical concepts and tools
  - Propositional and quantificational logic
  - Set theory
  - Relations and functions
  - Graph theory

Course Materials

1. **Textbook [SCH]:** *Mathematics: A Discrete Introduction*, 3rd ed., by Edward R. Scheinerman, Brooks/Cole/Cengage (2013).
   ISBN: 9780840049421
2. **Your own notes:** This course does not use slides. Regular note-taking is expected. Lectures will be recorded and posted to Canvas, but we can't promise anything with respect to recording quality on any given day.

Prerequisites and Expected Competencies

CS 11 or Math 32 is useful, but the primary outcome we are looking for is mathematical maturity, so that you are prepared to understand and write proofs, consider new number systems such as modular arithmetic, and be comfortable reading symbols such as set notation. You will need a few algebra skills here and there. This is not a programming course.

Requirements & Grading

Grades will be based on the following:

1. 30% Homework (10 assignments; all scores counted)
2. 40% Midterm examinations (3 exams; approx. 13.3% each)
3. 25% Final examination
4. 5% Participation/attendance during lecture sessions

Letter Grades: We use the following breakdown of letter grades and percentages:

98–100% A+	87–89% B+	77–79% C+	67–69% D+
93–97% A	83–86% B	73–76% C	63–66% D
90–92% A–	80–82% B–	70–72% C–	60–62% D–

Homework

Homework will be assigned regularly in the course. The homework due dates are listed on the schedule page. In general, students will have about one week for an assignment. Homework will be submitted using the Gradescope system, information about which will be provided when relevant.

Homework must be legible. If it is not type-written, then it must be so neatly written as to be equivalent. If illegibility makes grading slow and difficult, we will not grade the submission.

Students with extreme special circumstances must meet with the professor to make other arrangements to the scheduled homework and exams.

Students are welcome and encouraged to work together and discuss homework verbally. Every homework assignment should be written up separately and individually (please review the section on academic misconduct below). **Do not search online for solutions, and do not use AI tools to generate your work for you**.

Regrade requests for all homework assignments must be submitted within a week of the grades being released.

Exams

The exams will be in written format. Unless otherwise announced, exams will take place in the same room as the class itself. Example exams, to show the format and type of question, will be distributed before each exam occurs.

Students with extreme special circumstances must meet with the professor to make other arrangements to the scheduled homework and exams.

Students must submit an attempt at all exams in order to pass the course.

Regrade requests for all exams must be submitted within a week of the grades being released.

Policy on Late and Missing Work

For late homework assignments, handed in **within 24 hours after** the time at which it was due, a reduction of 10% will occur; if handed in **within 48 hours** of the expected time, a reduction of 20% will occur; **within 72 hours** the reduction will be 40%. No credit is given for assignments submitted after that point, without a documented reason. There is **no late deadline for exams**: they must be completed during the assigned exam period given.

Class Participation

Occasionally, we will distribute and collect lecture response forms during class. These will help us keep track of where everyone is with respect to the material, and will count for the participation portion of the grade. Up to 2 can be missed without penalty, but missing more than that will result in a reduction to the participation grade.

Policy on Collaboration

I encourage you to work together on the material. This is a great way to learn, and to share ideas. However, in order to actually learn something, it is important that you complete the real work on your own, unless specifically directed otherwise. It is perfectly fine for you to discuss the general approach to a problem with one another, work out how to understand a proof or concept, and to help one another with things like setting up and using LaTeX (if you decide to use it). However, it is **not okay** to copy proofs and other materials from anyone inside or outside of the class. While you can of course use online references to explain key concepts, and to learn what you need to succeed, don't simply copy answers you find online, and you should cite any such materials you consulted. Citation should extend to any AI tools that you endeavor to use to assist with homework, along with a clear explanation of how you used the tool, and how much of the work submitted should be credited to your own work, versus the work of the tool.

TL;DR: Do your own work. This is the only way to actually learn the material.

Piazza & Collaboration

When using the Piazza forum, the same sorts of considerations about collaboration are in play when posting questions and providing answers.

Questions may be posted as either **private** (viewable only by yourself and course staff) or **public** (additionally viewable by all students for the course registered on Piazza). Some issues warrant public questions and responses, such as: misconceptions or clarifications about the instructions, conceptual questions, errors in documentation, etc. Some issues are better with private posts, including: debugging questions that include extensive amounts of code, questions that reveal a portion of your solution, etc.

In addition, if you wish, remember that you can make your public posts **anonymous**, meaning that the instructors will know who is posting, but other members of the class will not. If that is something you want to do, for whatever reason, that is fine.

Please use your best judgment when selecting private vs. public. If in doubt, make it private.

Academic Misconduct

Students should read and know the Tufts policies on academic integrity. If a student does not understand these terms or any of the material listed on this page, it is his/her responsibility to talk to the professor. To be brief: do your own work, cite any sources from which you take ideas, and give credit where credit is due.

Inclusivity

In the classroom, we do not only expect participation, but also that everyone be treated with dignity and respect. Our perspective will be that everyone comes from a different background, each with their own personal history, identity, and background knowledge. Our knowledge will always be used to better everyone in the class. As an instructor, I have my own specific background and perspective on life, along with my own history of mental and physical challenges. I don't presume that my experience is the same as anyone else's, but will always do my best to meet all of my students where they live, to the best that I can. I'll probably make some mistakes, but I will be trying.

Policy on Sharing

This course is designed for everyone to feel comfortable participating in discussion, asking questions, learning, and facilitating the learning of others. In order for that atmosphere to be maintained, the recordings of our conversations will only be shared with the enrolled students in the class (not posted publicly), and it is prohibited

for any of us who have access to the video to share it outside the course. Additionally, some readings are provided on a fair-use basis while taking this course, and are not to be distributed otherwise. This especially includes any posting or sharing of readings, videos, or other recordings on publicly accessible websites or forums. Any such sharing or posting could violate copyright law or law that protects the privacy of student educational records.

Writing Support

The StAAR Center for accessibility and academic resources offers friendly, experienced, non-judgmental writing support to writers at all levels of expertise through any stage in the writing process (for free!), and I highly recommend that you take advantage of this excellent opportunity.

Accessibility

Tufts University values the diversity of our students, staff, and faculty, recognizing the important contribution each student makes to our unique community. We are committed to providing equal access and support to all students through the provision of reasonable accommodations so that each student may access their curricula and achieve their personal and academic potential. If you have a disability that requires reasonable accommodations you can contact the StAAR Center (617-627-4539) to make an appointment to determine appropriate accommodations. Please be aware that accommodations cannot be enacted retroactively, making timeliness a critical aspect for their provision.

Counseling & Wellness

As a student, there may be times when personal stressors or emotional difficulties interfere with your academic performance or well-being. Counseling and Mental Health Services (CMHS) provides confidential consultation, brief counseling, and urgent care at no cost for all Tufts undergraduates as well as for graduate students who have paid the student health fee. To make an appointment, call 617-627-3360. Please visit the CMHS website to learn more about their services and resources.

CS 115 Database Systems, fall 2022

Syllabus v1.2.1

Course Description

Fundamental concepts of database management systems. Topics include: data models (relational, object-oriented, and others); the SQL query language; implementation techniques of database management systems (storage and index structures, concurrency control, recovery, and query processing); management of unstructured and semi-structured data; and scientific data collections.

Learning Outcomes

Upon completion of the course, students will be able to:

1. Explain database design concepts (including ACID, CAP, eventual consistency, and vertical/horizontal scaling) for both relational

2. Write queries that correctly and efficiently retrieve (SELECT), INSERT, UPDATE, and DELETE data in relational (SQL) databases, as well as queries that perform equivalent operations in several types of NoSQL databases.
3. Design a relational (SQL) database schema from a problem description and express the design as an Entity-Relationship (ER) or UML diagram or SQL language schema.
4. Enforce constraints on the data in relational (SQL) databases using schema constraints in the SQL language as well as using triggers/rules.
5. Prove that a relational (SQL) database is or is not in a particular normal form (3NF, BCNF/3.5NF, or 4NF), and transform a database schema into a particular normal form.
6. Choose the appropriate database system to use for a particular application, from relation (SQL) or several types of NoSQL databases.
7. Design databases for each of several types of NoSQL databases.
8. Use software design patterns that facilitate application code interfacing with database systems.

Prerequisites

- **CS 15 (Data Structures)** or *Graduate Student* standing
    - Grad students are expected to have had some sort of Data Structures course (e.g., CS 15, CS 205, or a Data Structures programming course from another college).
- Ability to program in some high-level language (e.g., C, C++, Java, etc.) and a willingness to learn new languages
    - You will receive explicit instruction on the SQL language in this

JavaScript (and possibly other languages), and you will be

expected to get up to speed on your own with what you need to know to follow along with the examples. However, you will not be expected to write complete programs yourself in any of these languages; just a few lines at the most.
- In order to best understand some of the example and database concepts, you should be familiar with basic object-oriented programming concepts (classes, methods, abstraction/interfaces, etc.) as well as common data structures, such as trees

Course Materials

Required Textbook:

***Database Systems: The Complete Book, 2nd Edition*. H. Garcia-Molina, J. Ullman, and J. Widom (c) 2009. Pearson.**

ISBN-13: 9780131873254

(Hardback, paperback, and digital editions are all fine. New, used