Tuesday	**Lecture:** Memory Pt. 3: Putting it all together (pre-recorded lecture videos) partners.cpp students.txt **HW4 Due** by midnight
Wednesday	**Lab 5:** Black Book **HW5 Out:** Phone Tree
Thursday	No Class

eek 8: 3/2/2025

Tuesday	**Lecture:** Monster Mash: More Pointers in Structs **HW5 Due** by midnight
Wednesday	**Lab 6:** Query Quaziness **HW6 Out:** Mutations
Thursday	**Lecture:** Complexity and Randomness Read: pp. 994-997 (i.e., the part of Sec. 18.3 labeled "Running Times and Big-O Notation") Read: pp. 188-189 (i.e., the part of Sec. 4.2 labeled "Random Number Generation")

eek 9: 3/9/2025

Tuesday	**Lecture:** Midterm Review **HW6 Due** by midnight
Wednesday	**Lab 7:** Midterm Review **HW7 Out:** Airtight
Thursday	**Midterm Exam** In Class

eek 10: 3/16/2025

Monday	Spring Break!
Tuesday	Spring Break!
Wednesday	Spring Break!
Thursday	Spring Break!
Friday	Spring Break!

eek 11: 3/23/2025

Tuesday	**Lecture:** Recursion (Part 1) Read: Ch. 14 **HW7 Due** by midnight
Wednesday	**Lab 8:** Recurse, Recurse, Recurse, Recurse... **HW8 Out:** Lineage
Thursday	**Lecture:** Recursion (Part 2)

eek 12: 3/30/2025

Tuesday	**Lecture:** Object-Oriented Programming: Intro Read: Sec. 6.1 (Only the part titled "Introduction to Classes and Objects"), Sec. 10.2 (Up through th part titled "Summary of Some Properties of Classes") **HW8 Due** by midnight
Wednesday	**Lab 9:** Fetch **HW9 Out:** Advising Period
Thursday	**Lecture:** Object-Oriented Programming: Next Level Read: Sec. 10.2 (the rest of it), 11.4 (Stop when you hit "Copy Constructors") What goes in a .h and what goes in a .cpp?

eek 13: 4/6/2025

Tuesday	**Lecture:** OOP Implementation: Vectors Part 1 **HW9 Due** by midnight
Wednesday	**Lab 10:** Aces **HW10 Out:** Rack-O
Thursday	**Lecture:** OOP Implementation: Vectors Part 2

eek 14: 4/13/2025

Tuesday	**Lecture:** Inheritance **HW10 Due** by midnight
Wednesday	**Lab 11:** Quell the Compiler **Final Project Out:** Sushi Go!
Thursday	No Class - Work on Project

eek 15: 4/20/2025

Tuesday	No Class - Work on Project
Wednesday	**Lab 12:** Project Work (Optional)
Thursday	No Class - Work on Project

eek 16: 4/27/2025

Monday	Classes End **Final Project Due** by midnight
Tuesday	Reading Period
Wednesday	Reading Period
Thursday	Reading Period

eek 17: 5/4/2025

Tuesday	**Final Exam**

CS 11 - Spring 2025

---

tructor: Elyse Cornwall
ail: elyse.cornwall@
ice: Joyce Cummings Center, Room 457
ice Hours: Tues 2:30-4pm

Lecture: Joyce Cummings Center, Room 270
Section 1: Tues/Thurs 10:30-11:45am
Section 2: Tues/Thurs 12:00-1:15pm
Lab: Joyce Cummings Center, Rooms 235 and 240 (see

dterm Exam: This course will have an in-person midterm exam in class on Thursday, March 13th.
al Exam: This course will have an in-person final exam on Tuesday, May 6th. In order to take CS11, you must be ab
end this final exam. The Section 1 exam is from 3:30-5:30pm. The Section 2 exam is from 12:00-2:00pm.

ble of Contents

s page contains course information pertaining to:

ourse Overview

s course serves as an introduction to computer science via the programming language C++. You will learn how to
ise precise procedures for solving problems, and how to specify these procedures using the C++ programming
guage. Along the way, you will strengthen your computational thinking skills (helpful for any form of problem solvin
l begin to form a mental model of how a computer operates.

I1 is a fast-paced, challenging course that may require more of your time and effort than what you've experienced
er courses. By staying aware of the course policies, following the advice of the course staff, and taking advantage
provided resources, you will set yourself up for success in this course regardless of your background.

urse Goals

the end of this course, students should be able to:

1. Demonstrate computational problem solving with basic programming constructs.
2. Interpret a sequence of English instructions (an algorithm) as a C++ program and vice versa.
3. Apply computational thinking to solve problems.
4. Assess a C++ program's aesthetic value and functional correctness.

urse Expectations

se expectations are designed to prevent students from missing important information or misunderstanding policie
t are essential for succeeding in the course. If you experience a negative outcome in CS11 (e.g. losing points on an
essment, missing a regrade request window, not begin granted an extension) due to a failure to meet one of these
ectations, that outcome will not be changed.

a CS11 student, you agree to:

- Read this entire syllabus during the first week of class.
- Adhere to all the policies outlined in this syllabus.
- Read all pinned Piazza posts carefully throughout the semester (Piazza is explained below). This is the main mechanism for distributing course information such as changes to policies, deadlines, etc.
- Follow the three CS 11 axioms:

    1. **Start early.** Unlike a set of individual problems, a program has to function as a whole. The more you build, the more carefully you must think about how the various components fit together. If you're in a rush, it will be difficult to build a functional program; give yourself as much time as possible!
    2. **Think before you code.** A program is just a faster and more reliable version of a procedure that can be done with a pencil and paper. If you do not have a clear idea of how you would solve the problem with a pencil and paper, then you are not ready to write code for it.
    3. **Write a little, test a little.** Finding a bug in 5 lines of code is far easier than finding a bug in 500 lines of code. A good program is written in small, testable increments. You should not write the next section of your program unless you have a clear plan for how to test it.

nal Grade Breakdown

ır course grade will be produced as a weighted sum of your scores in four categories:

1. Labs (10%)
2. Homework (60%)
3. Midterm Exam (10%)
4. Final Exam (20%)

ades will be posted on Gradescope for students to view. We do not expect to apply a curve to any portion of this gr

omework Logistics

eneral, homwork assignments will go out every Wednesday and will be due at midnight the following Tuesday (the
be a longer project towards the end of the semester). Each homework assignment will consist of three componen

1. A written component that will be submitted via Gradescope
2. A programming component that will be submitted via our **`submit11`** system
3. A style component that evaluates how well your code conforms to our style guide

nework grades will be posted to Gradescope for students to view.

itten

e written component of your homework must be submitted as a PDF via Gradescope. Failing to submit your written
nponent or submitting a non-PDF document will result in a loss of credit for this component.

ogramming

e programming component of each assignment must be submitted via our **`submit11`** system. First and foremost, t
graming component of your homework **must compile** on our Halligan servers in order to receive any credit. Once
mission has compiled, it is evaluated by comparing its output to the output that our solution code produces.

h the exception of HW0, programming submissions will be graded by an automated system that uses the program
`ff` to compare a submission's output to our ground truth. Students will learn to use **`diff`** during their second lab
sion and are expected to use it for testing before submitting all of their subsequent homework assignments.

the majority of the homework assignments, we will be providing one or more sample input files that students may
testing. While these tests are designed to be generally representative of the tests we will be using for grading, the
intended to be comprehensive. Students are expected to create their own input scenarios and test them using **`di`**

e triplicate of **compile-diff-submit** should become the cornerstone of every student's submission process.

te Homework

e Late Token System

recognize that in some circumstances, it is not possible to meet an assignment deadline. In these circumstances,
utilize the late token system. Each student is automatically issued **five** "late tokens" to be used on homework
ignments; a maximum of two tokens can be used on a single assignment. A late token grants you a 24-hour exten
an assignment, and requires no action on your part; our grading software will automatically check the date and tim
r submission and deduct the appropriate number of tokens. Note the following late token rules:

1. Once you are out of late tokens, late homework will receive no credit.
2. If you don't want to use a token, don't submit anything after the due date. We grade your most recently submitted work!

e total number of tokens used on an assignment is reported along with your homework and lab grades on Gradesc
ortunately, Gradescope does not provide a summary of how many late tokens you've used over the semester; it is
ponsibility to keep track of your late token usage.

e tokens are designed to accommodate short-term setbacks like catching a cold or an ill-timed deadline in anothe
ss. Again, there is no need for any emails or explanations. Just turn in the assignment when you get it done, and th
e token accounting will happen automatically.

an-Approved Extensions

nore extreme extenuating circumstances, the late token system may not be enough. In these cases, students must
actively reach out to their dean to request an extension (note that we rarely grant extensions that are requested o
t the due date). Your dean will then work with the course instructor to make appropriate arrangements.

grade Requests

are always welcome and encouraged to ask for an explanation of the grade you received; simply post on Piazza
vately. If you want to request a regrade for an objective grading error on our part, you must submit a request via

descope within **one week** of the assignment grade being released. If you're unsure how to submit a regrade reque a particular component of an assignment, it's fine to submit the request on the first problem of the assignment or r request as a private Piazza post.

egrade request may or may not result in a new grade being assigned. In some cases, a regrade request will be grar incur a point penalty. If we have to make a small manual change to your code to regrade your assignment success r grade for that component will take a 10 point penalty (so your max grade would be a 90%). If this change could l n avoided by running the **`diff`** program appropriately before submitting the assignment, a 20 point penalty will be lied (so your max grade would be an 80%). Finally, since written components are worth 10 points themselves, we accept regrade requests for written components that would require uploading a new PDF to Gradescope.

b Logistics

attendance is mandatory. Our weekly labs are designed to prepare students for the current homework. In lab, dents will work with a partner to complete a small problem or activity in the allotted time.

ally, you will finish the lab in the allotted time, but it's totally ok if you don't. Simply submit what you have at the en r lab session and, if you feel comfortable with the concepts involved, move on to your weekly homework assignme u're also welcome to keep working on the lab after your lab session, and bring lingering questions to office hours.

s are due by the Friday of the week the lab was released. As long as you've put in a good faith effort, you will rece nplete (100%) lab grade. Lab grades will be posted to Gradescope as a component of the weekly homework grade here to the same regrade deadlines described above.

chnical Resources

xtbook

s course has one optional textbook, which serves as a helpful supplementary reference: Problem Solving with C++ lter Savitch. ISBN: 0133591743 (7th Edition or higher).

azza

e preferred means of contacting the course staff is via Piazza, an online forum where students can ask and answer estions. General questions about the homework, course policies, C++, etc. should be posted publicly so that your ssmates can benefit from the answers and any resulting discussion. Before posting a question, please check to see ether your question has already been asked. Questions that are personal in nature, or that pertain to specific piece le that you have written should be posted privately to the course staff.

zza will host all of our major course announcements. It is your responsibility to check it regularly to avoid ssing any information.

ch Guide

u'll rely on a small number of programs and scripts to complete and submit labs and homework assignments. It is cial that you arrive at a basic understanding of these tools in order to complete your work without issues. You will l ut these tools during the first few weeks of class, but further descriptions and instructions for them can be found CS11 Tech Guide.

s and Office Hours

s course is challenging, but we want to help you succeed! If you need help understanding a concept, tackling an ignment, or dealing with a bug, you can participate in our teaching assistant (TA) office hours (OH) sessions, whic e place in the large room overlooking the baseball fields on the 4th floor of the Joyce Cummings Center, Room 40 en you arrive, you will "join the queue" by writing your name on a list maintained on a whiteboard. If you plan to wa newhere other than the OH room (which must be somewhere nearby), write your location next to your name and a come find you when it's your turn. The full **OH schedule** is posted and kept up to date on a pinned Piazza post.

assistance is meant to get you un-stuck or provide a nudge in the right direction; we expect that you will try to gra n your issue yourself before asking for help. TAs may turn away students who cannot provide evidence of attempti ve their problem on their own. For example, "What does this compiler error mean?" or "I don't understand this part assignment's directions."

: of respect for our TAs' time, office hours will end at the posted time regardless of the length of the remaining que e queue will be longer closer to the assignment deadline, so don't assume that you'll receive help if you join the qu ing the last 30 minutes of office hours on a Tuesday night.

ollaboration and Academic Integrity

ı are encouraged to discuss general CS11 concepts with *anyone*. This includes lecture slides and coding demos, la ıdouts, reading material, and C++ concepts (functions, pointers, classes, etc.) and syntax ("how do I write a for loc w do I create an integer variable?"). You are also welcome and encouraged to fully collaborate on labs with a partr

e collaboration policy for homework assignments is more nuanced. As described above, homework assignments w nprised of two components, a written component and a programming component. For each of them, we have sligh erent policies about which forms of outside help and collaboration are acceptable:

itten

e goal of our written problems is to give you practice finding pieces of information that you have not expressly beer en in class and lab. This skill allows you to eventually function independently as a programmer, and is critical for all nputer scientists to master. Thus, you are expected to find the answers to written problems yourself. You may **not** ıfer with classmates, and may only minimally query the TA staff. However, any means of finding an answer is fair ga ı may search the internet, consult the textbook, write test programs - anything.

ogramming

h your classmates and the internet can be invaluable resources when programming, so long as they are used ording to the following policies:

1. You may search the internet and talk to other students about general programing concepts and C++ syntax, but **not** about assignment-specific code or strategies. For example, it is fine to ask "How do you structure a 'while' loop in C++?", but it is **not** fine to ask "How do you write a Caesar Cipher decryptor in C++?" Assignment-specific questions should only be posed to the course staff.
2. You should never be looking at or discussing another student's code and no one outside of our course staff should be looking at or discussing your code. This also applies to testing and debugging: you may not help test or debug another student's code, and you may not receive help testing or debugging your code from

anyone other than the current course staff. If a TA sees you participating in any of these activities, it will be reported.
3. Only submit code that you can explain. We reserve the right at any time to ask you to explain a piece of code that you submitted. If you cannot explain the code, then we will have no choice but to assume that it is not your work.
4. Do not plagiarize code! Lifting partial or complete solutions from anyone (classmates, online sources, strangers) is completely prohibited. You must not submit code that others wrote or that you wrote for a previous class (or a previous iteration of CS 11).
5. No questions, student solutions, or instructor-provided solutions should be posted online in any capacity (except as a submission to Gradescope). This means that you are prohibited from posting any of your work for CS11 in a public Github repository.

y violation of the above policies will be considered cheating, and all students involved in any capacity will b warded directly to the Office of Student Affairs, who will investigate the case independently. Their sanction ge from horrible to inconceivably horrible. It's not worth it.

clusivity and Accessibility

spect is demanded at all times throughout the course. We realize that everyone comes from a different backgroun n different experiences and abilities. In the classroom, participation is encouraged, and should always be used to b ryone in the class.

ts University values the diversity of our students, staff, and faculty, recognizing the important contribution each dent makes to our unique community. Tufts is committed to providing equal access and support to all qualified dents through the provision of reasonable accommodations so that each student may fully participate in the Tufts erience. If you have a disability that requires reasonable accommodations, please contact the StAAR Center to ermine appropriate accommodations. Please be aware that **accommodations cannot be enacted retroactively**, king timeliness a critical aspect for their provision.

CS 15 - Spring 2025

Note: All lectures will be recorded and posted on the course Canvas page under "Echo360."

Week 1: 1/12/2025

Tuesday	**No lab held** **Lab:** (Optional) Quick Intro to Writing, Running, Submitting Programs
Wednesday	**Lecture:** Welcome - Introduction Read: CS15 Admin & Policies Browse: CS15 Reference Page Read: Shaffer 1.1-1.2 **Lecture:** C++ Review: Classes (and more!) employee code **Lecture:** ArrayLists IntArrayList.h vectors.cpp Read: Intro to Valgrind
Thursday	
Friday	

Week 2: 1/19/2025

Monday	No School
Tuesday	**Lab:** Unit Testing, ArrayLists **HW1 Out:** ArrayLists
Wednesday	**Substitute Monday Schedule** **Lecture:** The Big Three shallow_example.cpp deep_example.cpp **Lecture:** (Virtual Mini-Lecture) Exceptions Slides exceptions.cpp Read: CS 15 Style Guide
Thursday	
Friday	

Week 3: 1/26/2025

Monday	**Lecture:** Linked Lists 1 StringLinkedList.h Read: Shaffer 4.1.2, 4.1.5
Tuesday	**Lab:** Linked Lists **HW1 due by 23:59:59** **HW2 Out:** Linked Lists
Wednesday	**Lecture:** Linked Lists 2 **Lecture:** Makefiles make example code Read: Makefile Handout
Thursday	
Friday	

Week 4: 2/2/2025

Monday	**Lecture:** Complexity Read: Shaffer 3.0-3.5
Tuesday	**Lab:** make, diff **HW2 due by 23:59:59** **Project 1 Out:** MetroSim
Wednesday	**Lecture:** Queues Read: Shaffer 4.3 **Lecture:** Stacks paren_matching.cpp Read: Shaffer 4.2
Thursday	
Friday	**Lab:** (prelab) Circular buffers

Week 5: 2/9/2025

Monday	**Lecture:** File I/O isprime.cpp Read: File I/O Handout **Lecture:** Recursion
Tuesday	**Lab:** Stacks, Queues, and Circular Buffers
Wednesday	**Proj 1 checkoff and phase 1 due by 11:59pm** **Lecture:** Binary Search
Thursday	
Friday	

Week 6: 2/16/2025

Monday	No School
Tuesday	**Lab:** Recursion **Project 1 due by 23:59:59** **Project 2 Out:** CalcYouLater
Wednesday	**Lecture:** Project 2 Background **Lecture:** Trees int_tree_example.cpp
Thursday	**Substitute Monday Schedule: WE HAVE LECTURE!** **Lecture:** Binary Trees and their Traversals **Lecture:** Sets
Friday	

Week 7: 2/23/2025

Monday	**Lecture:** Binary Search Trees (BSTs)
Tuesday	**Lab:** Binary Tree Traversals
Wednesday	**Proj 2 design checkoff and phase 1 due by 11:59pm** **Lecture:** BST (cont'd) and AVL Trees Lecture Recording Read: Shaffer 13.2-13.2.1
Thursday	
Friday	**Lab:** (prelab) AVL Trees Preview

Week 8: 3/2/2025

Monday	**Lecture:** AVL (continued) **Lecture:** Templates
Tuesday	**Lab:** AVL Trees **Project 2 Due by 23:59:59**
Wednesday	**Lecture:** Midterm Info **Lecture:** Huffman Coding
Thursday	
Friday	

Week 9: 3/9/2025

Monday	**Lecture:** Midterm Review
Tuesday	**Lab:** Midterm Review (no spec)
Wednesday	Midterm
Thursday	
Friday	

Week 10: 3/16/2025

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

Week 11: 3/23/2025

Monday	**Lecture:** Priority Queues and Heaps **Project 3 Out:** zap
Tuesday	**Lab:** Heaps
Wednesday	**Lecture:** Hashes
Thursday	
Friday	

Week 12: 3/30/2025

Monday	**Lecture:** Hashes (continued)
Tuesday	**Proj 3 phase 1 due by 11:59pm** **Lab:** Hashes
Wednesday	**Lecture:** Intro to Graphs
Thursday	
Friday	

Week 13: 4/6/2025

Monday	**Lecture:** Graph Traversals
Tuesday	**Lab:** Graph Traversals **Project 3 due by 23:59:59** **Project 4 Out:** gerp
Wednesday	**Lecture:** Dijkstra's Algorithm
Thursday	
Friday	

Week 14: 4/13/2025

Monday	**Lecture:** Sorting I
Tuesday	**Lab:** Dijkstra's Algorithm Worksheet

Wednesday	**Proj 4 design checkoff and phase 1 due by 11:59pm** **Lecture:** Sorting II
Thursday	
Friday	

Week 15: 4/20/2025

Monday	No School
Tuesday	**Lab:** Sorting
Wednesday	**Lecture:** Sorting: Non-Comparison Sorts **Project 4 due by 23:59:59**
Thursday	
Friday	
Saturday	

Week 16: 4/27/2025

Monday	**Lecture:** Final Info **Lecture:** Wrap Up **Lecture:** CS 40 Preview
Tuesday	**Reading Period**
Wednesday	**Reading Period**
Thursday	**Reading Period**
Friday	**Final Exam (May 2nd)** Sec. 1: 12-2pm Sec. 2: 3:30-5:30pm

CS 105: Programming Languages

Spring 2025

https://www.cs.tufts.edu/comp/105

Class Sessions	Monday, Wednesday 3:00pm–4:15pm
	JCC 270
Instructor	Richard Townsend, richard.townsend@tufts.edu
Richard's Office Hours (JCC 440A)	Tuesdays 2:00pm-4:00pm
	(or by appointment)
Final Exam	Friday, May 2, 3:30pm-5:30pm, JCC 270

Course Overview

Summary

CS 105 introduces you—through extensive practice—to ideas and techniques that are found everywhere in today's programming languages. You will learn high-level, flexible programming skills that are applicable to older languages, popular modern languages, and even languages that don't yet exist. No matter what language you work in, when you finish 105, you'll be writing more powerful programs using less code. At the same time, you will learn the mathematical foundations needed to talk precisely about languages and programs: abstract syntax, formal semantics, and type systems.

You will explore and apply programming language concepts through multiple case studies, conducted using (mostly) tiny languages that are designed to help you learn. In any given case study, you may act as a practitioner (by writing code in a language), as an implementor (by working on an interpreter for the language), as a designer (by inventing semantics for a related language), or as a scholar (by proving mathematical properties of the language).

Logistics

The main point of access for the course is the website (the link is at the top of this document): it provides all the slides, readings, homeworks, solutions, and recitation materials.

Here is the weekly flow of the course:

1. Come to class sessions: take notes, participate in partner-based activities, and ask questions! This will ensure you're prepared for assignments and recitations. **Recitations will not be useful to you if you haven't come to class first.**

2. Read over the homework spec *before* attending recitation. Ask questions about it on Piazza and in office hours (OH), or talk about it with your classmates. Start the assigned reading.

3. Go to recitation, where you will get supervised practice working on problems that resemble the homework problems.

4. Work on your assignment, first by doing the reading and comprehension questions and then by completing the programming and/or proof problems. Make sure you spread this out so you have breaks between your 105 work. Post on piazza or come to OH if you need assistance.

If you feel like you're falling behind or are overwhelmed with 105, please reach out to a member of the course staff; we want to help you!

Prerequisites

- CS 15 Data Structures: You need substantial programming experience to keep up with the homework, particularly working with dynamically allocated (i.e., heap-allocated) data structures and recursive functions. We will also be referencing many classic data structures (stacks, queues, lists, trees) and algorithms (sorts, graph algorithms) throughout the course.
- CS/MATH 61 Discrete Math: You need some experience proving theorems, especially by induction. You should also be comfortable reading and writing formal mathematical notation.

Course Goals

By the end of this course, students should be able to

1. Design code using algebraic laws.
2. Read and write code that uses functional programming techniques.
3. Recognize the merits of polymorphism, type checking, and type inference.
4. Use functional and object-oriented language features to hide implementation details.
5. Understand precise specifications of how programming languages work.
6. Mathematically prove the behavior of programs written in a given language.

Technical Resources

Office Hours

This course is challenging, but we want to help you succeed! If you need help understanding a concept or tackling an assignment, or dealing with a pernicious bug, you can participate in our TA office hours sessions. You can also attend an instructor's office hours for any of these reasons, to discuss any issues you're having with the course, or just to say "hi!"

TA office hours will take place in the middle of the main hallway on the 3rd floor of the Cummings Center (near the stairs). When you want help, write your name and your place in the queue (as a number next to your name) on the whiteboard.

Office hour assistance is meant to help you along or provide a nudge in the right direction; we expect that you will try to grapple with your issue yourself before asking for help. To help you follow this practice, you are expected to provide a specific topic or question to receive help. Here are some *poor* examples of specific topics/questions: "why doesn't my program work?", "debugging", "can you explain impcore?". Here are *much better* examples: "why does my function produce a number

when it should produce a list?", "debugging a syntax error", "how do I call a function in impcore?".

The full office hours schedule is posted and kept up to date as a pinned Piazza post; make sure you check it before trekking to Cummings!

Piazza

We will be using Piazza for class discussion; you can get to the 105 page via this link or on the Home page of the course website. **Piazza will host all of our major course announcements; it is your responsibility to check in regularly to avoid missing any important information.**

In general, you should post all questions related to the course on Piazza; post publicly if at all possible, as other students may have similar questions or be able to help you faster than the course staff. Please post privately to the course staff if your question reveals individual work (e.g., algebraic laws or code) or concerns your grades.

Textbooks

This course has two required textbooks:

- *Programming Languages: Build, Prove, and Compare.* Norman Ramsey. Tufts University, Fall 2022.

  You must have this specific edition; if you're looking at an old version it may no longer align with this semester's assignments! The correct edition can be purchased from the Tufts Bookstore. If paying for this book is a financial hardship or you have any other questions about obtaining the textbook, please contact Sarah White (she's great) at s.white@tufts.edu.

- *Seven Lessons in Program Design.* Norman Ramsey.

  This short booklet is linked from our course website (at the bottom of the home page).

There is also one optional textbook that is useful for the few ML assignments we have:

- *Elements of ML Programming, ML97 Edition.* Jeffrey Ullman. Pearson, 1997.

Tentative Schedule

The tentative schedule for the course is displayed on our course website. Topics and assignments are subject to change, and may be updated as the semester progresses (you will be informed of any changes as soon as they happen). Typically: Lectures will be on Mondays and Wednesdays; homeworks will be due on Tuesdays at 11:59pm and new homeworks will go out Wednesday mornings; and recitations are either on Thursday or Friday (depending on which recitation you are enrolled in). **There are exceptions to all of these throughout the semester** (e.g., sometimes homework will be due on different days, etc.). Therefore, it is important for you to keep up with the schedule linked above.

Assessments and Grading

Your course grade will be produced as a weighted sum of your scores in four categories:

1. Recitations (5%)
2. Homework: Comprehension Questions (5%)
3. Homework: Programming and Proof Problems (70%)
4. Final Exam (20%)

All grades will be posted on Gradescope for students to review.

Although your ultimate course grade will be a conventional letter, much of your specific work will be graded on a scale of No Credit, Poor, Fair, Good, Very Good, and Excellent. These grades correspond to numeric values of 0, 65, 75, 85, 95, and 100, respectively. For example, if an assignment has 3 equally weighted components and you received a Good on two parts and a Very Good on the third, your grade for that assignment would be approximately .33 * 85 + .33 * 85 + .33 * 95 = 87%. In a typical class, a consistent record of Very Good work will lead to a course grade in the A range. Work rated Good corresponds to a wide range of passing grades centered around B. Work rated Fair will lead to low but satisfactory course grades around a C; if a significant fraction of your work is Poor, you can expect an unsatisfactory grade.

Recitations

Recitations are graded as Attended or Not Attended (a recitation TA will take attendance in each session); your final recitation grade is the percentage of recitations you attended. We drop your two lowest recitation grades, effectively letting you miss two recitations at no penalty. There is no such thing as an "excused absence" from recitation, so if you need to miss one just let your recitation leader know. If you need to miss more than two recitations due to a personal emergency, please explain the situation to your academic dean and ask them to contact a course instructor.

Homework

In general, homeworks will go out every Wednesday and be due at 11:59:00PM EST the following Tuesday (there will be a few larger assignments in the second half of the course that are exceptions to this rule). Each homework will have two components:

1. Your responses to a set of reading comprehension questions (CQs), which will be submitted as a .txt file.

2. A programming and/or proof component.

You will submit all of your work directly to Gradescope. No submissions will be accepted via email or other means. To avoid missing the strict 11:59:00PM EST deadline, you should submit whatever you have a bit earlier (or much earlier!) even if incomplete; you may submit as many times as you wish before the deadline. We will always grade the latest submission.

After submitting to Gradescope, you must always look over the files you submitted to make sure that they are what you expect. In particular, make sure: (1) you submitted all necessary files; (2) the code/files you submitted are the latest versions of your work; and (3) any PDFs you submitted are readable and not corrupted. (3) is important, since VSCode has been known to corrupt PDFs that are synced with the department servers. **It is your responsibility to check that the files you submitted are the correct and readable versions. Regrade requests will not be considered for submitting incorrect files.**

Comprehension Question Grades A set of comprehension questions (CQs) is associated with each homework assignment. Your answers to these questions are graded based on the number of questions answered and the correctness of each answer, roughly following this scale:

- **Excellent**: All answers are completely correct.
- **Very good**: All answers are mostly correct.
- **Good**: A majority of answers are mostly correct.
- **Fair**: Most answers are incorrect, but all were reasonably attempted.
- **Poor**: Most answers are incorrect or blank.
- **No Credit**: No answers are provided.

Programming and Proof Problem Grades The programming and proof problems on each assignment will be graded both for correctness (e.g., Do your programs pass our automated tests? Are your theoretical proofs and rules accurate and complete?) and for structure and organization (e.g., Does your code follow our course coding standards? Are your algebraic laws accurate and complete?). Correctness is typically evaluated by automated testing scripts that list which tests you passed or failed and why. Structure and organization is evaluated manually by our course staff. Both of these components are graded with a coarse five-point scale:

- **Excellent**: Your submission is outstanding in all respects.
- **Very Good**: Your submission does everything asked for, and does it well. There may be something small that could be improved.
- **Good**: Your submission demonstrates quality and significant learning.
- **Fair**: Your submission is quite lacking in one or more aspects; key issues need to be addressed. This is the lowest satisfactory grade.

- **Poor**: Your submission shows little evidence of effort or has other serious deficiencies. This is an unsatisfactory grade.
- **No Credit**: No submission provided OR the submission exhibits plagiarism OR the submission violates a rule that the homework indicated would result in no credit.

Late Policy

For every assignment we have, you are allowed to submit up to 24 hours after the posted deadline. There is no penalty for submitting within this 24-hour grace period. However, typically a new assignment will be released immediately after each deadline—so by submitting late, you lose some time on the next assignment. If you submit more than 24 hours after a deadline, your submission will not be accepted.

You should think of these late submissions as extensions you have been granted ahead of time and use them when you might have otherwise tried to ask for an extension. There are no additional "late tokens" in this course. You can think of our late policy as granting one "late token" per assignment.

If you experience an extraordinary difficulty, such as serious illness, family emergencies, or other extraordinary unpleasant events, your first step should be to contact your advising dean as soon as you can: explain the situation to them and ask them to contact a course instructor. Your dean will work with the instructor to make appropriate arrangements. **IMPORTANT: The earlier you notify your dean, the more flexibility the course staff will have to make appropriate arrangements.**

Regrades and Grade Explanations

If you want to request a regrade for an objective error on our part, you must submit a private request to the Instructors via Piazza; provide your UTLN and explain the error. The deadline for these requests is **one week** from the time the grades were posted for a given assignment; no exceptions to this policy will be made. A regrade request may or may not result in a new grade being assigned.

If a regrade request requires the course staff to make a manual modification to your submission, your grade for the component being regraded will be capped at a "Good" as a penalty. **IMPORTANT: If you misname a function, you could face a significant penalty from the autograder. Therefore it is crucial that you check that all function names match what the spec calls for exactly.**

You are always welcome to ask for an explanation of the grade you received. We want to help you understand how your work was evaluated and how you can continually improve in the course!

Collaboration and Academic Honesty

If you have **any** questions about the below policies or a specific situation dealing with academic integrity, do not hesitate to ask a course instructor. All students are expected to read and adhere to the Tufts Academic Integrity Policy.

Collaboration

In general, you are encouraged to discuss the lecture content, reading material, and what's being asked of you on assignments with your classmates. However, **you must not discuss anything that you produce for an assignment with classmates in any form (e.g., English, pseudocode, pictures, etc.).** Examples include proofs, code, algebraic laws, and function contracts. You may not show your work at this level to other students nor may you look at others' work at this level. This also applies to testing and debugging: you may not test or debug another student's code or let them test or debug your code. Due to this policy, any Piazza post that discloses your work on an assignment must be posted privately.

Lifting or looking at partial or complete solutions from anyone (classmates, online sources, strangers) is completely prohibited. You may not work on the specific problems with anyone other than TAs or a course instructor, and you should not use the internet in any capacity to help solve any specific problem. No homework problems, student solutions, or instructor-provided solutions should be posted online in any capacity (except as a submission to Gradescope); **this means that you are prohibited from posting any of your work for 105 in a public Github repository**.

If an assignment contains a *pair programming* component and you choose to pair program, you are welcome to discuss/view any part of that component with your partner, but not any other students.

Finally, you **may not** use ChatGPT, Copilot, or similar large language models to assist in completing your homework. Our assignments exist for your benefit—by relying on AI to do your work for you, you only waste your own opportunity to learn.

Policies for Students Retaking CS 105

If you are repeating the course or any part of it (e.g., you withdrew or dropped after doing one or more assignments), you must let a course instructor know at the beginning of the semester. In this case, you are expected to abide by the following policies:

- There is no acceptable use of instructor-provided solutions from prior semesters. If you have kept any such solutions, destroy them.
- If you have written your own solutions for past semesters, it is acceptable to consult them for ideas, and it is acceptable to submit parts verbatim. Such use must be explicitly acknowledged in a README.
- In every homework, your README file must note what work is new, what work is based on work from a prior semester, and what work is submitted verbatim from a prior semester. Even if nothing is from a prior semester, your README file must disclose this information.
- In every homework, you must cite collaboration with every partner with whom you worked on the assignment in a past semester—even if none of that partner work survives.

Academic Honesty

Any violation of the above policies will be considered cheating, and all students suspected of being involved in any capacity will be forwarded directly to the Office of Student Affairs, who will investigate the case independently. Their sanctions range from

horrible to inconceivably horrible. It's not worth it. We do use automated tools to detect plagiarism as well.

Inclusivity, Accessibility, and Additional Help

This course strives for inclusion of all participants, regardless of personal identity (gender, race, sexual orientation, religion, etc.), socio-economic background, disabilities, or neurodivergence. In the classroom and our discussion forums, everyone is expected to treat everyone else with dignity and respect. If you feel unwelcome or mistreated for any reason by either another student, a TA, or the course itself, please let an appropriate member of the teaching staff know so we can work to make things better.

CS 105 can be especially difficult for first-generation college students and for members of historically underrepresented groups in computing, who may not have the family or social support that helps them develop their skills in "how to be a college student." If you are a student in either of these categories, you are strongly encouraged to meet with a course instructor or contact one early in the term to talk about your support system.

Accommodations for Students with Disabilities

Tufts University values the diversity of our body of students, staff, and faculty and recognizes 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 StAAR Center at StaarCenter@tufts.edu or head to the StAAR Center website to make an appointment with an accessibility representative to determine appropriate accommodations. **Please be aware that accommodations cannot be enacted retroactively; all accomodation notes must be provided to the instructor within one week of the note being written to guarantee consideration.**

Academic Support at the StAAR Center

The StAAR Center offers a variety of resources to all students (both undergraduate and graduate) in the Schools of Arts and Sciences, and Engineering, the SMFA, and The Fletcher School; services are free to all enrolled students. Students may make an appointment to work on any writing-related project or assignment, attend subject tutoring in a variety of disciplines, or meet with an academic coach to hone fundamental academic skills like time management or overcoming procrastination. Students can make an appointment for any of these services by visiting tutorfinder.studentservices.tufts.edu, or by visiting students.tufts.edu/staar-center.

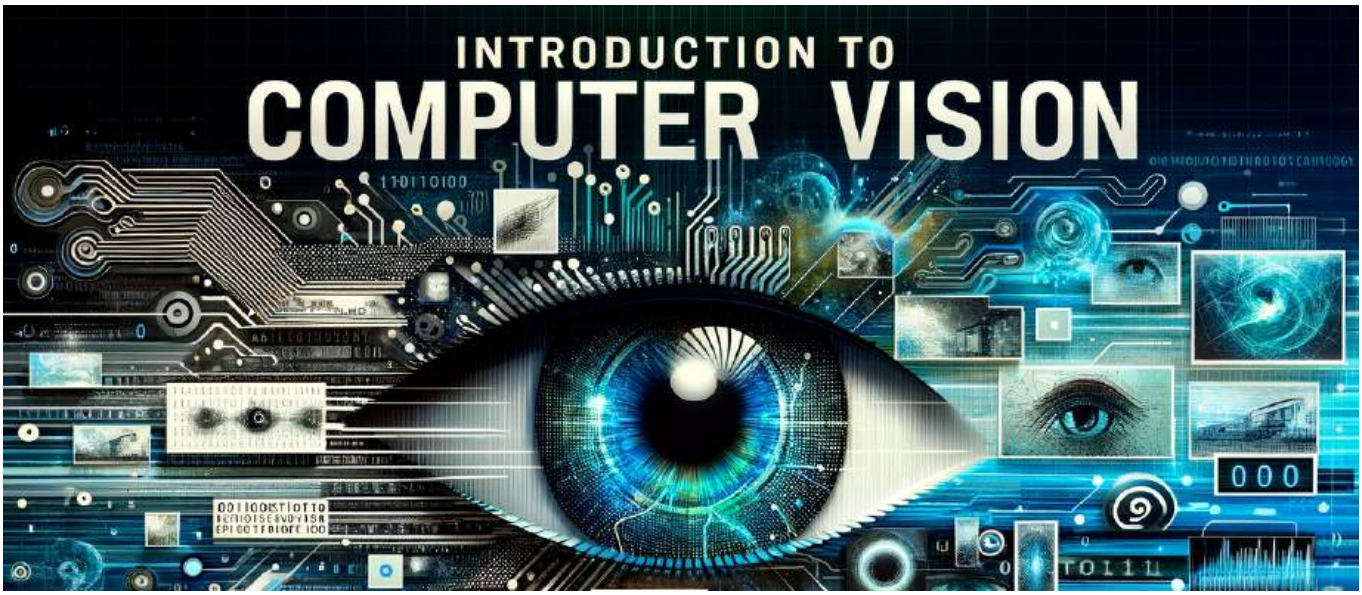
Important note on exam accommodations: All students approved for extended-time and/or distraction-reduced environment on their exams must book their exam space at the StAAR Center, no later than 10 days before the exam date. Students who do not comply with these guidelines and timeframes set by the StAAR Center may have to take their exam without any accommodation.

Religious Holy Days

We will reasonably accommodate any student who, for reasons of observing religious holy days, will be absent from recitation or experience any hardship in the completion of their work during the holy days. Please contact the instructor in advance of the holy day if you will miss recitation or if you need any additional accommodation (such as for assignments); reasonable accommodations will be allowed at no penalty **only if the instructor is notified well in advance**.

Acknowledgments

This syllabus (and the entire course) uses many ideas and materials developed by previous instructors of CS 105, including Norman Ramsey, Kathleen Fisher, Jeff Foster, and Richard Townsend. Thank you!



Introduction to Computer Vision

CS152-02 DS-153 | Spring 2024

This course is an introduction to low and intermediate level of classic and modern Computer Vision. We will learn how to design algorithms that process visual scenes to automatically extract information. The course will cover fundamental principles and important applications of computer vision, including image formation, processing, detection and matching features, image segmentation, and multiple views. We will also cover the basics of machine learning and deep learning for computer vision. The course will include a mix of lectures, paper readings, and hands-on Python programming assignments.

Cover photo hallucinted by ChatGPT

Course Information

Hours: Monday & Wednesday, 4:30PM – 5:45PM

Location: Joyce Cummings Center, room 160

Instructor: Dr. Roy Shilkrot roy.shilkrot@tufts.edu | roy@cs.tufts.edu | LinkedIn | GitHub | Twitter/X | YouTube

Office hours: Schedule online with Prof. Roy: https://calendly.com/royshilkrot/15min

Prerequisites: Algorithms and data structures, linear algebra (I or II), statistics and probability theory (I), calculus (I), working knowledge of Python

Teaching assistant:

- Yukun Li Yukun.Li@tufts.edu
- Fox Huston fox.huston@tufts.edu

TA Office hours: TBD

Piazza: https://piazza.com/tufts/spring2024/sp24cs015202ds0153/home

Canvas: https://canvas.tufts.edu/courses/54844

Course Goals

By the end of the semester, students should be able to:

- Identify and understand the fundamental aspects and techniques of Computer Vision
- Interpret and analyze a given problem using these techniques
- Adopt and Implement the appropriate techniques to solve a given problem

Textbooks

We will follow chapters from the books below:

- Computer Vision: Algorithms and applications, 2nd Ed (2022). Richard Szeliski. The book PDF is freely available online (http://szeliski.org/Book)
- Simon J.D Prince. Computer Vision Models (2012)
- Aeorlien Geron. Hands on Machine Learning 3rd ed. (2022)
- Francois Chollet. Deep Learning with Python 2nd ed. (2021)
- Excerpts from Hartley & Zisserman (2004), and Marr (2010)

As well as reference videos from Prof. Shree Nayar's lecture series First Principles of Computer Vision (https://www.youtube.com/@firstprinciplesofcomputerv3258)

Syllabus

Class	Content	Links	Assignment
1	Introduction: Class, Computer Vision, Math	slides video	
2	Image Formation: Human Vision, Optics, Physics, Cameras	slides video	
3	Image Formation cont.: Perspective projection, Pinhole camera model, Digital Imaging, Colorspaces	slides video	HW1: Hello Vision World
4	Image Processing: Sampling, Filters, Edges: Convolutions, Separability, Local operators, Frequency domain and Fourier Transform, Histograms, non-linear (median, bilateral), Pyramids	slides video	
5	Features: Interests points, Corners, Harris, Blobs, FAST, Multiscale	slides video	HW2: Pyramid blending, Corners & Blobs, Key points
6	Features: Descriptors, Gradient Histograms, Edges & Contours, Binary Images	slides video	

Class	Content	Links	Assignment
7	Geometric / Planar Transforms Parameterization, Feature Matching, Least Squares optimization	slides	HW3: Feature Extraction & Matching, Binary Images
8	Image Alignment, RANSAC, Stitching, Panoramas, Cylindrical coordinates, Global optimization / Bundle Adjustment, Blending	slides video	
9	Tracking: Meanshift, Motion estimation parameterizations	slides video	HW4: Panorama Stitching
10	Tracking: Correlation Filters, Optical flow, Lucas-Kanade, Shi-Tomasi	slides video	
11	Multiview 1: Intro, Camera Calibration, Calibration and Projection Matrices	slides video	
12	Multiview 2: Parallel Motion Stereo, Depth and Disparity, Epipolar Geometry, Essential and Fundamental Matrices	slides video	
13	Multiview 3: Triangulation, Camera Pose Estimation, Augmented Reality	slides video	HW5: Tracking
14	Mid-term (in class)		
15	Bundle Adjustment, Intro to Visual Machine Learning	slides video	
16	Linear Models, Linear Regression and Classification, Logistic Regression	slides video	HW6: MVG, Bundle Adjustment
17	Linear SVMs, Neural Networks, Gradient descent, Backpropagation and Learning Practices	slides video	
18	Convolutional Neural Networks, Deep Learning	slides video	HW7: Hello CNNs, Classification
19	Object detection 1: V-J, Intro to CNN object detection	slides video	
20	Object detection 2: Region Proposals Nets, R-CNN, SPP, Fast-RCNN, YOLO, SSD	slides video	
21	Segmentation I: Clustering, Flood-fills, Superpixels, Graph-cuts	slides video	HW8: Object Detection
22	Segmentation II: Semantic, FCNs, Losses, U-Net, Transpose Conv	slides video	
23	Vision and Language: Captioning, VQA, CNN+RNN	slides video	
24	Pose Estimation I: ASM, AAM, DPM, 2.5D Pose	slides video	HW9: Segmentation and Pose
25	Pose Estimation II: DeepPose, Heatmaps, PAFs, HRNet	slides video	

Class	Content	Links	Assignment
26	Final class. What have we missed? Further reading, final exam prep Final exam: 5/8 from 12:00pm to 2:00pm, JCC 160		

Problem Sets, Tests

Problem sets: Eight (8) (or Nine (9)) problem sets will be given during the class. Any material regarding the set solution will be submitted electronically. Homework will be due up to one or two weeks from the assigned date at midnight. Late assignments are penalized at 20% for each 24 hours' delay. No homework will be accepted one week after the deadline. If a serious illness or another major life event prevents you from completing a homework assignment on time, you should report the event to your instruction team, after which alternate arrangements can be made. Reporting must be done before the assignment in question is due.

Tests: There will be two (2) tests over the course of the semester. The tests will be a combination of multiple choices and open-ended answers. If, for any serious illness or major life event you must miss a test, you must inform me before the day of the exam so we can work out an arrangement. Please Note: If you wish to dispute a grade, it is mandatory that you do so within one week of receiving the grade. After such a term, the grade will be considered final. Note: Grading is generous, re-grading is strict.

Final grades: You must show proficiency in all grading areas to pass the class. A failing average (below 50%) in any of the grading areas (problem sets, tests, attendance) will result in a failing grade in the class. Your final grade will be determined using the following percentage breakdown:

Grading Area	Percentage
Problem Sets	50%
Tests	40% (mid 13%, final 27%)
Attendance, Behavior, Attitude, Decorum	10%

Communications

Piazza will be our primary means of communication. All course announcements will also be made through Piazza. Rather than emailing questions to the teaching staff, we encourage you to post your questions as a public discussion. You are also encouraged to help each other, as long as the question does not contain any code or portion of a problem set answer. In such a case, the question must be made private (please, refer to the section Academic Honesty below). To schedule office hours outside the posted times, please email any of the teaching staff or post a private message on Piazza.

Academic Honesty

Science is, to its core, a collaborative effort. The advantages of coming together for examination or comparison, sometimes even just to explain the problem, are well known. I strongly encourage students to discuss course material, problems, and applications outside the classroom with the teaching staff and other students. You are also encouraged to form study groups for the tests.

Having said that, integrity and honesty are equally important qualities of any future academic, scientist or engineer. We take plagiarism very seriously. You must do homework, the problem sets and the final projects on your own. If you need help, the teaching staff will be more than happy to help you!

Using AI. We encourage and expect you to use AI tools such as GitHub Copilot and ChatGPT to aid you in this class, as would be expected of you by your future employers or colleagues. However, understand the risks of using these tools without a solid understanding of their inner workings. Research hallucinations and AI plagiarism before using any products of AI tools as submissions in this class. OpenAI has helpful information on the topic, as well as the faculty in Wharton.

Academic Integrity Policy

Tufts holds its students strictly accountable for adherence to academic integrity. The consequences for violations can be severe. It is critical that you understand the requirements of ethical behavior and academic work as described in Tufts' Academic Integrity handbook. If you ever have a question about the expectations concerning a particular assignment or project in this course, be sure to ask me for clarification. The Faculty of the School of Arts and Sciences and the School of Engineering are required to report suspected cases of academic integrity violations to the Dean of Student Affairs Office. If I suspect that you have cheated or plagiarized in this class, I must report the situation to the dean.