

Fall 2026 COMS E6998

Advanced Topics in Competitive Programming

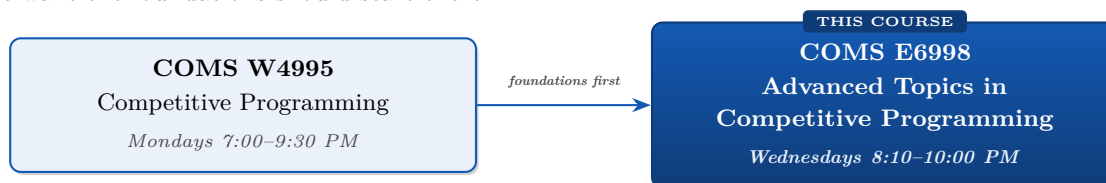
Tentative Syllabus

Last Updated: July 18, 2026

Course Description

This course is the advanced sequel to Competitive Programming (COMS W4995). The topic of the course is **Advanced Topics in Competitive Programming**: techniques that appear in the harder half of ICPC regional, championship, and world-finals problem sets, and in Division 1 Codeforces contests. Building on the foundations of the introductory course, we study advanced algorithm design and optimization techniques, and place a strong emphasis on speed, rigor, and contest execution under time pressure.

The introductory course, COMS W4995 Competitive Programming, is also offered this fall (Mondays); students who first want the foundations should start there.



Our Commitments to You

This syllabus is intentionally comprehensive: every policy, requirement, and grading rule for the semester is written here, on day one. There will be no administrative surprises. In addition, the instructors commit to the following:

- **Predictable grading:** The complete grading rubric (see “Grade Breakdown”) is published on day one and is **fixed for the semester**. No grading policy will change after the first week of classes except in truly unavoidable circumstances, and any such change will only ever be in students’ favor.
- **Timely feedback:** Your recorded progress (problems, contests, attendance, presentations) is updated on CourseWorks **within one week**, so you always know where you stand.
- **No hidden rules:** Attendance implications, citation requirements, academic honesty procedures, and contest requirements are all stated in this document and the Course Contract. If a rule is not written down here, it will not be used against you.
- **Rigor with support:** This course is intentionally demanding, and we will not water it down. But difficult is not the same as unforgiving: being stuck is a normal, expected part of competitive programming. Most students will *not* solve every problem on every practice set — that is by design. If you have been stuck on a problem for an hour of honest effort, that is normal; ask for help in office hours or on the course forum.

How This Course Has Changed Based on Student Feedback

We take course evaluations seriously. Concrete changes made in response to past student feedback:

- Fixed grading rubric published on day one; no mid-semester grading changes
- Progress and grades updated weekly on CourseWorks
- Added an explicit C++ readiness self-check and prerequisite warm-up plan (see “Prerequisites”)
- Slower, from-scratch walkthroughs of representative problems in lecture, including *why* tempting-but-wrong approaches fail
- Every lecture ends with a guaranteed-solvable exercise and clear takeaways, so each week includes a concrete win
- Two co-instructors and a TA offering office hours, for greater support capacity

Contact Information

Co-Instructor: [Josh Alman](#)

Email: josh@cs.columbia.edu

Office Hours: By appointment (contact via email).

Co-Instructor: [Christian Yongwhan Lim](#)

Email: yongwhan.lim@columbia.edu

Office Hours: By appointment via [Calendly](#) (typically held via Zoom).

Teaching Assistant: TBD

Email: TBD

Office Hours: By appointment (contact via email).

Lectures

Wednesdays, 8:10 PM–10:00 PM ET (room TBD)

First lecture: Wednesday, September 9, 2026. Last lecture: Wednesday, December 9, 2026. No class on Wednesday, November 25, 2026 (Academic Holiday; Thanksgiving is Thursday, November 26).

Course Website

All course materials will be posted on Columbia CourseWorks (link TBD).

Access Instructions:

- Log in using your Columbia UNI and password
- Navigate to the course page or search for “COMS E6998 Advanced Topics in Competitive Programming” in CourseWorks
- Course materials include: weekly problem lists, lecture slides, presentation guidelines, and announcements
- If you cannot access the course page, contact the instructors immediately
- All assignments and submissions will be tracked through CourseWorks

Progress Tracking:

- Problem submissions: Track your progress toward the 150-problem requirement through CourseWorks. Submit Codeforces problem links with AC verdict for verification.
- Contest participation: Submit contest participation records (Codeforces contest links showing your username and solved problems) through CourseWorks after each contest.
- Regular updates: Check CourseWorks weekly for your current progress status. Contact the instructors or TA if you notice any discrepancies.

Exams

No midterm or final exam.

Grade Options

Letter grading is strongly encouraged. P/D/F is available upon request.

External Resources

Curated list of useful websites: [Useful Websites Document](#)

Terse Guides

Additional study materials: [Terse Guides Folder](#)

Prerequisites

This is an **advanced** course. Students are expected to arrive already comfortable with the core competitive programming toolkit (dynamic programming, graphs, standard data structures, basic number theory and combinatorics).

- Competitive Programming (COMS W4995; offered most semesters, including Fall 2026) or equivalent experience is strongly recommended
- Cumulative GPA of 3.6+ is highly recommended
- Recommended coursework:
 - COMS W3134: Data Structures in Java or COMS W3136: Essential Data Structures in C/C++
 - COMS W3203: Discrete Mathematics
 - CSOR W4231: Analysis of Algorithms I
- Proficiency in C++ strongly recommended (Java or Python acceptable)
- Exceptions granted case-by-case. “Equivalent experience” may include:
 - Similar coursework at another institution
 - Competitive programming experience (Codeforces rating, ICPC participation)
 - Significant industry experience in algorithm design

Contact the instructors before enrolling if you believe you have equivalent experience.

C++ Readiness — read this before enrolling: This class assumes you can already write basic C++ comfortably (standard I/O, STL containers, writing and debugging a 50-line program without reference material). If you cannot yet do this, you are not excluded — but you should spend the **first two weeks** completing the prerequisite exercises in the [Terse Guides Folder](#) (C++ guide) alongside the regular coursework, and you should expect the early weeks to be significantly more work. If you are unsure whether you are ready, email the instructors before the semester begins and we will help you assess honestly.

Course Objectives

By the end of this course, you will have developed the advanced knowledge and problem-solving skills necessary to:

- Compete at a high level in the ICPC contest series, whose early stages take place during the fall semester (see the pathway below)
- Solve the harder problems on ICPC regional and championship problem sets
- Excel in the most demanding technical interviews at leading technology companies and quantitative trading firms
- Push into the top rating bands on major platforms (Codeforces ~purple/Candidate Master and beyond, AtCoder ~blue and beyond)

The ICPC Pathway:



Course Structure

To earn full credit (100%), fulfill the following requirements:

1. Attend all Wednesday lectures (up to 3 absences permitted without penalty)
2. Solve at least 150 problems on Codeforces from assigned problem lists
3. Solve at least one problem correctly in at least 10 live Codeforces contests (partial credit available for 8–9 contests)
4. Participate in the Columbia University Local Contest (CULC)
5. Complete at least one satisfactory presentation (see “Grade Breakdown” for the satisfactory criteria)

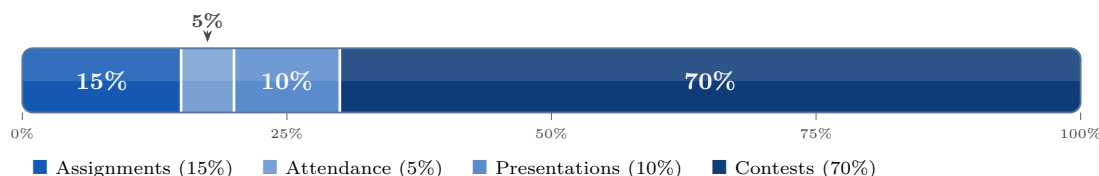
Lecture Format

- **Schedule:** Wednesdays, 8:10 PM–10:00 PM ET (110 minutes), room TBD. Attendance is required and recorded.
- **First Half:** Instructor lecture on advanced programming contest topics. Rather than presenting only polished solutions, each lecture includes at least one representative problem solved **entirely from scratch**, with an instructor narrating the thought process — including the natural approaches that *fail* and why — before arriving at the correct solution. Homework problems are assigned and due the following Wednesday before class.
- **Closing Every Lecture:** Each lecture ends with (1) one guaranteed-solvable exercise so every student leaves with a concrete win, (2) the key takeaway of the week, and (3) an honest difficulty calibration for the week’s problem set (e.g., “most students will not solve Problem E — that is expected”).
- **Second Half:** Weekly group sessions (groups of 5–7 students) with timed practice problems. Your group will work collaboratively on problems:
 - First 45 minutes: Work out solution ideas for most problems
 - Remaining 10 minutes: Each group selects a problem to present to the classAfter the session, students select a problem via Google Form to present the following week.
- **Homework Substitution:** Each homework problem can be replaced with 10 Kattis points.
- **Presentations:** Based on practice contest problems. Submit presentation materials before class on presentation day. See “Grade Breakdown” for evaluation criteria.

Grade Breakdown

The rubric below is complete and **fixed as of day one**: it will not change after the first week of classes (see “Our Commitments to You”). All recorded progress is updated on CourseWorks within one week.

At a glance:



- **Assignments:** 15% (completion of at least 150 homework problems on Codeforces)
 - **Grading:** You must achieve AC verdict on Codeforces for a problem to count. Problems must be from the assigned problem lists provided each week.
 - **Minimum Threshold:** 150 problems is the target for full credit. Solving more helps understanding (see

“Incremental Progress Benefits” below for additional benefits).

- CULC problems count toward this requirement (see “ICPC Columbia University Local Contest (CULC)” section for details)
- **Attendance:** 5%
 - Required for all Wednesday lectures (recorded)
 - Up to 3 absences permitted without penalty
 - Additional absences result in gradual grade reduction:
 - * 4 absences: 1 letter-grade reduction (e.g., A $\rightarrow$ A-)
 - * 5 absences: 2 letter-grade reductions (e.g., A $\rightarrow$ B+)
 - * 6 absences: 3 letter-grade reductions (e.g., A $\rightarrow$ B)
 - * 7+ absences: Additional reductions continue (B $\rightarrow$ B- $\rightarrow$ C+, etc.)
 - Grade reductions applied after calculating grade based on all other components
 - Extenuating circumstances (medical, family emergencies) may be excused with documentation
- **Presentations:** 10% (based on best presentation performance)
 - **Grading:** Grade based on **best** presentation, not average. Satisfactory presentations earn full credit (10%); unsatisfactory presentations earn 0%.
 - **Satisfactory Criteria:** All of the following must be met:
 - * Solution correct with AC verdict on Codeforces
 - * Clearly explains problem statement and constraints
 - * Clearly explains solution approach (key algorithmic ideas, data structures)
 - * Includes clear time and space complexity explanation
 - * Demonstrates understanding by answering questions
- **Contests:** 70% (participation in at least 10 live Codeforces contests, solving at least one problem correctly in each contest)
 - **Grading:** Meeting the minimum requirement (10 contests with at least one solved problem each) earns full credit (70%).
 - **Performance Beyond Minimums:** While meeting the 10-contest minimum earns full credit, strong performance (consistent problem-solving, rating improvements, solving multiple problems per contest) demonstrates mastery and is strongly considered for A+ evaluation and in borderline grade cases

Extra Credit: May be available for attending external contests (beyond Codeforces and AtCoder). Details announced as opportunities arise.

Note: Meeting minimum requirements ensures passing grades. Exceeding requirements demonstrates mastery and is considered for A+ evaluation.

Incremental Progress Benefits: While grading is threshold-based (with proportional partial credit below the minimums for Assignments and Contests), exceeding minimum requirements provides several benefits:

- **Buffer for Errors:** Solving more than 150 problems provides a buffer if some submissions are later found to be invalid or if there are tracking discrepancies.
- **A+ Consideration:** Strong performance beyond minimums (e.g., solving 200+ problems, participating in 15+ contests, achieving significant rating improvements) is strongly considered for A+ evaluation.
- **Mastery Demonstration:** Exceeding requirements demonstrates deeper understanding and commitment, which may be considered in borderline grade cases.
- **Learning Outcomes:** The primary benefit is improved problem-solving skills and better preparation for ICPC and technical interviews.

A+ Grade: Awarded at the sole discretion of the instructors, typically for students who significantly exceed minimum requirements and demonstrate exceptional mastery of advanced competitive programming concepts.

Live Contests

Participation in live Codeforces contests is required. Full immersion in timed environments with real contest problems provides the most effective preparation.

Important: Only live participation during scheduled contest time counts. Virtual participation after the contest ends does not count.

Requirements:

- Participate in at least 10 live Codeforces contests during the semester
- Must solve at least one problem correctly (AC verdict) in final results to count
- **Contest Substitution:** Codeforces contests can be substituted with AtCoder contests. For AtCoder Beginner Contests, you must solve at least 2 problems (not 1)
- **Division 2 Bonus:** Solving 4+ problems in a Division 2 Codeforces contest counts as 2 contests

Rules: Review Codeforces Contest Rules by Mike Mirzayanov, especially the “Can-do’s and Can’t-do’s” section. Violations of contest rules or academic integrity policies will result in grade penalties and may be referred for academic integrity violations. Please review these rules carefully before participating.

Textbooks

Recommended (not required) supplementary materials:

- *Competitive Programming 4*, Steven Halim, Felix Halim, and Suhendry Effendy
- *Guide to Competitive Programming*, Antti Laaksonen
- *Competitive Programming Beyond the Basics*, Josh Alman and Christian Yongwhan Lim (preprint)

ICPC Columbia University Local Contest (CULC)

In-person participation is required. The fall edition of the contest will be held early in the semester (date and location TBD) and serves as a tune-up for the ICPC North America Qualifier and the Greater New York Regional.

- Each correctly solved problem counts as two homework problems toward the 150-problem requirement (which contributes to the 15% Assignments component)
- Individual contest (not a team event)
- This is a mandatory course requirement — mark your calendar accordingly
- Failure to participate without prior approval for extenuating circumstances will result in significant grade penalties

Course Contract

All enrolled students must sign the [Course Contract](#) by the first Friday after enrollment. The contract outlines additional policies, expectations, and academic integrity requirements.

Key Points from Contract:

- Academic integrity expectations and citation requirements
- Course participation and engagement standards
- Communication protocols and response time expectations
- Additional contest rules and platform-specific guidelines

Allowed Languages

C++ (highly preferred), Python, or Java are strongly encouraged. If you plan to use another language, discuss with the instructors before submitting assignments during the first week.

Policy on Academic Honesty

Students must complete their own work and adhere to the Faculty Statement on Academic Integrity and the Honor Code. Violations (cheating, plagiarism, dishonesty) will result in grade penalties and may be referred to Dean's Discipline.

External Resources and Citation: Consulting external resources (programming libraries, online solutions, discussions with classmates) is acceptable, but you must cite all sources explicitly at the top of your submission or presentation (e.g., as a comment). Proper citation is good academic practice. Your grade reflects the quality of your work and understanding, even if sources are cited.

The primary goal is learning and understanding, not submitting borrowed solutions. You must fully understand any problem you solve and be able to explain your approach clearly.

Important: Failure to cite external resources will result in grade penalties and may be referred to the appropriate office.

Using templates from online sources is permitted if properly cited.

If you have questions or concerns regarding academic integrity:

- Consult with your TA
- Contact the instructors
- Refer to the Columbia University Undergraduate Guide to Academic Integrity

See also the Course Contract for additional details.

Course Policies

Late Submission Policy

- **Homework Problems:** Due before class on the specified due date (typically Wednesday before class). Late submissions are not accepted without prior approval from the instructors.
- **Extenuating Circumstances:** Contact the instructors as soon as possible to discuss accommodations. Documentation may be required.
 - **Medical emergencies:** Illness, injury, or hospitalization (doctor's note or medical documentation required)
 - **Family emergencies:** Death or serious illness of immediate family member (documentation may be required)
 - **Technical issues:** Codeforces platform outages or verified technical problems preventing submission (screenshot or evidence required)
 - **University obligations:** Official university events, academic conferences, or required academic activities (advance notice and documentation required)
 - **Religious observances:** Conflicts with religious holidays (advance notice required)
 - **Other circumstances:** Case-by-case evaluation. Contact the instructors immediately to discuss.
- **Request Process:** For extenuating circumstances, email the instructors as soon as possible (ideally before the deadline) with:
 - Clear explanation of the circumstance
 - Requested accommodation (e.g., extended deadline, alternative assignment)
 - Supporting documentation (if applicable)
 - Proposed timeline for completion
- **Note:** Extenuating circumstances do not include general workload, other course deadlines, personal travel, or poor time management. Plan ahead and manage your schedule accordingly.

Disability Accommodations

Columbia University is committed to providing equal access to qualified students with disabilities. If you require accommodations, contact the Office of Disability Services (ODS) at disability@columbia.edu or visit <https://health.columbia.edu/content/disability-services>.

Once approved by ODS, provide your accommodation letter to the instructors as early as possible. Accommodations are not retroactive and must be requested in advance.

Grade Appeals

If you believe there was an error in recording your completion status (e.g., a problem marked incomplete when you achieved AC, or contest participation not recorded), you may request verification within one week of when the grade/completion status was recorded. To request verification:

1. Submit a written explanation of what was incorrectly recorded
2. Include specific evidence (Codeforces submission links, contest participation records)
3. Send request to the instructors via email

Note: Grade appeals are for verification of recorded completion status. Subjective evaluation of work quality is not subject to appeal.

Discord

We encourage you to join the following Discord communities for additional support and discussion:

- [ICPC Greater New York Region \(GNYR\)](#)
- [Problem Solving Warriors](#)
- [ICPC Training Warriors](#)
- [Quant Warriors](#)
- [Programming Zealots](#)

Frequently Asked Questions (FAQ)

Requirements and Grading

- **Q: What happens if I solve 145 problems instead of 150? A:** You will receive proportional credit (14.5% out of 15%) for the Assignments component. Full credit requires 150 problems.
- **Q: Can I still pass if I complete only 8 contests? A:** Yes, you will receive proportional credit (56% out of 70%) for the Contests component. However, meeting the 10-contest minimum is strongly recommended for full credit.
- **Q: How is my progress tracked? A:** See the “Course Website” section for detailed progress tracking instructions.
- **Q: What counts as a “satisfactory” presentation? A:** A presentation is satisfactory if it meets all criteria: correct solution (AC verdict), clear problem explanation, clear solution approach, complexity analysis, and ability to answer questions.
- **Q: Can grading policies change during the semester? A:** No. The rubric in this syllabus is fixed as of day one and will not change after the first week of classes, except in truly unavoidable circumstances — and any such change will only ever be in students’ favor. See “Our Commitments to You.”

Contests and Participation

- **Q: Can I use virtual participation to count toward the contest requirement? A:** No, only live participation during scheduled contest time counts. See the “Live Contests” section for details.
- **Q: What if I miss a contest due to an emergency? A:** Contact the instructors as soon as possible with documentation. Extenuating circumstances may be excused on a case-by-case basis.
- **Q: Can AtCoder contests substitute for Codeforces contests? A:** Yes, see the “Live Contests” section for details on contest substitution rules.

Assignments and Problems

- **Q: Can I solve problems from any source, or must they be from assigned lists? A:** Problems must be from the assigned problem lists provided each week on CourseWorks.
- **Q: What if I solve a problem but don’t get AC on my first submission? A:** Only problems with AC (Accepted) verdict count. You can resubmit until you achieve AC.
- **Q: How do CULC problems count toward the 150-problem requirement? A:** See the “ICPC Columbia University Local Contest (CULC)” section for details.

General

- **Q: Do I need to have taken the introductory Competitive Programming course (COMS W4995)? A:** It is strongly recommended but not strictly required. If you have equivalent experience (see “Prerequisites”), contact the instructors before enrolling.
- **Q: What if I have a conflict with the CULC date? A:** Contact the instructors immediately. The CULC

is a mandatory requirement, but accommodations may be possible for documented extenuating circumstances.

- **Q: Can I work with classmates on problems?** **A:** Yes, collaboration is encouraged, but you must cite all sources and collaborators. You must understand and be able to explain any solution you submit.
- **Q: Is it normal to be stuck on problems for a long time?** **A:** Yes — completely normal, especially in an advanced course. Most students will not solve every problem on every practice set; that is by design. If you have been stuck for an hour of honest effort, that is a good time to look at hints, discuss with classmates (with citation), or come to office hours. Struggling productively is how competitive programmers improve.
- **Q: How do I get help if I'm struggling?** **A:** Schedule office hours with the instructors or TA, join Discord communities, form study groups, and refer to recommended textbooks and resources.

Practicum Course (Separate Enrollment — Optional)

Note: The practicum course is **completely optional** and **separate** from this course. It requires separate enrollment and has its own grading. Participation in the practicum course does not affect your grade in this course.

- Vergil link: TBD
- **First Session:** TBD (early in the semester)
- **Location:** TBD (typically Computer Lab, 252 Engineering Terrace)

Sundays include weekly practice sessions as follows:

- **Mock Contest:** 1:00 PM–6:00 PM ET
- **Discussion:** 6:00 PM–7:00 PM ET

Attendance Policy (Practicum Course Only):

- Based solely on physical presence
- Must be present at roll call at session start
- Expected to remain through discussion (typically concludes before 7:00 PM ET)
- Each unit requires 4 attendances: 1 unit = 4 attendances; 2 units = 8 attendances; 3 units = 12 attendances

(Tentative) Weekly Topics

Thirteen Wednesday lectures, per the Columbia Fall 2026 academic calendar (classes begin Tuesday, September 8, 2026, and the university's last day of classes is Monday, December 14, 2026; no E6998 class on Wednesday, November 25, 2026).

Week	Date	Topics
1	September 9, 2026	Kickoff — Logistics; Expectations; Advanced Contest and Practice Strategies; ICPC Team Strategy; Review of Core Techniques
2	September 16, 2026	Advanced Dynamic Programming — Divide-and-Conquer Optimization, Knuth's Optimization, Convex Hull Trick, Slope Trick
3	September 23, 2026	Advanced Graph Algorithms — Strongly Connected Components, 2-SAT, Bridges and Articulation Points, Euler Tours
4	September 30, 2026	Network Flows and Matchings — Max-Flow/Min-Cut, Min-Cost Flow, Bipartite Matching, Applications
5	October 7, 2026	Advanced Data Structures — Persistent Structures, Segment Tree Beats, Heavy-Light Decomposition, Small-to-Large
6	October 14, 2026	Advanced String Algorithms — Suffix Arrays, Suffix Automata, Aho-Corasick, Palindromic Trees
7	October 21, 2026	Polynomials and Convolutions — FFT, NTT, Polynomial Techniques
8	October 28, 2026	Advanced Number Theory and Combinatorics — Möbius Inversion, Multiplicative Functions, Burnside's Lemma
9	November 4, 2026	Advanced Computational Geometry — Sweep Line, Convex Hull Applications, Half-Plane Intersection
10	November 11, 2026	Randomized and Bitwise Techniques — Hashing, Randomization, Bitsets, Meet-in-the-Middle
11	November 18, 2026	Game Theory and Advanced Topics — Sprague-Grundy Theory, Matroids, Offline Techniques
—	November 25, 2026	<i>No class — Academic Holiday (Thanksgiving is Thursday, November 26)</i>
12	December 2, 2026	Interactive, Constructive, and Ad Hoc Problems — Interactive Protocols, Constructive Techniques, Heuristics
13	December 9, 2026	Course Review and Closing — Semester Recap, ICPC Outlook, Where to Go Next