

Welcome to AP!

COMS 3157

Advanced Programming

Fall 2026

Course Homepage: <https://cs3157.github.io/www/2026-9/>

Teaching staff

- 16 TAs
 - Photos will be posted on CourseWorks
 - Email to all teaching staff:
cucs3157-tas@googlegroups.com
- Instructor: Jae Woo Lee
 - Email: jae@cs.columbia.edu
 - Office: 401C Schermerhorn Hall
 - Homepage: <http://www.cs.columbia.edu/~jae/>
- Office hours on the course homepage

Who am I?

- Jae Woo Lee
 - Teaching Professor in Computer Science
 - Teaching first, research second
 - Just call me Jae (pronounced 'Jay')
 - Note that this is NOT a general rule – address instructors as Professors unless told otherwise
- My background
 - Undergrad in Columbia College
 - Many years of professional experience
 - Designing and coding large-scale software systems
 - Running a start-up company
 - Came back to Columbia for Ph.D.
 - More info at <http://www.cs.columbia.edu/~jae/>

This course

- Introduction to Systems Programming
- Course objective
 - Learn how computers and programs work
 - Turn a programming student into a *programmer*
- *Follow the River and You Will Find the C*
 - Published in SIGCSE 2011 (link on my homepage)
 - Great overview of this course: what, how, and why
 - Skim it now, and read again after the course

But then, it's just another class

- Focuses on systems programming
 - Precision and attention to detail
 - Systematic approach to problem solving
- And that's just one narrow aspect of CS
 - Not a gauge for general CS potential
 - Not even a gauge for programming ability
- Please don't get stressed out about AP

Set your expectations

- AP may not be your cup of tea, and that's ok
 - Systems programming may not be your thing
 - You may have other priorities in life
- You may not do well even if you try
 - This stuff is not easy, even for those who like it
- 12 hours/week is NOMINAL 4-credit workload
 - Could be a bit lighter, or could be a lot heavier
- Think of it as 13-week workout regimen
 - Your IMPROVEMENT will be what you put in

What you can expect from me

- Honesty
 - You get no-BS answers to the best of my ability
 - *Cons*: People say I am very blunt
- Transparency
 - You will know everything – how HW & exam are graded, why I do certain things in class, etc., etc.
 - Ask anything; I'll either answer it or tell you why I cannot
 - *Cons*: None I can think of – at least to students
- Fairness
 - Hard rubric, no extensions, no tolerance on cheating
 - *Cons*: Students are denied exceptions for the sake of fairness to the whole class

What past students wrote

- Past evaluations of all my classes:
<http://www.cs.columbia.edu/~jae/evals/>
- Reviews on CULPA, etc.

TLDR:

- Some people love the course, others hate it; some people think I am great, others think I am horrible
- Give it a chuckle and focus on your own learning

Lectures & Recitations

- In-person Lectures
 - Will be recorded, but no guarantee on quality
 - Auditors are welcome to lectures & listserv
 - No Linux account, no lab/exam submissions, no TA access
- Recitations by me
 - Beginning of most Sec 2 classes on Fridays
 - Sec 2 students attend live; others watch recordings
 - Some pre-recorded sessions as well
 - Mandatory unless I say optional
- Optional Review Sessions by TAs
 - Reviews and Q&As, but NO new materials
 - Normally NOT recorded, except for exam reviews

Section 2

- COMS 3157 Sec 2: Small flipped class of ~30 students
 - Fridays 12:10-2pm (starting 9/18), 330 Uris Hall
 - Must attend Sec 1 lectures before class
 - Must be free in Sec 1 slot (TR 4:10-5:25) to take all Sec 1 exams
- What do Sec 2 students do in there?
 - After my recitation, work on problems
 - Consistent attendance is expected
 - All recordings & problems available to Sec 1 students too
- Who is this for?
 - Students who self-identify as needing extra help
 - Priority given to students who didn't do well in 1004 & 3134
 - Register for Sec 1 first; add yourself to Sec 2 waitlist; apply via email (details at the end); drop Sec 1 once you are in Sec 2

Prerequisites

- Absolutely required
 - Two semesters of solid programming
 - Must be able to write decent size programs yourself
- Pretty much required
 - COMS 3134 Data Structures or equivalent
 - For general CS & programming maturity
 - Ex) I'll assume you can write recursive functions
- No C knowledge assumed
- No Java knowledge assumed

Topics covered

- C language, top to bottom
 - The most important part of the course
 - Everything else depends on it
- Intro to UNIX Systems Programming
 - I/O, Process control, TCP/IP networking
 - Sockets API and HTTP protocol
 - Final assignment: write your own web server from scratch!

Why C?

- Your kung fu will be better
 - C programmers operate at a different level, even when they use other languages
- You will actually understand stuff
 - Understand how computers work
 - Understand how other languages work
- You need to know C cold if you:
 - Want to become a systems programmer
 - Want to become an expert C++ programmer

Grading

- **GRADING POLICY MAY CHANGE LATER**
- You get an overall score out of 100:
 - 3 Exams (27% each)
 - 7 Lab assignments & HW0 (15% total)
 - Caveat: max lab score is less than 100
 - Attendance (4%)
 - Logistics TBA but you will be allowed 10 absences
- I look at everyone's lab & exam scores in a big spreadsheet sorted by the overall score
- I decide cutoffs for letter grades A+, ..., D, F
 - No predetermined formula
 - Mean & median are usually around B/B+

Exams

- Three in-person exams
 - Exam 1 on Thu Oct 15, 4:10-5:25pm
 - Exam 2 on Tue Nov 17, 4:10-5:25pm
 - Exam 3 (final) on Tue Dec 22, 3:30-5:00pm (90 min)
- **There are no make-up or alternate exams**
 - If you cannot make any of those exams, please take the course next semester
- Extended-time exams at ODS/CARDS must overlap with official exam time by at least an hour
 - You may not be able to have a class before/after AP

7 assignments (aka “labs”)

- Lowest lab score is dropped (converted to 0)
 - $(\text{sum}(\text{your labs} + \text{hw0}) - \text{min}(\text{your labs})) / \text{sum}(\text{total labs} + \text{hw0}) * 100$
 - Labs 1-5 are worth 100, lab 6 is 120, lab 7 is 150, and HW0 is 50
 - Maximum possible lab score: $(820 - 100) / 820 * 100 = 87.8$
 - May skip grading some labs, in which case formula will change
- Deadline
 - Soft deadline, and then hard deadline 2 days later
 - You use 1 late day if you submit within 24 hours after the soft deadline
 - You use 2 late days if you submit between 24 and 48 hours
 - After 48 hours past the soft deadline, no submission will be accepted
 - You have 7 late days total; up to 2 can be used for a single lab
 - Check late days: `/home/w3157/submit/check-late-days`
 - Late days are for unforeseen circumstances such as sickness
 - Please do not ask for additional extensions

How to do well in AP

1. First and foremost, WORK
 - 4 credit course → 12 hour/week NOMINAL workload
 - That is 2 hours of AP, 6 days a week, starting now
 - Your mileage may vary, but consider it a bare minimum
2. Do the labs. I mean, *really* do the labs.
 - Don't just "get it working" – understand every detail
 - Don't code by trial & error – understand your errors
 - Don't let TAs fix your problems – it's all about the process
3. Learn to read code on paper
 - Read & understand every line of solution & exam code
 - Then try coding them yourself without looking
4. Attend lectures and pay attention

Please don't cheat

- REQUIRED READING:
<http://www.cs.columbia.edu/~jae/honesty.html>
- You are cheating if you:
 - Take code from friends, previous solutions, the Internet, or AI
 - Upload any class materials (including your own code) to public repository (ex. GitHub) during or after this semester
- We can tell
 - You submit work history – minimum 5 commits required
 - As a beginner, once you peek at cheat code, you won't be able to come up with any other way to do the same thing

Please don't use AI

- Don't become a “**pass-through engineer**”
 - Prompt AI → Take AI slop wholesale → Let others review/fix it
 - Pass-through engineers don't know how to do much else
 - Many stories from those other engineers (former AP students)
- Can I use AI for learning and not become a pass-through?
 - No, not at this early stage of learning
 - AI will rob you of **cognitive struggle**
- Can I just learn to read & understand code?
 - No, not unless you also write code yourself
- Bottom line:
 - AI is useful, so learn to use it well, but not in AP
 - Not using AI will make AP difficult – exactly what you want
 - AI will replace many humans – your choice to be one of them or not
- More on this in my interview at Mary in America:
 - <https://youtu.be/zxJnTs1lhQ4?si=LVbfsWWtN3AwMKUT>

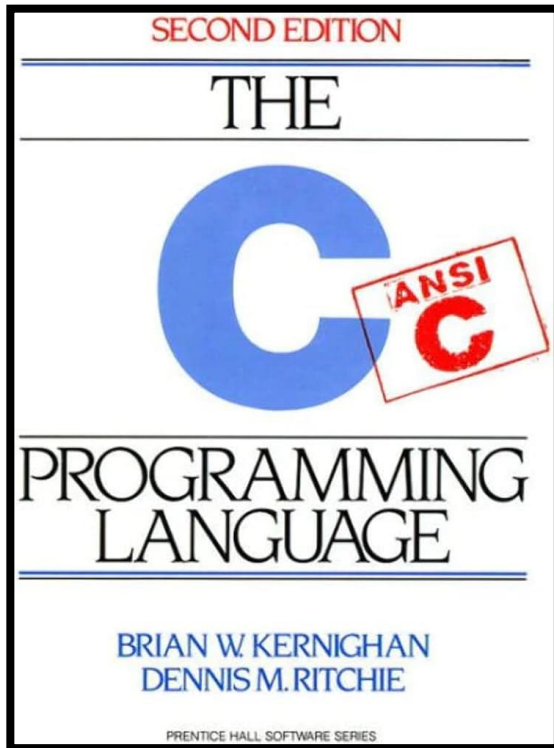
Class ListServ

- Questions, answers, and announcements
- Do:
 - Ask & answer questions
 - Provide helpful tips and fun links for your classmates
 - Be considerate & friendly
- Don't:
 - Ask questions without first trying to solve it on your own
 - Post code or critical info that leads directly to solution
 - Be impatient & rude
- Use class listserv rather than TA list if possible
 - Please help us not duplicate work
 - General questions to TAs anonymized & redirected to class
 - Never send a same question individually to multiple TAs
- Anonymous feedback form also available

Manage ListServ emails

- Manage high volume – filter by tags in subject
 - [cs3157] – all emails from class listserv will have this tag
 - [ANN] – important announcements from me or TAs
 - [LAB n] – information relevant on a particular lab
 - Examples:
 - [cs3157][ANN] Sample midterm
 - [cs3157][ANN][LAB7] Correction on lab7 instruction
 - [cs3157][LAB6] in case you're curious about fdopen()
- Setup Gmail filters
 - I will send an example soon
- Please keep up
 - At a minimum, you must read every single ANN

Textbooks



- Highly recommended
 - The C Programming Language* (2nd ed.)
 - aka K&R C
 - By Kernighan and Ritchie
 - Simply the best
- Recommended for self-studying beyond this class
 - *Advanced Programming in the UNIX Environment* (3rd ed.)
 - By Stevens & Rago

HW0: 50 points total

- **Part A (20 points): due Tuesday 9/8, 11:59pm (tonight)**

Subscribe to 3157 ListServ today

- <https://lists.cs.columbia.edu/mailman/listinfo/cs3157>
- In the textbox “Your name (optional)” put **Your Full Name (UNI)**
 - For example: Jae Woo Lee (jwl3)
- **Follow the link next to “Or visit this web page” in the confirm email (which could be in your spam folder)**
- Then receive “Welcome to the “Cs3157” mailing list”
 - This email contains your password for accessing archives of past postings
- **All emails to listserv, TAs, or me MUST include your UNI**
 - Sign it with UNI if you don’t use UNI@columbia.edu
 - Or just use UNI@Columbia.edu instead of first.last (please)

HW0 continued

- **Part B (30 points): due Thursday 9/10, 11:59pm**
 1. Read the following two documents:
 - <http://www.cs.columbia.edu/education/honesty>
 - <http://www.cs.columbia.edu/~jae/honesty.html>
 2. Send me an email (to jae@cs.columbia.edu) containing:
 - Subject: “[3157] hw0-UNI”
 - Without the quotes, sole space before hw0, UNI replaced with your actual UNI in lowercase
 - Your name, major & school program, year
 - Ex) Jae Woo Lee, Physics, Columbia College, class of 1994
 - Your pledge
 - see honesty.html above
 - CS classes taken and/or other programming background
 - Optionally anything else you want to let me know
 - Optionally attach a picture of you, but please reduce image file size (preferably to about 100KB)

How to apply for Section 2

- If you'd like to join Sec 2, follow these steps:
 1. Register for Sec 1
 2. Add yourself to Sec 2 waitlist
 3. Apply via email (see below)
 4. If you're approved and in Sec 2, drop Sec 1
- Email subject: “[3157] sec2-application-UNI”
 - Without quotes, sole space before sec2 (lowercase sec), UNI replaced with your actual UNI in lowercase
- Include the following info in the email:
 1. Confirm you are free in TR 4:10-5:25 slot
 2. Confirm you are (or will be shortly) registered in Sec 1
 3. Your grades in 1004 & 3134
 4. Why you want to be in Sec 2 (briefly)

I will start reviewing applications this weekend.