

Welcome to AP!

COMS 3157

Advanced Programming

Spring 2021

Teaching staff

- 15-20 Teaching Assistants (TAs), all former AP students
 - Full list with photos will be posted on CourseWorks
 - Emails to cucs3157-tas@googlegroups.com go to all teaching staff
 - TA office hour calendar: <http://bit.ly/3157-cal>
- Instructor: Jae Woo Lee
 - Email: jae@cs.columbia.edu
 - Office: 715 CEPSR
 - Jae's office hour calendar: <http://bit.ly/jae-cal>
 - Home page: <http://www.cs.columbia.edu/~jae/>

Who am I?

- Jae Woo Lee
 - Senior Lecturer 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
 - Right now, you are a programming student
 - After this course, you will become a *programmer*
- How?
 - Go beyond Java & IDE: learn C & command line tools
 - Learn advanced techniques & design principles
- *Follow the River and You Will Find the C*
 - Paper published in SIGCSE 2011 (link on my home page)
 - Great overview of this course: what, how, and why
 - Read it now, and 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 one narrow aspect of CS
 - Not a gauge for general CS potential
 - Not even a gauge for general 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
- It also means that you may not do well even if you try
 - This stuff is not easy, even for those who like it
- 12 hours/week is the NOMINAL workload for 4-credit course
 - Could be a bit lighter, or could be a lot heavier
- Think of this course as a 13-week workout regimen
 - Your IMPROVEMENT will be what you put in

What you can expect from me

- Honesty
 - You get straight, no-BS answers to the best of my ability
 - Cons: People say I am very blunt
- Transparency
 - You will know everything – how hw & exams are graded, why I do certain things in class, etc., etc.
 - Please ask anything. I'll either answer it, or tell you why I cannot.
 - Cons: None I can think of – at least to students
- Fairness
 - Fairness plays a big role in how I run my class
 - Ex) hard grading rubric, no extensions, no tolerance on cheating
 - Cons: Students are often denied exceptions for the sake of fairness to the whole class

What past students wrote

- Past evaluations of all my classes are posted here:
 - <http://www.cs.columbia.edu/~jae/evals/>
- CULPA.info has reviews on me

TLDR:

- Some people love the course, others hate it; some people think I am great, others think I am horrible
- Focus on your own learning

Lectures

- Mix of sync and async online instructions
- For each class day, one of the following:
 1. Live lecture
 - Synchronous participation expected and encouraged
 - But all live sessions will be recorded and uploaded to CourseWorks
 2. Pre-recorded lecture + Live lecture
 - Recorded lecture from Fall 19 may be assigned, which you must watch BEFORE the live session
 3. Pre-recorded lecture only
 - Occasionally there might be no live lecture
- TA review sessions scheduled as needed
 - Exam preps for example

Section 2

- SEC 001 TR 4:10–5:25pm
 - Regular online lectures
 - Auditors are welcome to lectures & listserv, but no Linux account, no lab/exam submissions, no TA access
- SEC 002 F 2:10–4:00pm
 - Hybrid flipped class
 - Limited to small number of students
 - Must be free in SEC 001 slot (TR 4:10–5:25pm) for midterm & final exams
 - Must attend SEC 001 lectures or watch video recordings before Friday class
 - Application details at the end of this slides

Exam dates

- Online but **synchronous** exams for both SEC 001 and SEC 002
 - Thu 2/18, 4:10pm: Midterm exam #1
 - Thu 3/25, 4:10pm: Midterm exam #2
 - Tue 4/20, 4:20pm: Final exam
- All students in both SEC 001 and SEC 002 **MUST** take the exams at those times. **There are no make-up or alternate exams.**
 - If you cannot make any of those exams, please take the course next semester.

Prerequisites

- Absolutely required
 - 2 or 3 semesters of Columbia-level programming courses
 - Ex) 1006-1004-3134
- Pretty much required
 - 3134 Data Structures
 - For general CS & programming maturity
 - Ex) I'll assume you know all about recursion
 - Taking 3134 and 3157 together is not recommended unless you have a very light load
- No C knowledge assumed
- No Java knowledge assumed

Topics covered

- C
 - Mastery of the C language is the most important part
 - 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?

- It's cool
 - There are two kinds of programmers: those who know C and those who don't
 - *Corollary*: There are two kinds of *Java* programmers: those who know C and those who don't
 - Your kung fu will be better than theirs
- It's fundamental
 - Understand how other languages work
 - Understand how computers work
- It's useful
 - C is still useful for some things
 - Knowing C, you can learn C++ the right way

Grading

- **GRADING POLICY MAY CHANGE LATER**
- You get an overall score out of 100, comprised of:
 - Lab assignments (35%)
 - Midterm exam 1 & 2 (20% each)
 - Final exam (25%)
- 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
 - Usually mean/median are around B/B+
 - No one will get F as long as they keep trying until the end
- I reserve the right to raise one's overall score by a small amount, based on things like:
 - Class & mailing list participation
 - Beautiful code & documentation

7 assignments (aka labs)

- Lab grading
 - Your lowest lab score will be converted to zero
 - Lab score = $(\text{SUM}(\text{your lab 1-7 \& HW0}) - \text{MIN}(\text{your lab 1-7})) / 820 * 100$
 - 100 for lab 1-5, 120 for lab 6, 150 for lab 7 (and 50 for HW0)
 - Maximum possible lab score is $720/820*100 = 87.8$, not 100
- 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
 - Absolutely no exception under any circumstances
 - You have 2 weeks to send re-grade request

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 that 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
 - Private tutors are not recommended
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, or search for code on the Internet
- Look at solutions your friend has from previous semester
- Upload any class materials (including your own code) to public repository (ex. GitHub) during or after this semester

- We can tell

- We compare your submissions to **CURRENT AND PREVIOUS** submissions
- 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

Class ListServ

- Communication between all of us
 - Official announcements, lecture notes, lab assignments
 - Should be the 1st place to go for non-personal questions
- 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
- Please use class listserv rather than the TA mailing list
 - The class is huge; please help us not duplicate work
 - General questions to the TAs may be redirected to class listserv with your ID removed
 - Never send a same question individually to multiple TAs
- There will be an ongoing anonymous feedback form

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
 - [LABn] – 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

- Required

- *The C Programming Language* (2nd ed.) – aka K&R C

- By Kernighan and Ritchie
 - Simply the best

- Survey in Spring 2016: only 4% bought them at the local bookstore

- So get them wherever you usually get your textbooks

- Recommended for self-studying beyond this class

- *Advanced Programming in the UNIX Environment* (3rd ed.)

- By Stevens & Rago

How to apply for SEC 002

- Send me an email by **Fri, 1/15, 11:59pm**
- Subject: “[3157] sec002-application-UNI”
 - Without the quotes, sole space before sec002 (lowercase sec), UNI replaced with your actual UNI in lowercase
- Include the following info:
 1. Confirm you are free in TR 4:10-5:25 slot
 2. Confirm you are (or will be shortly) registered in SEC 001
 - **Only way to get into SEC 002 is to request a section switch**
 3. Your grades in 1004 & 3134
 4. Why you want to be in SEC 002
 - Whether you think you need extra help & extra work, why you think flipped class might work for you, or anything else you’d like to tell me

HW0: 50 points total

- **Part A (20 points): due Tuesday 1/12, 11:59pm (tonight)**
 1. **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)
 - **You must reply to the confirm email (which might 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 or whatever... (please)
 2. **Get the textbooks**
 - Start reading K&R chapters 1,2,3,4

HW0 continued

- **Part B (30 points): due Thursday 1/14, 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 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 to about 100KB