Welcome to AP!

COMS 3157
Advanced Programming
Spring 2020
Teaching staff

• 16 Teaching Assistants (TAs), all former 3157 students
  – Gustaf Ahdritz gwa2107@columbia.edu – Head TA
  – Catherine Chu cjc2243@barnard.edu – Head TA
  – Hans Jr Montero hjm2133@columbia.edu
  – Amanda Liu al3623@columbia.edu
  – Lucie le Blanc ll3163@barnard.edu
  – Maria Kogan mk4036@columbia.edu
  – Lauren Anna Ogden lao2125@columbia.edu
  – Julia Guo jzg2110@columbia.edu
  – Benjamin Most bmm2168@columbia.edu
  – Brennan Xavier McManus bm2530@columbia.edu
  – James (Jae Hyun) Yang jy2816@columbia.edu
  – Kent John Hall kjh2166@columbia.edu
  – Michael Jan mj2886@columbia.edu
  – Bill (Gen Yuan) Chen gc2677@columbia.edu
  – Stanley Ye yy2922@columbia.edu
  – Spencer Bruce sgb2145@columbia.edu
Teaching staff contact info

• TA email & office hours
  – Email to cucs3157-tas@googlegroups.com goes to all teaching staff
  – TA room – 1st floor, Mudd building
  – TA calendar: http://bit.ly/3157-cal (will be filled by this weekend)

• Instructor email & office hours
  – Jae Woo Lee jae@cs.columbia.edu – 715 CEPSR
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.
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/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 after all, it’s just another class

- Focuses on skills for 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 this class
Exam dates

- T 2/25, 4:10–5:25pm: Midterm exam #1
- T 4/7, 4:10–5:25pm: Midterm exam #2
- T 5/12, 4:30–6:30pm: Final exam
- There are no make-up and no alternate exams
  - Please take AP next semester if the exam dates don’t work for you
Registration and forms

• Auditors are welcome to lectures
  – But no Canvas; no Linux account; no homework; no exams; no TA access; no review sessions

• SPS students must contact SEAS Dean’s office
  – Registrar told me never to sign add-drop form

• All forms and other paperwork in my office hours
  – Please don’t bring paperwork after class
Review sessions

• Logistics
  – One topic / week, multiple sessions by different TAs
  – Most likely evenings between Friday and Monday
    • Time and place TBA
  – Attendance optional, but recommended

• Topics
  – UNIX basics, editors, Git, etc. (in the beginning)
  – Lecture reviews
  – Lab assignment clarifications & reviews
  – Lab solutions walk-through
  – Exam preps
  – Exam solution walk-through
Prerequisites

• Absolutely required
  – 2 or 3 semesters of Columbia-level programming courses
    • Ex) 1006-1004-3134; 1007-3137; etc.

• Pretty much required
  – Data Structures (3134 or 3137)
    • For general CS & programming maturity
    • Ex) I’ll assume you know all about recursion
    • Taking DS and 3157 together is not recommended unless you have a very light load

• Recommended
  – Familiarity with UNIX environment – if not, learn ASAP
  – Knowledge of Java – only to draw comparisons with C++

• No C/C++ knowledge assumed
Topics covered

Course is divided into 3 parts:

1) C
   - Mastery of C language is the most important part
   - Everything else depends on it!

2) UNIX systems programming
   - Process control, signal, I/O, TCP/IP networking
   - Sockets API and HTTP protocol
     - Write your own web server from scratch!

3) C++
   - C++ language: we will not cover everything
   - Generic programming: templates and STL
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
  – Learn C++ the right way by learning C first
COMS 3136 for non-CS majors

• COMS W3136 Essential Data Structures in C/C++
  – Please consider 3136 if you’re not a CS major
  – Will be offered in Fall 2020

• A fusion of 3157 and 3134
  – 3157-lite: C & C++, but no heavy systems stuff
  – 3134-extract: only the most important data structures
  – Bridges E1006 and many 4000-level CS courses
  – Perfect for EE & IEOR folks who came to 3157 to learn C/C++ but found it a bit too much
Grading

- Grading logistics may change later
- You get overall score out of 100, comprised of:
  - Midterm #1 (15%), Midterm #2 (25%), Final (30%), Lab assignments (30%)
- 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+
- 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
10 assignments (aka labs)

- Lab grading
  - Your lowest lab score will be converted to zero
    - Lab grade = \((\text{Sum\_of\_your\_lab\_scores} - \text{Min\_of\_your\_labs}) / 1020 \times 100\)
    - 1020: 100 for lab 1-5 & 9, 120 for lab6, 150 for lab 7 & 10
  - Lab 8 is optional and not graded
  - Additional labs may not be graded
    - All labs (except 8) will be graded unless I say otherwise after the deadline

- 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 the soft deadline
    - 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 your late days by running: /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
  - After you receive grade, you have 2 weeks to send re-grade request
How to do well in AP

1. First and foremost, WORK
   – 4 credit course → 13-14 hours of work / week on average
   – That is 2 hours of AP every single day, starting TODAY
   – 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 code & sample exams
   – Then try coding them yourself without looking

4. Attend lectures and pay attention
Zero tolerance on cheating

• **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 that 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 you submissions to CURRENT AND PREVIOUS submissions
  – You submit work history – minimum 5 commits required
  – Once you look at cheat code, you won’t be able to come up with anything else

• Result of cheating
  – Academic penalty – anywhere between 1 letter grade down and F
  – Referral to the Office of Judicial Affairs
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

• TAs and I respond to emails in this order:
  1. All pending questions on the listserv first
  2. All pending questions sent to cuck3157-tas@googlegroups.com
  3. Then individual emails
  4. NEVER send a same question separately to multiple people
     • You will get banned from ever sending an email if you get caught doing this.
Manage ListServ emails

• Learn to manage high volume – filter by tags in subject
  – [cs3157] – all emails from the 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
• Keep up diligently

• Yes, I know about Piazza. Thanks for your suggestion.
Textbooks

• Required
     • By Kernighan and Ritchie
     • Simply the best
  2. *A Tour of C++*
     • By Bjarne Stroustrup
     – 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
HW0: 50 points total

- **Part A (20 points):** due Tuesday 1/21, 11:59pm (tonight)
  1. Subscribe to 3157 ListServ today
     - 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
  2. Get the textbooks
     - Start reading K&R chapters 1,2,3,4
HW0 continued

• Part B (30 points): due Thursday 1/23, 11:59pm
  1. Read the following two documents:
     • [link](http://www.cs.columbia.edu/education/honesty)
     • [link](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