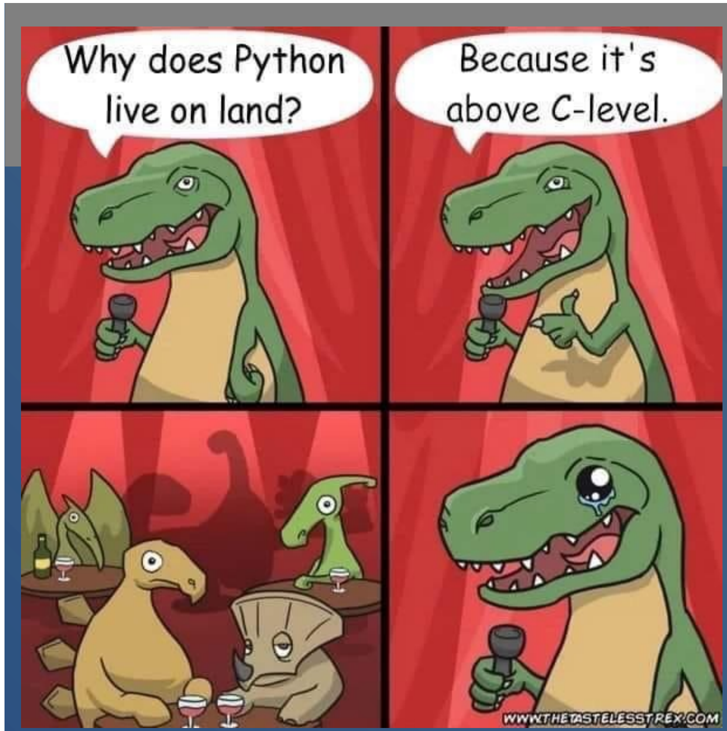
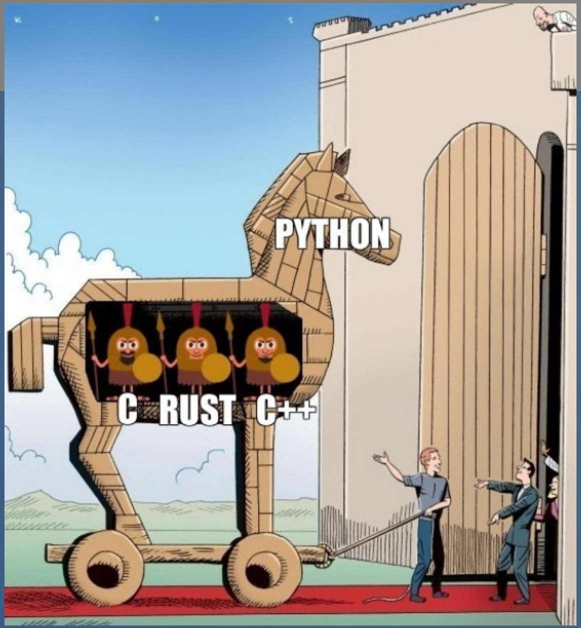
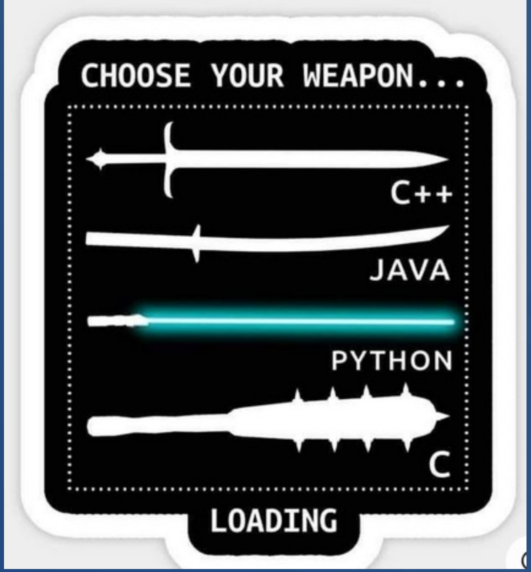
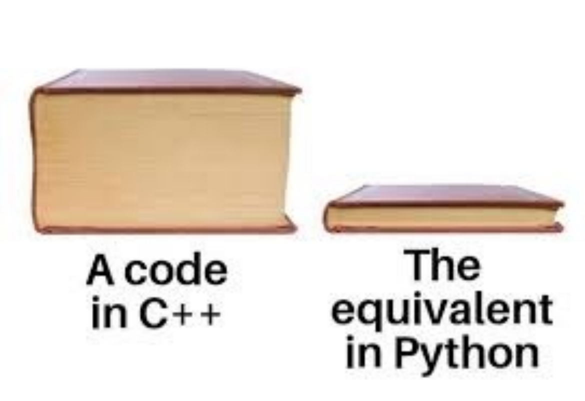








Welcome to CMPT 201 System Programming

Instructor: Dr. Linyi LI

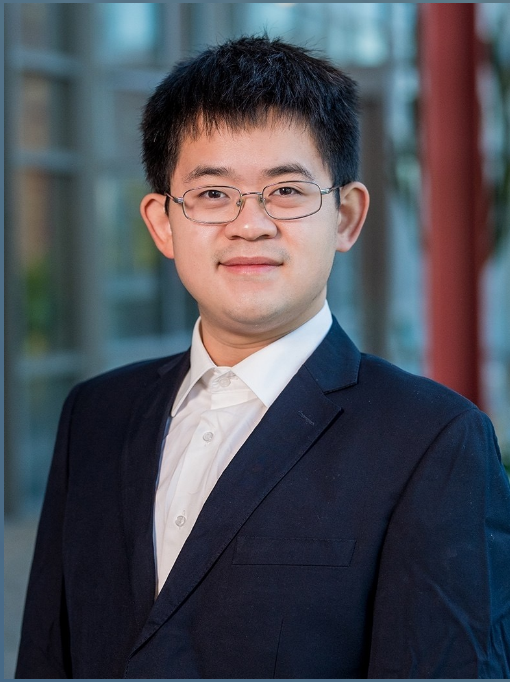
Slides adapted from Dr. Brian version (Spring 2025)

Topics

1) Introductions
2) What is System Programming?
3) Course overview
4) Demo of coding environment

About Me

- *Dr.* Linyi Li: Can just directly call me Linyi
- Degrees:
  - BSc: Tsinghua University (China), 2018
  - PhD: University of Illinois Urbana-Champaign (US), 2023
- Joined SFU (& moved to here) in August, 2024
  - New here, learn together!
- My Research: AI
  - Large Language Models (LLMs)
  - AI Trustworthiness, Safety, and Reliability
  - *Welcome to chat more if you're interested in them!*
- Office Hours:
  - Tues, Thurs 12:30pm – 1:30pm (1hr after class)
  - In person @ TASC1 9215



My webpage:
https://cs.sfu.ca/~linyi
My lab:
https://sfu-tai.github.io/

About TAs

We have 3 TAs who are ready to help you

Yuefeng Huang

- yha278@sfu.ca
- Office Hours (rick.yh on Discord): 10:30am – 12:30pm Wed, Fri
- Lab Session: 2:30 – 3:20pm Thurs @ ASB 9838 (except first & last week)

Lingjie Li

- lingjiel@sfu.ca
- Office Hours (larry_li. on Discord): 4:30pm – 5:30pm (Wed, Fri)
- Lab Session: 3:30 – 4:20pm Thurs @ ASB 9838 (except first & last week)

Tingyu Zhang

- tza80@sfu.ca
- Office Hours (_makkapakka on Discord): 1:30pm – 3:30pm (Mon), 2:30 – 4:30pm (Wed)
- Lab Session: 4:30 – 5:20pm Thurs @ ASB 9838 (except first & last week)

Stay Updated

- Course Website:
  - https://opencoursehub.cs.sfu.ca/cmpt201/grav/home/d1-linyi
  - Link also available at CourSys and my lab's page
  - Info; setup tutorial; deadlines; slides; assignment links; resources; lecture recordings; …
- Discord Channel:
  - Invite link: https://discord.gg/5KmGckY6
  - Online office hours; casual discussion
- Piazza:
  - Sign-up link: https://piazza.com/sfu.ca/summer2025/cmpt201
  - More formal questions; re-grading requests

Register and join now

Stay Updated (Cont.d)

- GitHub
  - Code management platform
  - Use for assignment submission
  - Sign up here: https://github.com/
    - Suggestion: use school email for educational benefits
  - Then submit your GitHub username on CourSys

  Register and submit now

- CourSys: score release; lab submission
- Emails: more personal staffs; we target for reply within 48 hours
  - Linyi: linyi@sfu.ca
  - Yuefeng, Lingjie, Tingyu: {yha278, lingjiel, tza80}@sfu.ca

Course Expectation

- Only one thing

  Respect

  - Use a positive tone for all communication (asking questions, on Piazza forums, with TAs)
  - Anon trolling hurts and won’t be tolerated
  - Students have wide range of backgrounds; respect it

- If sending a message
  - Give a little context (class, your name, topic, ...)
  - Email: If you are sending more than 2 per week on average, over multiple weeks, it may be too many.

Guide to Slides

- Slide Colour Guide (often...):
  - Green: headings.
  - Yellow: Highlighted text.
    - This course has a midterm and final
  - Blue: Term being defined.
    - Hour: 60 minutes.

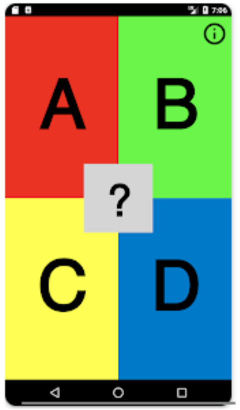
- Joke:
  - There are 10 types of people in the world...

    Those who understand binary, and
    those who get invited to parties.

Interactive Lectures

- Lectures will have
  - Lots of coding!
  - Lots of questions!
  - Lots of interactions!
- Install the ABCD app now for interactions!

Android



iPhone

What is Systems Programming?

Systems Programming

- Systems Programming
  - Low-level programming that directly interacts with hardware or the OS (Operating System)
- Languages Used
  - Need ability to manage raw memory access and other low-level tasks.
  - E.g.: C, C++, Rust
  - Python and Java don't allow you to do that.

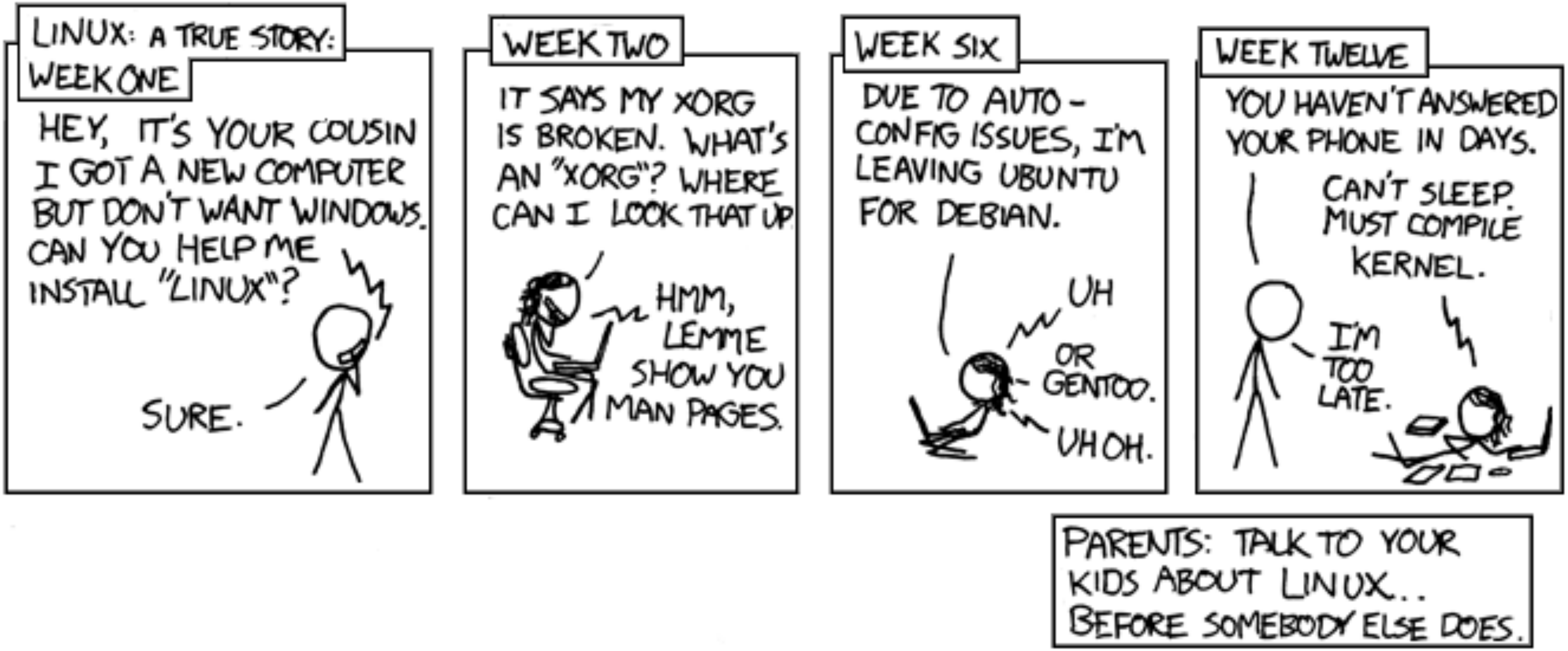
Discussion

In groups of 3 to 4 people:

- Exchange contact info (email / Discord / ...)
- Answer the following:

1. What are 3 things that would be systems programming? Which is most interesting?
2. What is one important attribute of doing systems program?
3. What do you think would be hardest about systems programming?

Course Overview


LINUX: A TRUE STORY:
WEEK ONE
HEY, IT'S YOUR COUSIN I GOT A NEW COMPUTER BUT DON'T WANT WINDOWS. CAN YOU HELP ME INSTALL "LINUX"?
SURE.
WEEK TWO
IT SAYS MY XORG IS BROKEN. WHAT'S AN "XORG"? WHERE CAN I LOOK THAT UP
HMM, LEMME SHOW YOU MAN PAGES.
WEEK SIX
DUE TO AUTO-CONFIG ISSUES, I'M LEAVING UBUNTU FOR DEBIAN.
UH
OR GENTOO.
UH OH.
WEEK TWELVE
YOU HAVEN'T ANSWERED YOUR PHONE IN DAYS.
CAN'T SLEEP. MUST COMPILE KERNEL.
I'M TOO LATE.
PARENTS: TALK TO YOUR KIDS ABOUT LINUX.. BEFORE SOMEBODY ELSE DOES.

Course Overview

- Goal
  - Be a confident developer with low-level OS services.
  - Course is very applied
    - *May* spend hours solving build issues, and debugging complex behaviour.
- Course Components

Understand user-level services of the OS

Write low-level programs using OS services

Correct

Efficient

Reliable

ABCD: Perspective

- How would you rate your programming ability?

- How good would you like to be at programming by the end of the course?

a) Excellent!
b) Good
c) OK
d) Poor

- How much work are you willing to put in?
  - Do you have the time?
  - Do you have the commitment?
  - CMPT 201 is a 4-credit course.

a) A whole lot
b) Average
c) Minimal
d) Who knows :)

What to expect

- Course is polarizing

Really love it!

The assignments really helped solidify my understanding of concepts

..teaching is very engaging and informative

(In long assignments,) you're given a spec, some general pointers and guidelines, and then basically told figure it out. (These) were, hard, yes, but super gratifying to do.

Really hate it!

This course feels like 6 credits worth.

The homework is completely unreasonable.

...assignments are just soo much tougher going beyond the lecture material

[Assignments] are a full time job that you cannot complete. They can take up to 40 hours easily then if you do not figure out one seg fault you just spent 40 hours in your week to get a 0 on a 10% assignment.

Topics

- Linux command-line interface (CLI), shell scripting, and basic development tools
- Processes and threads
- Memory management
- Virtual memory
- Scheduling
- Synchronization
- Storage and file system abstractions
- Communication abstractions such as IPC, sockets, and RPC
- Basics of OS security and cryptographic functions.

What you know

- From prereqs you know
  - Solid programming skills to write functions, loops, if, arrays, pointer, input/output.
  - Solid debugging skills to recognize defects and methodically track down errors.
  - Solid understanding of C or C++ programming.
- Passing prereq does not make this an easy course
  - Lower grades in prereq?

    or less familiar with above skills?

    Then expect to spend a lot of time becoming excellent this semester.
  - CMPT201 does not teach C; and uses Linux terminal

Opinionated

- Course is opinionated about good coding.
  - Focuses on Linux command-line interface (CLI)
  - Uses C for low-level programming (we don’t teach C).
  - These are frequently missing skill in new grads, so we’ll learn it well here!
- You may not love this approach to coding, but it will expand your skills!

Book

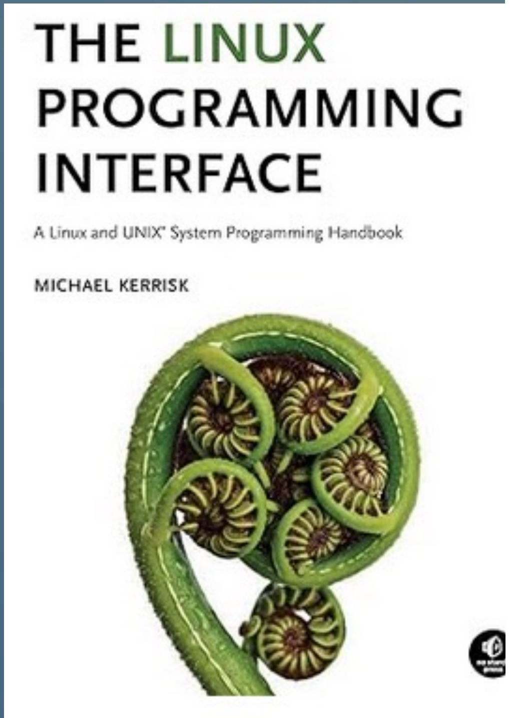
- Highly-recommended Book
  The Linux Programming Interface:
  A Linux and UNIX System Programming Handbook,
  Michael Kerrisk, 2010
- Why Recommended?

–Arguably the best book on systems programming using Linux.

–Course draws on it

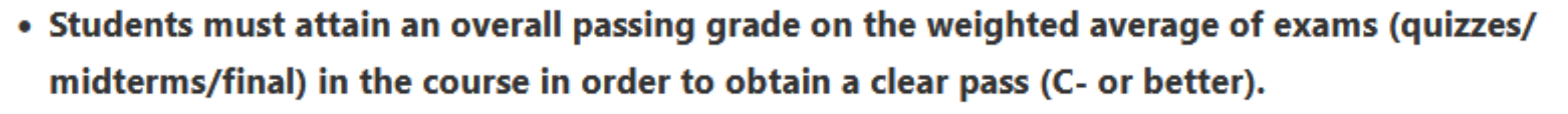
–Welcome to come in & read during my office hour

–If struggling, then very highly recommended to read along during semester!


THE LINUX PROGRAMMING INTERFACE
A Linux and UNIX System Programming Handbook
MICHAEL KERRISK

Admin Review

- Assessment
  - Midterm: 15% (Week 6 Thurs, closed-book, CAL-compatible)
  - Final: 15% (exam period, closed-book, CAL-compatible)
  - Programming assignments: 66% (8 short 2% each, 5 long 10% each)
  - Labs: 4% (equal weights for each lab)
  - Bonus: as bug bounty & reward for great feedback
  - Grade cut-offs (“% for B+?”) may be non-standard



- **Students must attain an overall passing grade on the weighted average of exams (quizzes/ midterms/final) in the course in order to obtain a clear pass (C- or better).**

- Academic Honesty
  - I am *passionate* about proving who did their own work.
  - Corollaries:
    - I'll give you credit for the work you do.
    - I'll catch those who don't do their own work.

Assignments & Labs Schedule

Usual Deadline: Sunday 11:59 PM

- Assignments

  Links on course website

- Week 1 due: A0
- Week 2 due: A1, A2
- Week 3 due: A3, A4
- Week 4 due: A5, A6
- Week 5 due: A7
- Week 6 due: A8 (long)
- Week 8 due: A9 (long)
- Week 10 due: A10 (long)
- Week 12 due: A11 (long)
- Week 14 due: A12 (long)

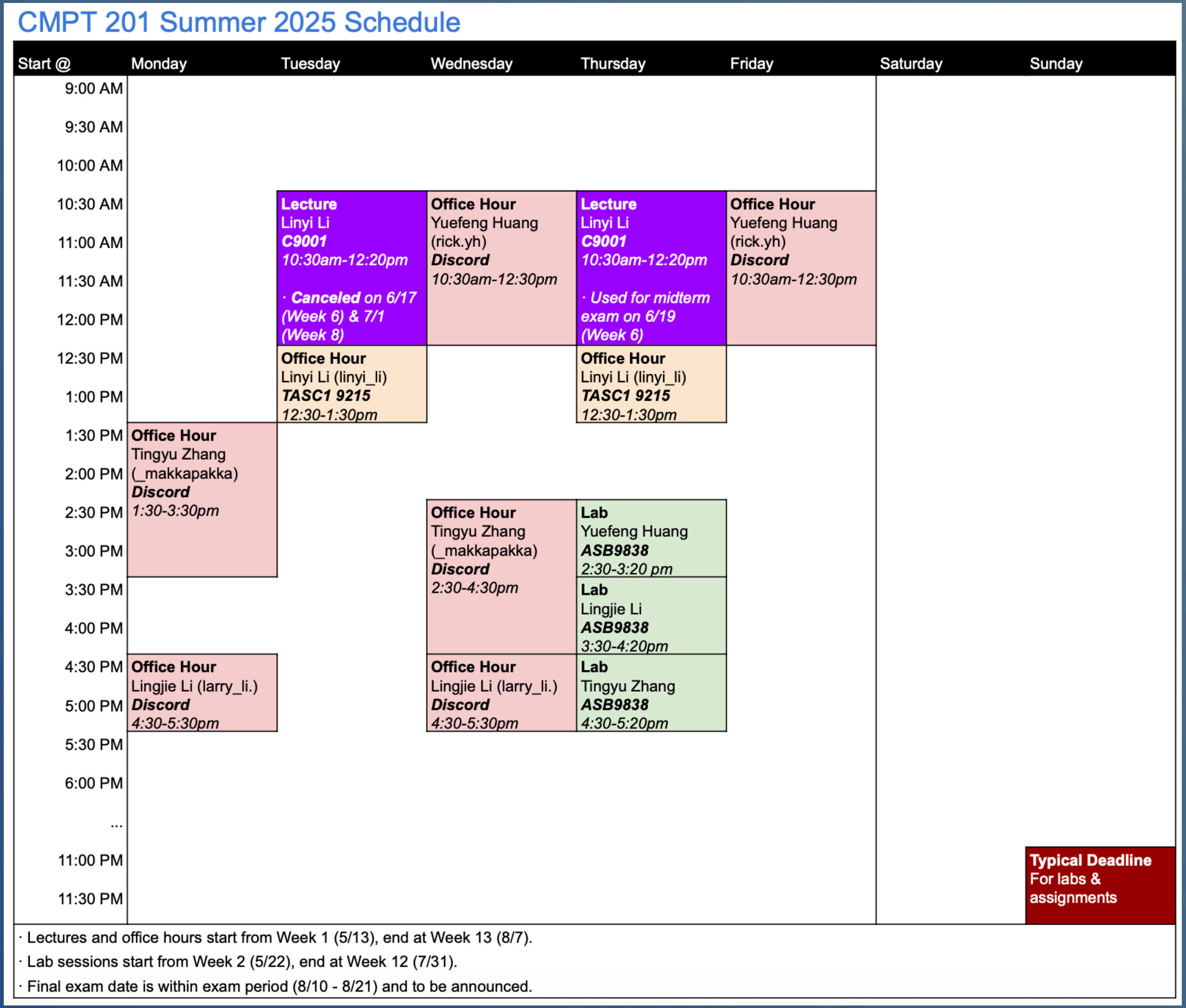
- Labs

  Submit through CourSys
  Encourage to finish during lab session
  Labs 1 – 11, occurs in Week 2-12

Main features:

- Keyboard recordings needed
- Copy-and-paste operations leads to at least 10.01% penalty

Overall Schedule



CMPT 201 Summer 2025 Schedule

Start @	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
9:00 AM							
9:30 AM							
10:00 AM							
10:30 AM		**Lecture** Linyi Li ***C9001*** *10:30am-12:20pm* · ***Canceled*** *on 6/17 (Week 6) & 7/1 (Week 8)*	**Office Hour** Yuefeng Huang (rick.yh) ***Discord*** *10:30am-12:30pm*	**Lecture** Linyi Li ***C9001*** *10:30am-12:20pm* · *Used for midterm exam on 6/19 (Week 6)*	**Office Hour** Yuefeng Huang (rick.yh) ***Discord*** *10:30am-12:30pm*		
11:00 AM							
11:30 AM							
12:00 PM							
12:30 PM		**Office Hour** Linyi Li (linyi_li) ***TASC1 9215*** *12:30-1:30pm*		**Office Hour** Linyi Li (linyi_li) ***TASC1 9215*** *12:30-1:30pm*			
1:00 PM							
1:30 PM	**Office Hour** Tingyu Zhang (_makkapakka) ***Discord*** *1:30-3:30pm*						
2:00 PM							
2:30 PM			**Office Hour** Tingyu Zhang (_makkapakka) ***Discord*** *2:30-4:30pm*	**Lab** Yuefeng Huang ***ASB9838*** *2:30-3:20 pm*			
3:00 PM							
3:30 PM				**Lab** Lingjie Li ***ASB9838*** *3:30-4:20pm*			
4:00 PM							
4:30 PM	**Office Hour** Lingjie Li (larry_li.) ***Discord*** *4:30-5:30pm*		**Office Hour** Lingjie Li (larry_li.) ***Discord*** *4:30-5:30pm*	**Lab** Tingyu Zhang ***ASB9838*** *4:30-5:20pm*			
5:00 PM							
5:30 PM							
6:00 PM							
...							
11:00 PM							**Typical Deadline** For labs & assignments
11:30 PM							

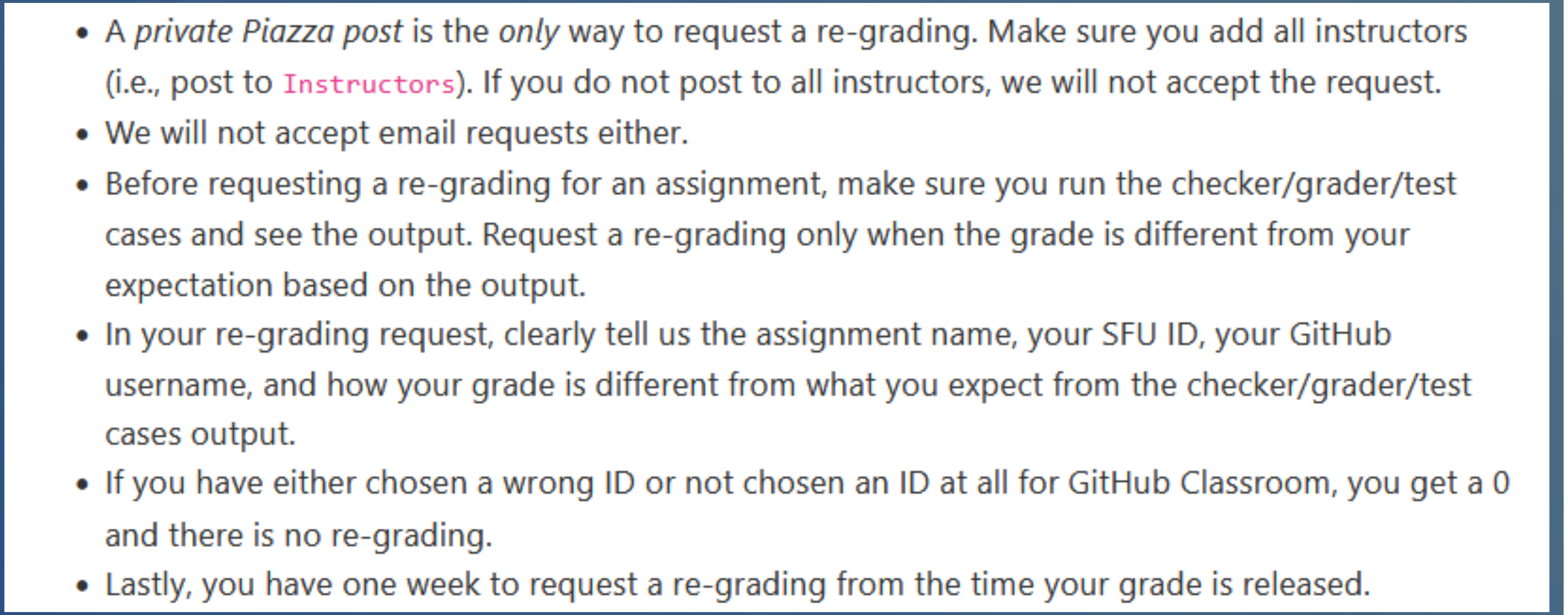
· Lectures and office hours start from Week 1 (5/13), end at Week 13 (8/7).

· Lab sessions start from Week 2 (5/22), end at Week 12 (7/31).

· Final exam date is within exam period (8/10 - 8/21) and to be announced.

Policies

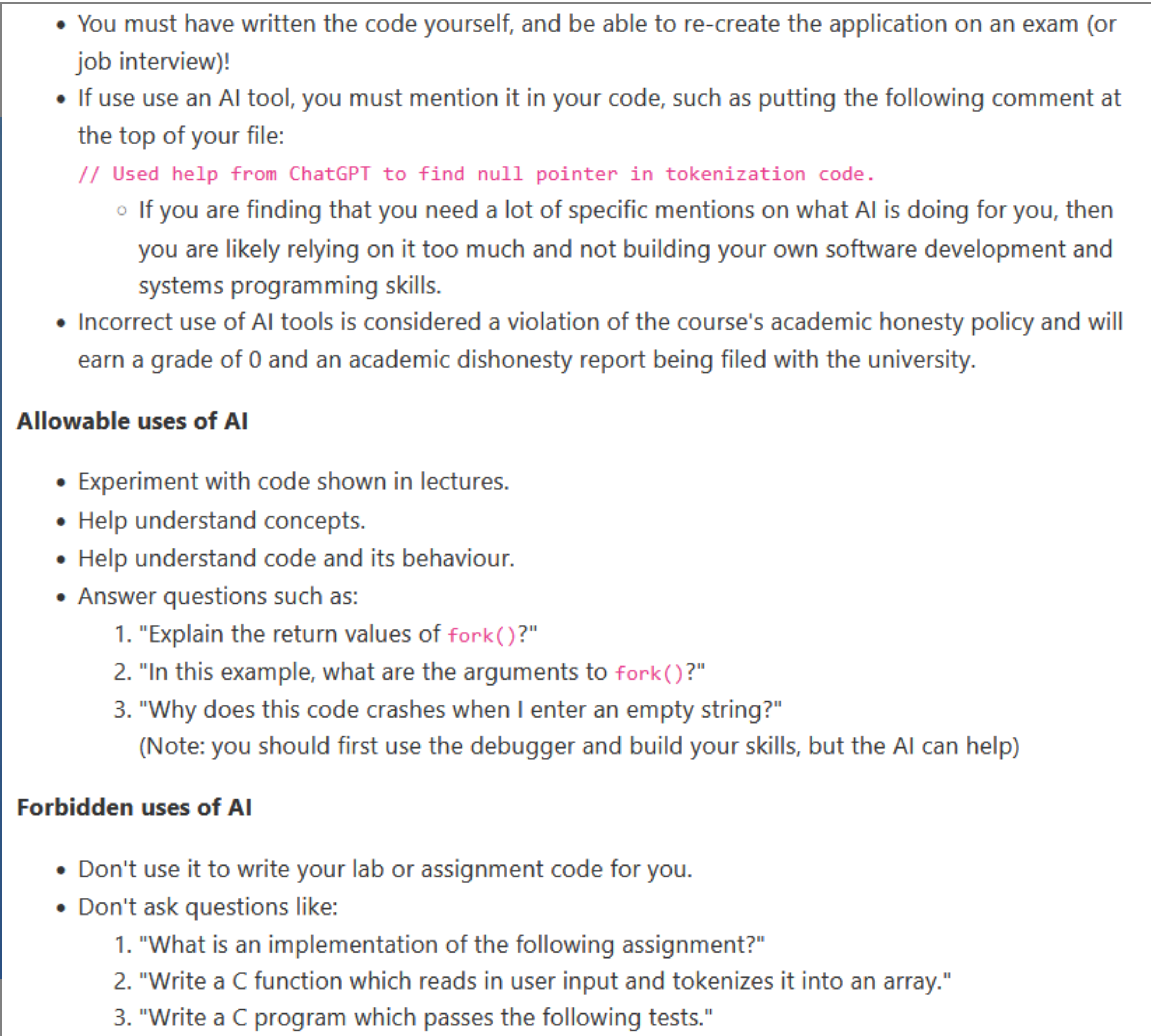
- Late
  - For labs, no late
  - For assignments,
    - Have a total of 15 days to use across the semester
    - Each assignment may be submitted at most 3 days late
    - Use the last submission timestamp to count
    - Within late budget, 0% penalty; otherwise; 100% penalty
- Regrade



- A *private Piazza post* is the *only* way to request a re-grading. Make sure you add all instructors (i.e., post to `Instructors`). If you do not post to all instructors, we will not accept the request.
- We will not accept email requests either.
- Before requesting a re-grading for an assignment, make sure you run the checker/grader/test cases and see the output. Request a re-grading only when the grade is different from your expectation based on the output.
- In your re-grading request, clearly tell us the assignment name, your SFU ID, your GitHub username, and how your grade is different from what you expect from the checker/grader/test cases output.
- If you have either chosen a wrong ID or not chosen an ID at all for GitHub Classroom, you get a 0 and there is no re-grading.
- Lastly, you have one week to request a re-grading from the time your grade is released.

Policies

- AI Tools



- You must have written the code yourself, and be able to re-create the application on an exam (or job interview)!
- If use use an AI tool, you must mention it in your code, such as putting the following comment at the top of your file:

```
// Used help from ChatGPT to find null pointer in tokenization code.
```

  - If you are finding that you need a lot of specific mentions on what AI is doing for you, then you are likely relying on it too much and not building your own software development and systems programming skills.
- Incorrect use of AI tools is considered a violation of the course's academic honesty policy and will earn a grade of 0 and an academic dishonesty report being filed with the university.

Allowable uses of AI

- Experiment with code shown in lectures.
- Help understand concepts.
- Help understand code and its behaviour.
- Answer questions such as:
  1. "Explain the return values of `fork()`?"
  2. "In this example, what are the arguments to `fork()`?"
  3. "Why does this code crashes when I enter an empty string?"
     (Note: you should first use the debugger and build your skills, but the AI can help)

Forbidden uses of AI

- Don't use it to write your lab or assignment code for you.
- Don't ask questions like:
  1. "What is an implementation of the following assignment?"
  2. "Write a C function which reads in user input and tokenizes it into an array."
  3. "Write a C program which passes the following tests."

Demo

- VM for doing Assignments
  (Like take-home tests)
- Follow the setup tutorial in our course website

  https://opencoursehub.cs.sfu.ca/cmpt201/grav/home/d1-linyi

  - Install Docker
  - Launch the VM
  - ./start_here.sh
  - Setup GitHub
  - Explain record and copy-and-paste policy
  - Neovim snapshots
  - Submit

Summary

- Course is hands on
  - Expect to learn systems programming skills
  - Expect to spend quite a bit of time figuring things out
  - Decide now if you are in for learning.
- Assignments
  - 2 short per week for first 4 weeks
  - 1 long per ~2 weeks
- Stay Updated
  - Course website, Piazza, Discord, GitHub, CourSys
  - Lectures are recorded
- Setup through the tutorial in course website