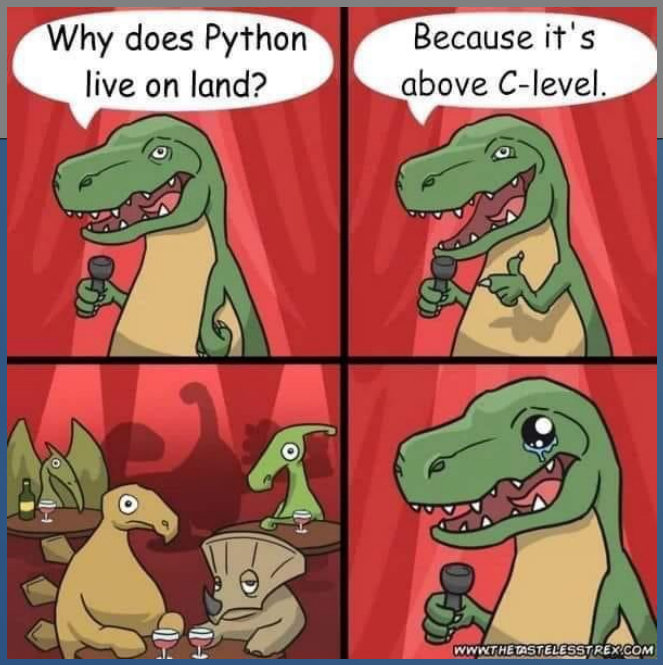
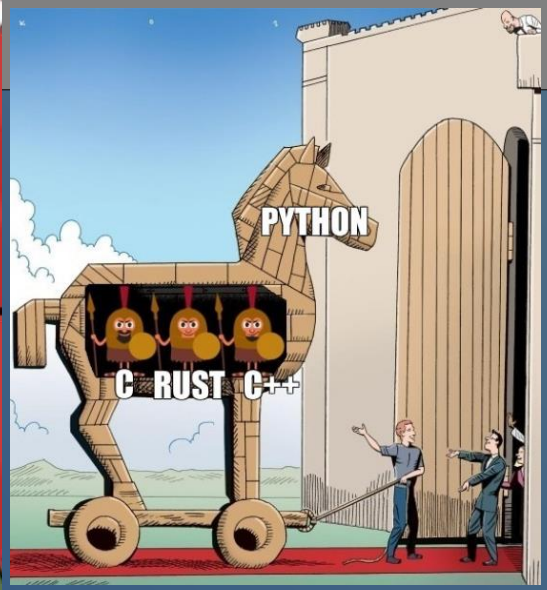
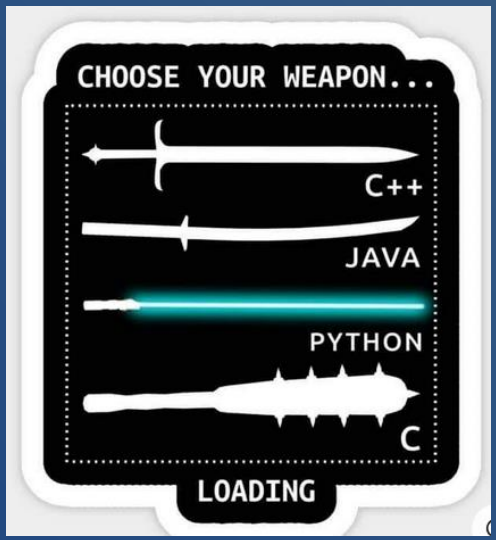
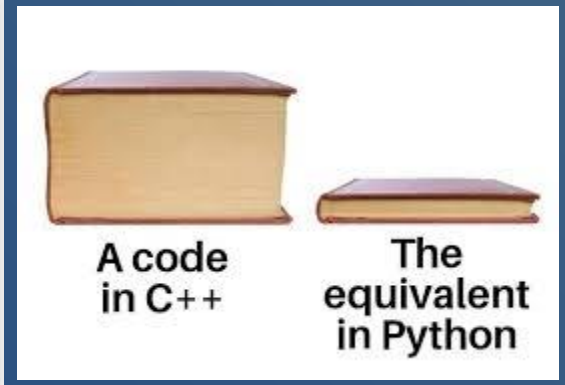








Welcome to CMPT 201 System Programming

Instructor: Dr. Linyi LI

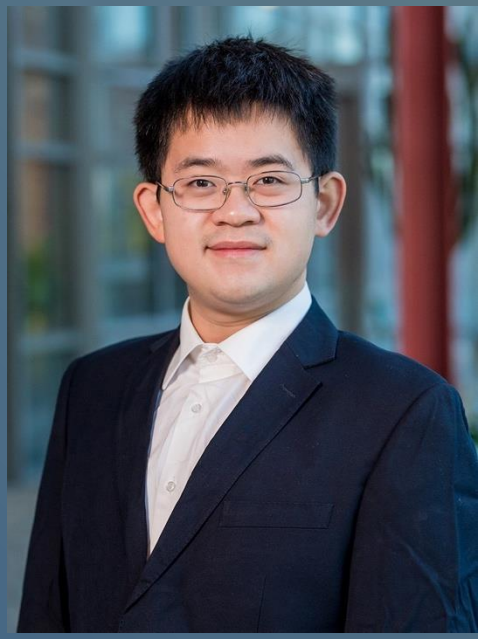
Slides adapted from Dr. Brian version (Spring 2025)

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 4:30pm – 6:20pm (1hr after class)
  - In Classroom (AQ 3182) or Office (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

Mohammadreza Doroodian (MRD)

- mohammadreza_doroodian@sfu.ca
- Office Hours (on Discord): TBD
- Lab Session: 8:30 – 9:20am Fri @ ASB 9700 (except first & last week)

Mohammadhossein Zarei (MH)

- mza148@sfu.ca
- Office Hours (on Discord): 2:00 – 6:00 pm (Fri)
- Lab Session: 9:30 – 10:20am Fri @ ASB 9700 (except first & last week)

Mohammad Reza Sadeghian (MRS)

- mr_sadeghian@sfu.ca
- Office Hours (on Discord): TBD
- Lab Session: 10:30 – 11:20am Fri @ ASB 9700 (except first & last week)

Stay Updated

- Course Website:
  - https://opencoursehub.cs.sfu.ca/cmpt201/grav/
  - Info; setup tutorial; deadlines; slides; assignment links; resources; lecture recordings; …
  - **Under construction, will be fully ready in this week**
- Discord Channel:
  - Invite link: https://discord.gg/cPBEPCHF9b
  - Online office hours; casual discussion
- Piazza:
  - Sign-up link: https://piazza.com/sfu.ca/spring2026/cmpt201linyi
  - More formal questions; re-grading requests

- Normally we don't use Canvas
- Sometimes we use massmails for important information

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 72 hours
  - Linyi: linyi@sfu.ca
  - MRD, MH, MRS: {mohammadreza_doroodian, mza148, mr_sadeghian}@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.

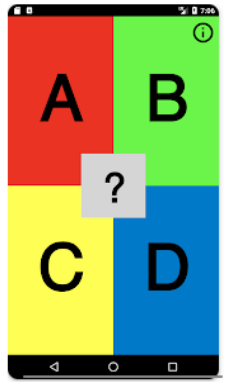
- Joke:
  - There are 10 types of people in the world...

    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:

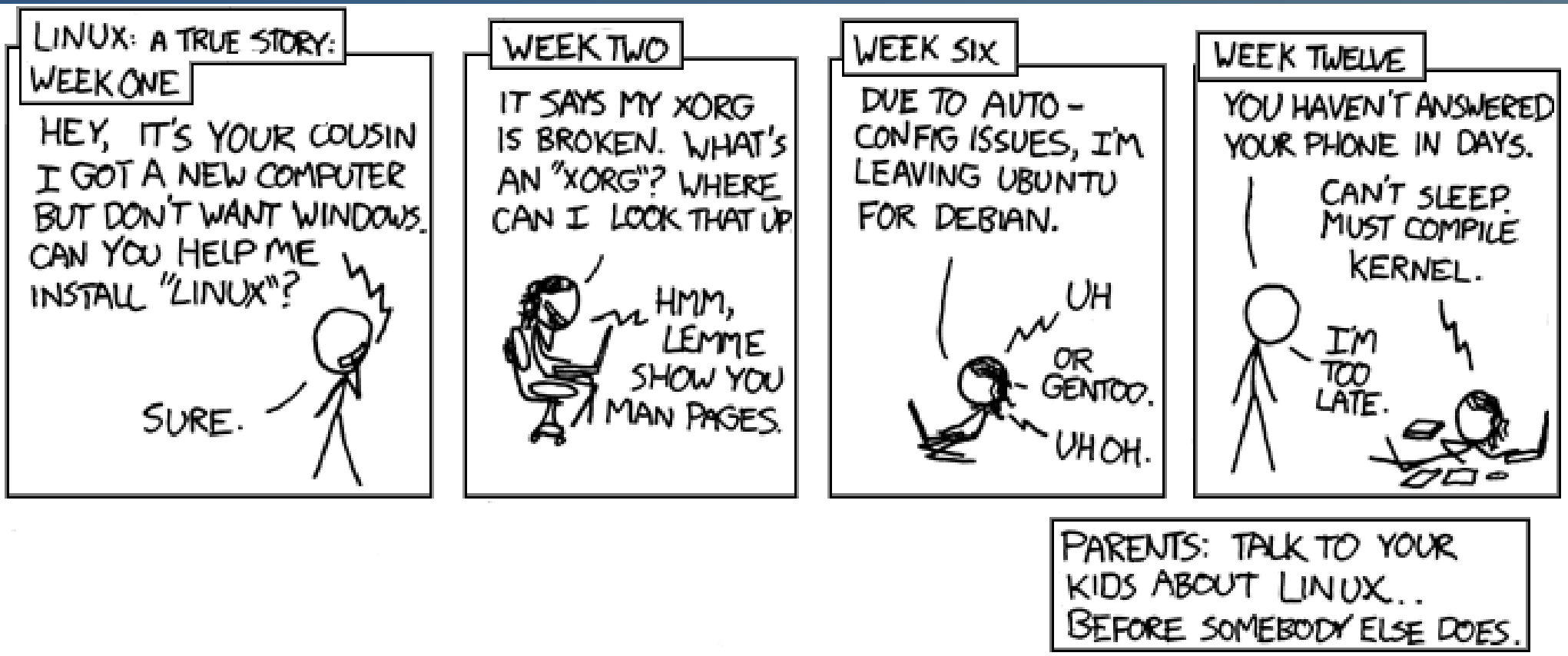
- Exchange contact info (email / Discord / ...)
- Answer the following:

1. What are 3 things that would be systems programming? Which is most interesting?

2. What is one important attribute of doing systems program?

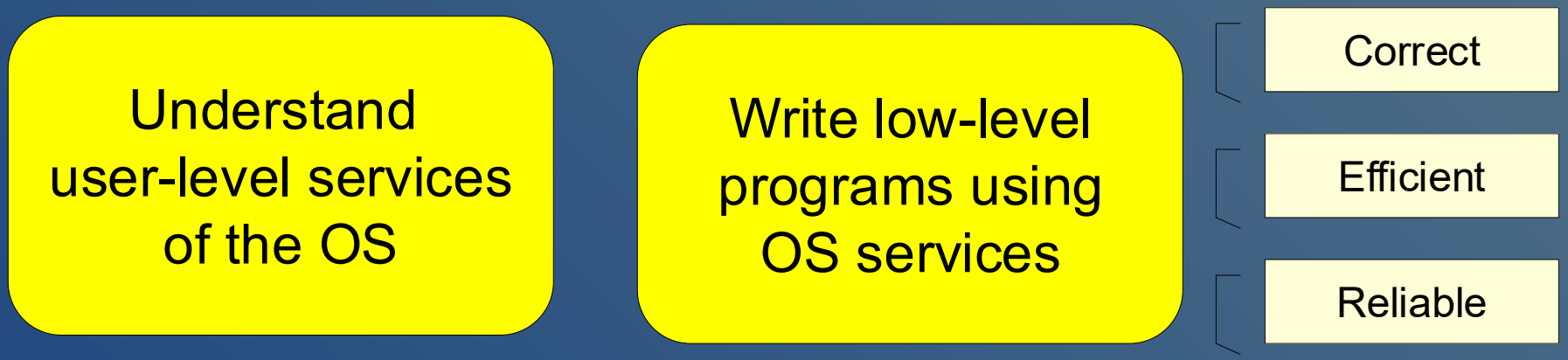
3. What do you think would be hardest about systems programming?

Course Overview



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



ABCD: Perspective

- How would you rate your programming ability?

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

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

- 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

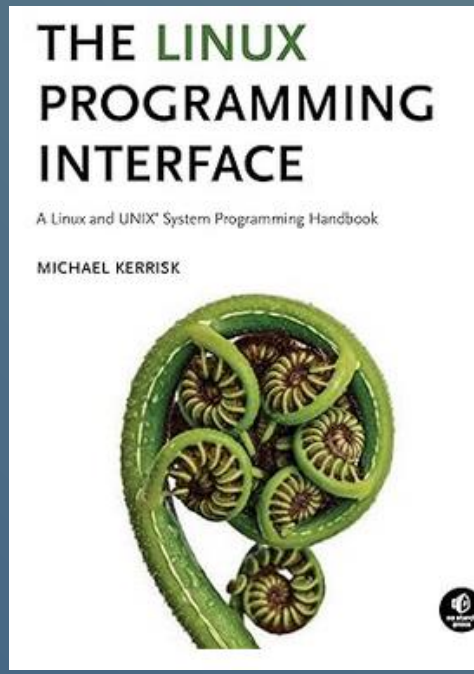
- From prereqs you know
  - Solid programming skills to write functions, loops, if, arrays, pointer, input/output.
  - Solid debugging skills to recognize defects and methodically track down errors.
  - Solid understanding of C or C++ programming.
- Passing prereq does not make this an easy course
  - 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 to my office (TASC1 9215) to read

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

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.

Assignment Marks – Calibrated with Quiz

- https://opencoursehub.cs.sfu.ca/cmpt201/grav/course-info

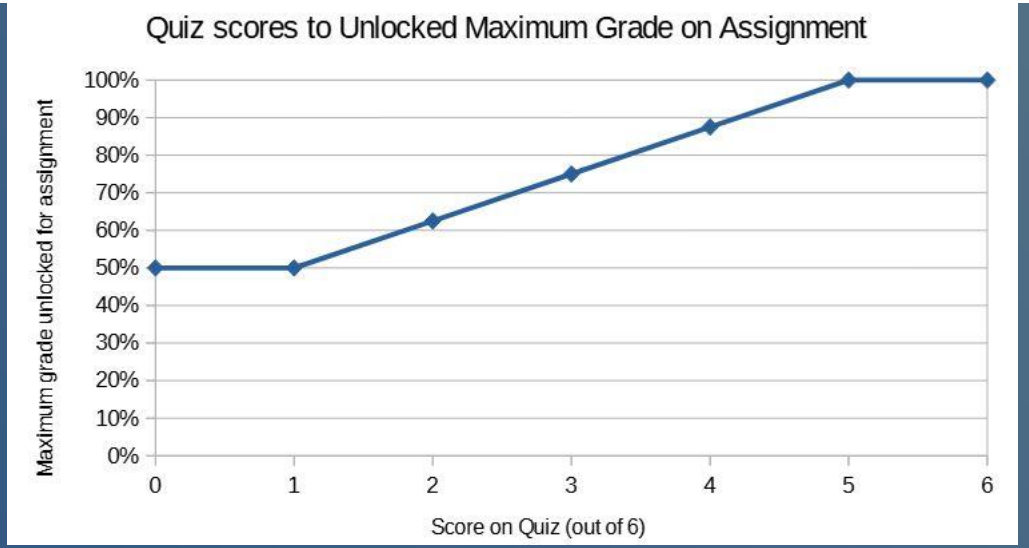
- Assignments are marked as follows:
  - You submit your assignment via pushing it to GitHub and it will automatically be marked by auto-marking scripts after the due date.
  - You will then later, in class, write a short in-class multiple-choice quiz related to the assignment. This is your "quiz mark".
  - Initially the *maximum* mark a student can get on an assignment is 50%. Your score on the related quiz can unlock a higher possible *maximum* mark on the assignment. Specifically, if the quiz is out of 6 marks then:
    - Quiz score of 0 or 1: leaves the maximum grade at 50.0% on the assignment.
    - Quiz score of 2: unlocks a maximum grade of 62.5% on the assignment.
    - Quiz score of 3: unlocks a maximum grade of 75.0% on the assignment.
    - Quiz score of 4: unlocks a maximum grade of 87.5% on the assignment.
    - Quiz score of 5 or 6: unlocks a maximum grade of 100.0% on the assignment.

Assignment Main features:

Keyboard recordings needed

Copy-and-paste operations leads to at least 5.01% penalty

- Fractional score triggers us to inspect whether cheating



Tentative Schedule

	Tues	Thurs	Fri	Sun
Week 1 (1.5-1.11)	Lecture	Lecture	Lab 0 (virtual); Github Name Due	
Week 2 (1.12-1.18)	Lecture	Lecture	Lab 1; A0/A1/A2 Due	Lab 1 Due
Week 3 (1.19-1.25)	(Pre-Recorded Lectures)		Lab 2; A3/A4 Due	Lab 2 Due
Week 4 (1.26-2.1)	Lecture; A0-A4 Quiz	Lecture	Lab 3; A5/A6 Due	Lab 3 Due
Week 5 (2.2-2.8)	Lecture	Lecture	Lab 4; A7 Due	Lab 4 Due
Week 6 (2.9-2.15)	Lecture; A5-A7 Quiz	Lecture	Lab 5; A8 Due	Lab 5 Due
Week 7 (2.16-2.22)	Reading Break			
Week 8 (2.23-3.1)	Midterm; A8 Quiz	Lecture	Lab 6; A9 Due	Lab 6 Due
Week 9 (3.2-3.8)	Lecture; A9 Quiz	Lecture	Lab 7	Lab 7 Due
Week 10 (3.9-3.15)	Lecture	Lecture	Lab 8; A10 Due	Lab 8 Due
Week 11 (3.16-3.22)	Lecture; A10 Quiz	Lecture	Lab 9	Lab 9 Due
Week 12 (3.23-3.29)	Lecture	Lecture	Lab 10; A11 Due	Lab 10 Due
Week 13 (3.30-4.5)	Lecture; A11 Quiz	Lecture	Lab 11	Lab 11 Due
Week 14 (4.6-4.12)	Lecture	Lecture	No Lab; A12 Due	
	Final exam (time and location to be scheduled)			

- Labs: Submit through CourSys; Encourage to finish during lab session
- Due: 11:59 PM

Overall Schedule

CMPT 201 2026 Spring E100 Schedule

Start @	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
8:00 AM							
8:30 AM					**Lab** MRD ASB 9700		
9:00 AM							
9:30 AM					**Lab** MH ASB 9700		
10:00 AM							
10:30 AM					**Lab** MRS ASB 9700		
11:00 AM							
11:30 AM							
12:00 PM							
12:30 PM							
1:00 PM							
1:30 PM							
2:00 PM					**Office Hour** MH Discord		
2:30 PM							
3:00 PM							
3:30 PM							
4:00 PM							
4:30 PM		**Lecture** Linyi Li AQ 3182 *(Cancelled 1/20, 2/17)*		**Lecture** Linyi Li AQ3182 (Cancelled 1/22, 2/19)			
5:00 PM							
5:30 PM							
6:00 PM							
6:30 PM		**Office Hour** Linyi AQ 3182 or TASC1 9215 *(Cancelled 1/20, 2/17)*		**Office Hour** Linyi AQ 3182 or TASC1 9215 *(Cancelled 1/22, 2/19)*			
7:00 PM							
7:30 PM							
…							
11:00 PM					**Assignment Deadlines**		**Lab Deadlines**
11:30 PM							

* Office Hours for TA MRD and MRS will be determined soon.

Policies

- Late
  - For labs, no late
  - For assignments,
    - Have a total of 7 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 write 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 buildin 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)

Policies

- AI Tools

Forbidden uses of AI

- Do not use AI to write any of your code (labs or assignments).
  - Do not ask AI to write code. Hint: If you are prompting AI, add "without showing me any code, explain how..."
  - Do not take any code from an AI system. If it outputs the code, then it's not your code.
- Do not use AI to read or change any of your code.
  - Do not copy-paste or upload your code to AI.
  - Do not use local tools to analyze your code.
  - Do not ask AI to find a bug, suggest a change, check if you met assignment requirements, refactor your code, add comments, check for style violations, write code based on your comments or directions, translate your code between programming languages, translate your code between human languages.
- Do not copy the assignment or lab description into an AI tool.
  - Do not have AI summarize assignment requirements.
  - Do not have AI translate assignment requirements into another language.
- Do not use an AI agent to try and pass assignment test cases.
- 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."

Reminder: You must write all your assignment code in NeoVim, and all of your development process must be recorded. You may not develop using a different tool (such as VS Code or Cursor). You may not develop your code and then later retype your code while the recording tool is on. Your development work must be recorded. If you have an error when recording your development work you must:

1. Clearly list in your `.c` file what code was developed without a recording.
2. Email your instructor and let them know of the situation.

If the recording does not show all of your development process we will have no evidence that you developed it yourself and assume that it was written using AI tools or disallowed tools.

Coding practice (assignments) are heavily weighted to encourage everyone to solve problems and learn to code. We give grades for what *you* have done,

Demo

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

  https://opencoursehub.cs.sfu.ca/cmpt201/grav/resources

- Install Docker (return students please pull new image)
- 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
  - Roughly 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