SFU SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

ENSC 254: Introduction to Computer Organization

Introduction and Logistics

Yingjie Li, PhD
Assistant Professor
Computer Engineering, SFU
Email: yingjie_li@sfu.ca

Whenever There's Trouble, Just Yelp for Help!

Outline

- !!Course Format!!
- Why Care About This Course
- What You Should Already Know
- What You Will Learn
- !!Grading and Important Logistics!!
- More Information

Course Format and Webpage

- **It will be F2F instruction with Zoom streaming.**
  - **Zoom link:** https://sfu.zoom.us/j/6669638387
- **Course Webpage:**
  - **https://coursys.sfu.ca/2026su-ensc-254-e1/**

Textbook Chapters

- **Textbook**
  - Computer Organization and Design RISC-V Edition: The Hardware Software Interface. 2nd Edition. By David Patterson and John Hennessy. Publisher: Morgan Kaufmann. (https://www.cs.sfu.ca/~ashriram/Courses/CS295/assets/books/HandP_RISCV.pdf)

- **Relevant to this slide set:**
  - Section 1.1-1.3, 1.10

- **Relevant to the next slide set:**
  - Sections 1.4-1.9, 1.11

Outline

- !!Course Format!!
- Why Care About This Course
- What You Should Already Know
- What You Will Learn
- !!Grading and Important Logistics!!
- More Information

Why Care About This Course?

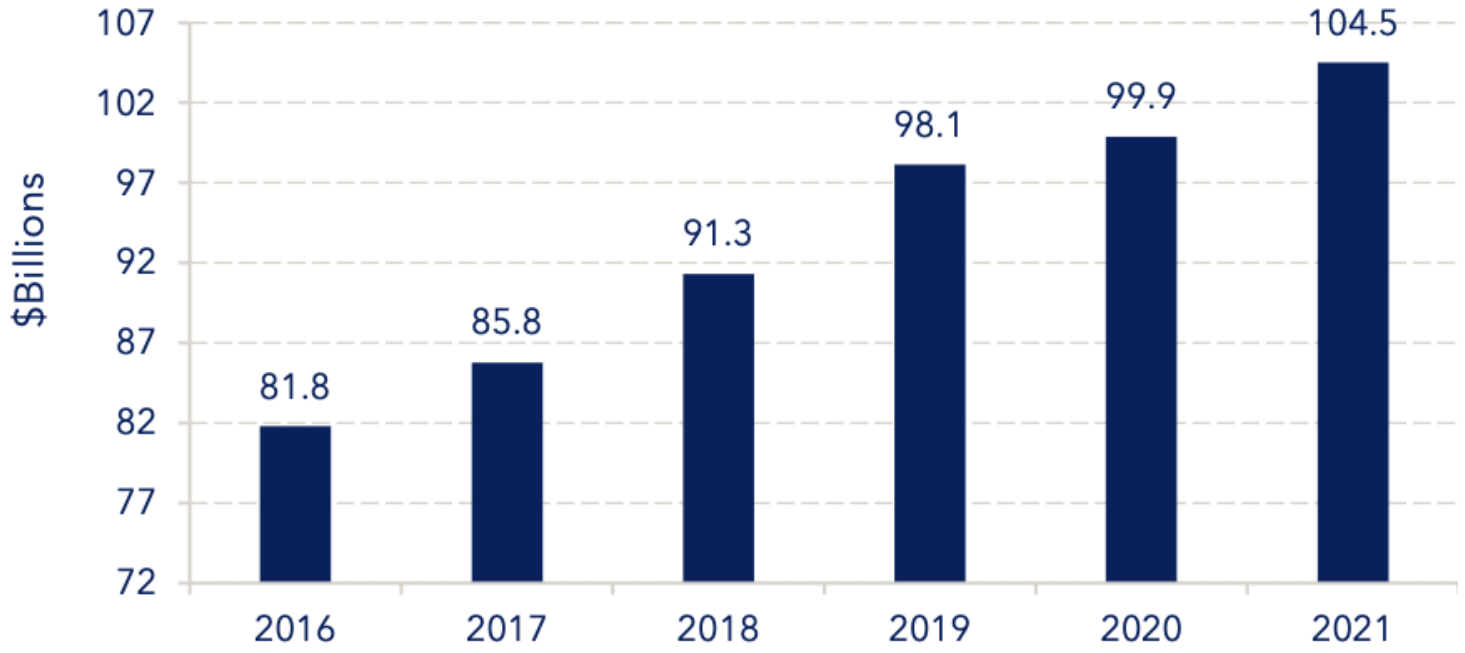
- **Pessimistic view:**
  - **It's "required" – a core course for all options and a prerequisite for many other courses**

- **Optimistic view:**
  - **Assume you are now seeking your next coop: the skills learned in this course will be crucial**
  - **You probably already had your first coop and knew how important software expertise is**
  - **Hardware/software codesign skills will further set you apart**

Significance of IT Industry

SFU

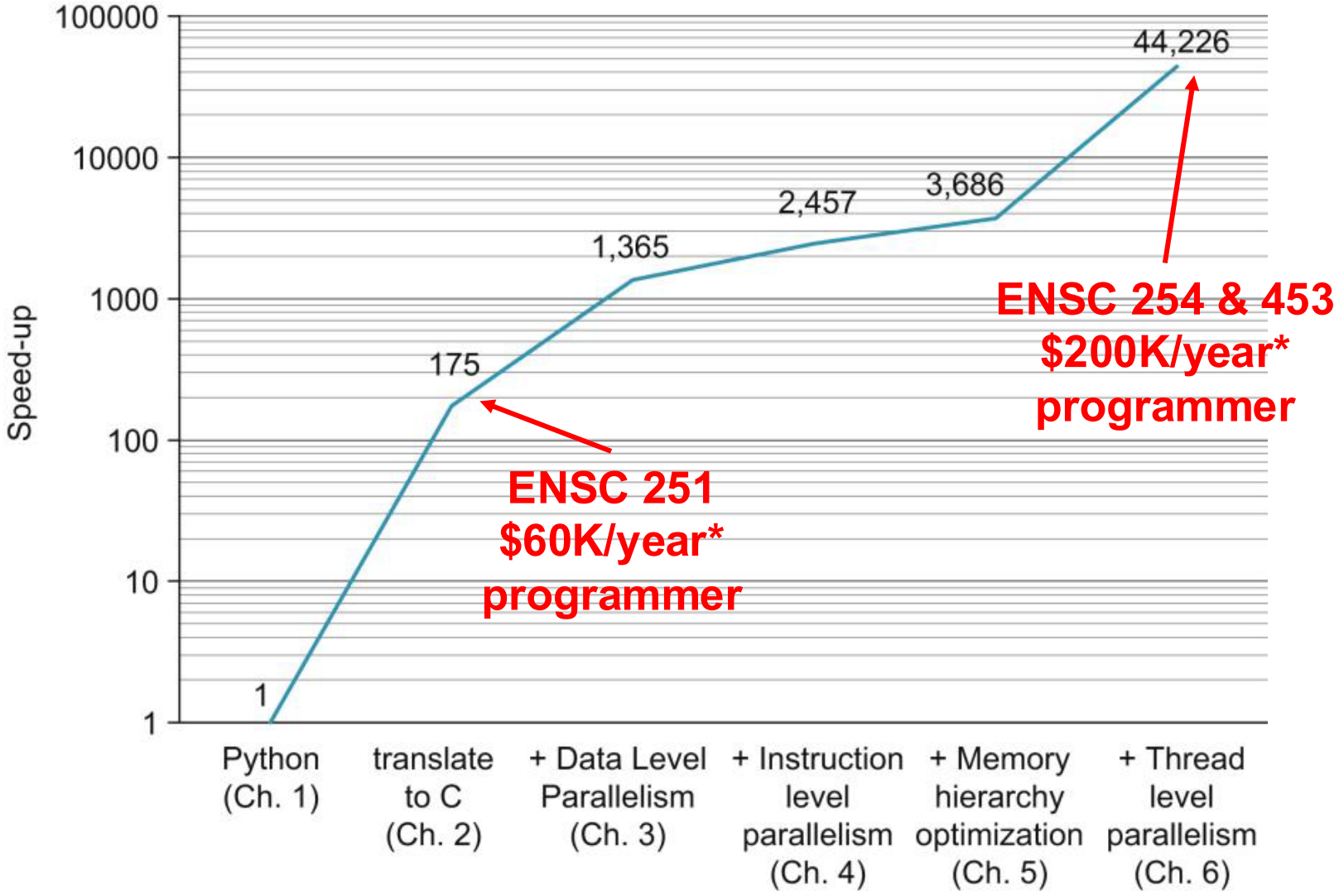
- **IT counts for about 10% of US GDP**
  - **Source: Our textbook**
- **IT counts for about 5.3% of Canada GDP**
  - **Source: Innovation, Science and Economic Development Canada**



Hardware/Software Codesign Shines

SFU

Matrix multiply: speedup over naïve Python version



***Salary numbers are not from actual data, just for illustration purpose.**

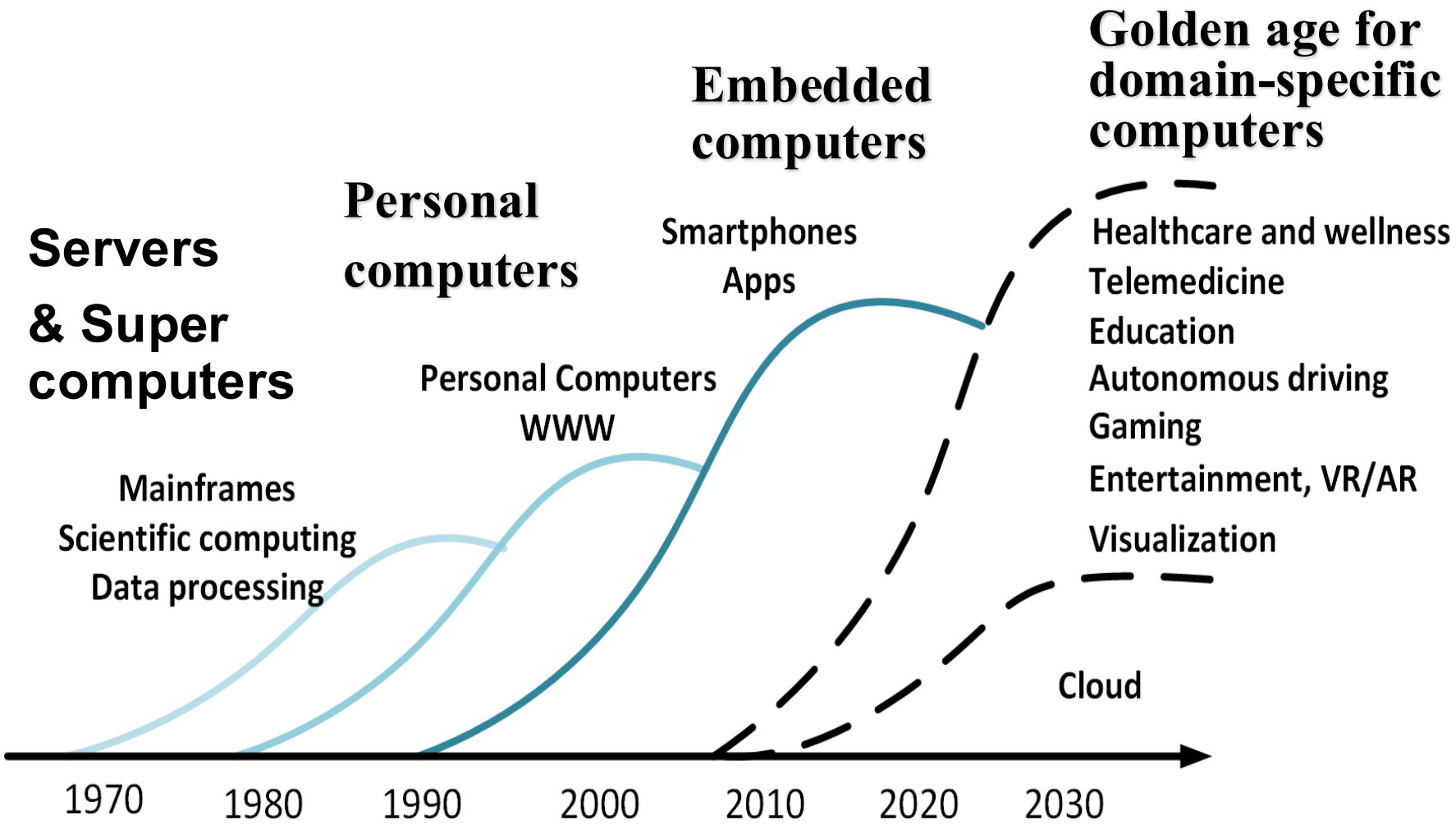
Which Companies Need Such Skills?

- Almost all major software companies (incomplete)

- Almost all major hardware companies (incomplete)

And a lot more

Computers Are Pervasive

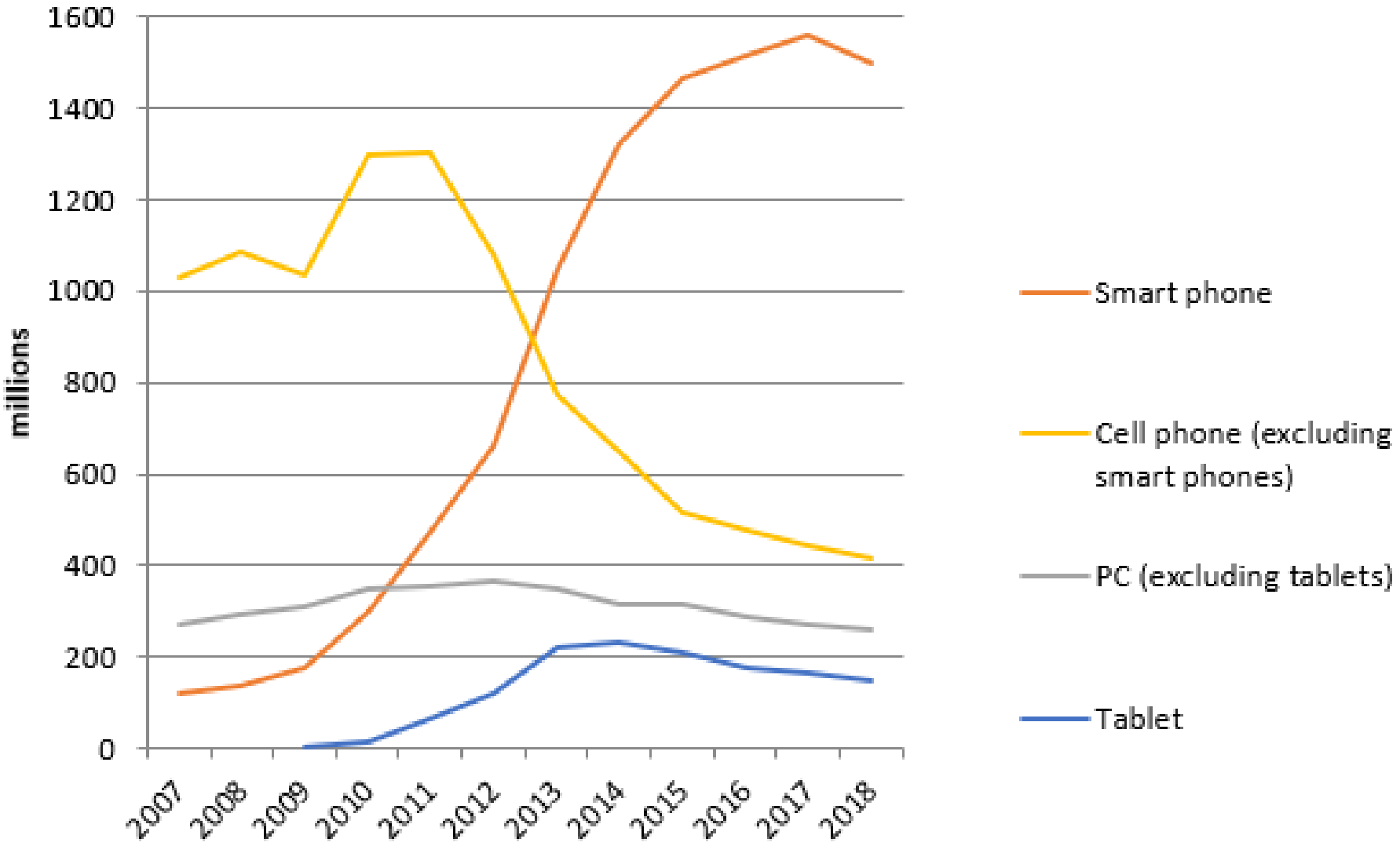


The computer revolution, UC Berkeley, ESSCIRC 2018

Personal Computers (PC) / Tablets

- **Personal computers (PC)**
  - General purpose, variety of software
  - Subject to cost/performance tradeoff

The PostPC Era



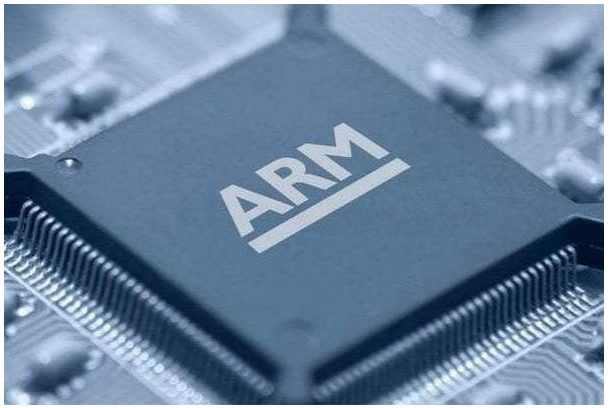
The PostPC Era

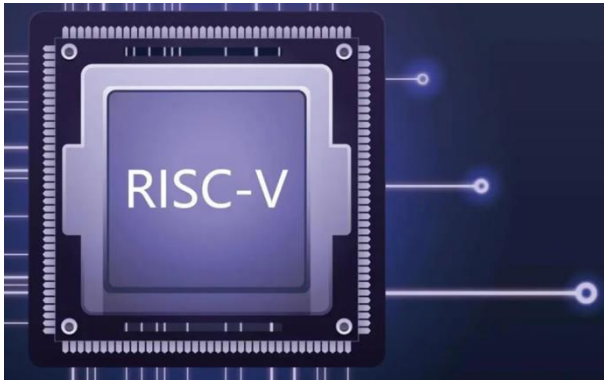
- **Personal Mobile Device (PMD)**
  - Battery operated
  - Connects to the Internet
  - Hundreds of dollars
  - Smart phones, tablets, electronic glasses

- **Cloud computing**
  - Warehouse Scale Computers (WSC)
  - Software as a Service (SaaS)
  - Portion of software run on PMD and portion run in Cloud
  - Amazon and Google

Embedded Computers

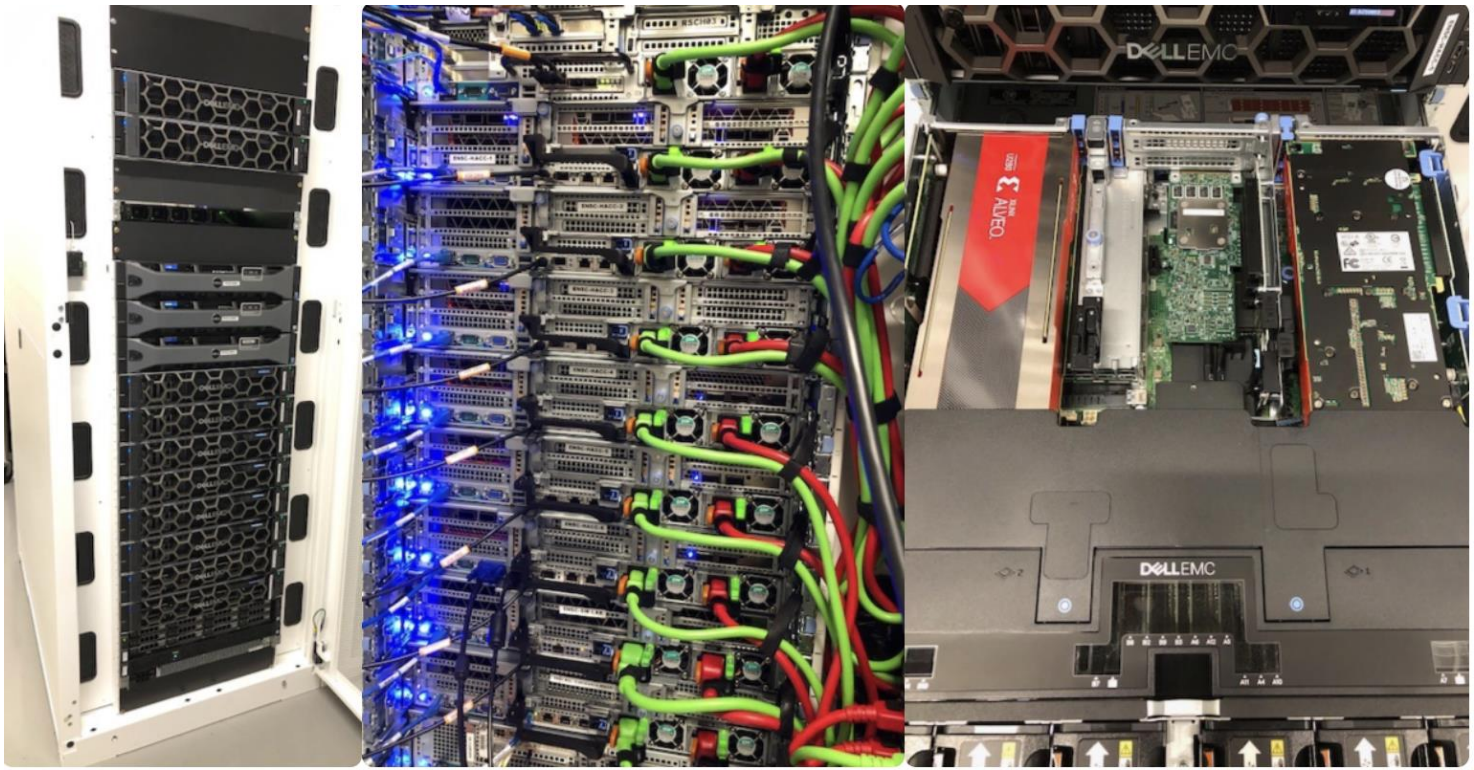
- **Embedded computers**
  - Usually hidden inside the device
  - Designed for a specific purpose
  - Often has limited memory, power, and processing capability
  - Usually interacts with sensors, motors, buttons, displays, or networks





Servers

- **Server computers**
  - **Network based**
  - **High capacity, performance, reliability**
  - **Range from small servers to building sized**



Supercomputers / Datacenters

- **Supercomputers**
  - Type of server
  - High-end scientific and engineering calculations
  - Highest capability but represent a small fraction of the overall computer market

SFU Cedar: Most Powerful Academic Supercomputer in Canada

Outline

- !!Course Format!!
- Why Care About This Course
- What You Should Already Know
- What You Will Learn
- !!Grading and Important Logistics!!
- More Information

What You Should Already Know

- **ENSC 251: Basics of software programming in C++**
  - Function, loop, control, data type, variable, operation
  - Basic algorithms and data structures, e.g., arrays and pointer-based data structures and associated algorithms
  - Dynamic memory allocation and management
  - Software debugging and testing
  - Programming in the Linux environment
- **ENSC 252: Basics of digital hardware design**
  - Gates, truth tables, and logic equations
  - Combinational and sequential logic
  - Memory elements: Flip-flops, latches, registers, and RAM
  - Finite state machine (FSM) and control logic

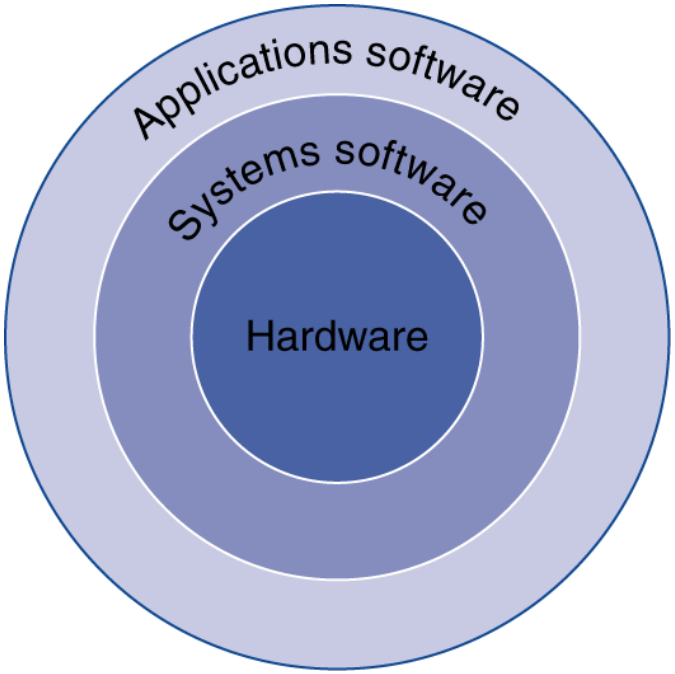
Outline

- !!Course Format!!
- Why Care About This Course
- What You Should Already Know
- What You Will Learn
- !!Grading and Important Logistics!!
- More Information

What Will You Learn?

- How programs are translated into the machine language and how the hardware executes them
- The hardware/software interface (abstraction)
- What determines program performance and how it can be improved
- How hardware designers improve performance
- Recent trends in computer architecture
  - Introductory. More in ENSC 453

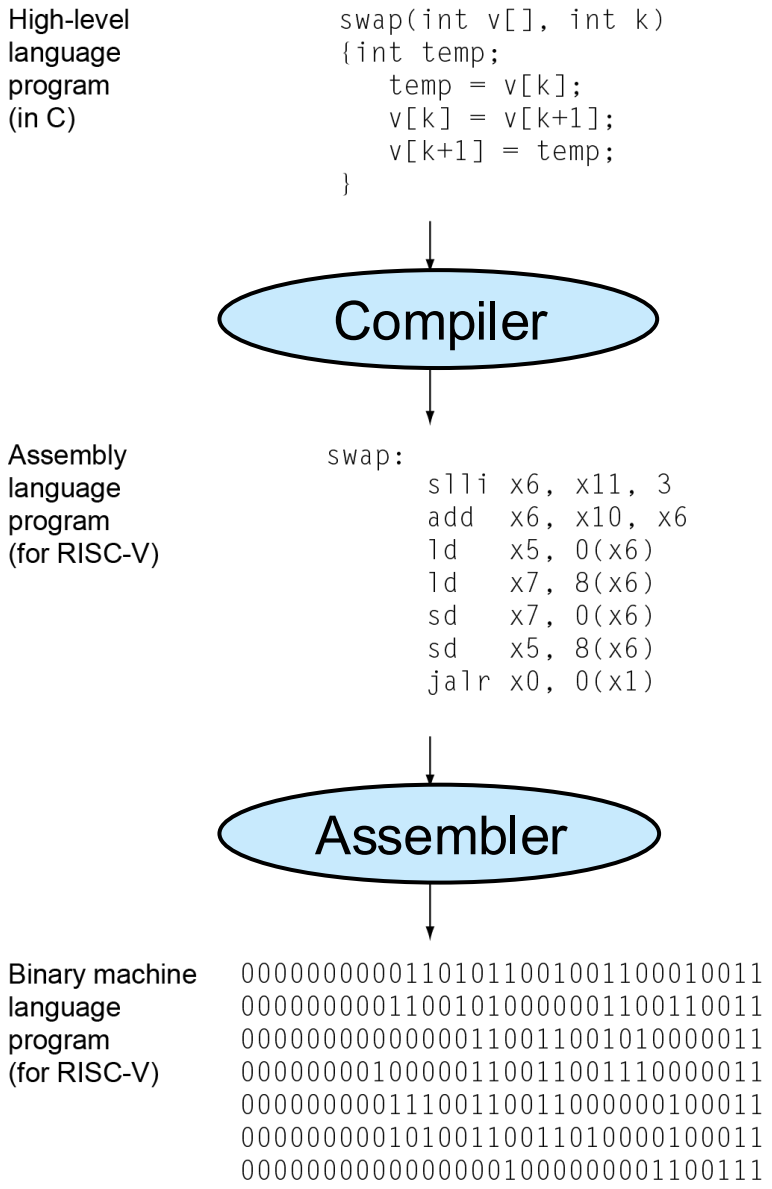
Below Your Program



- **Application software**
  - Written in high-level language (HLL), e.g., C/C++, Java
- **System software**
  - Compiler: translates HLL code to machine code
  - Operating System: service code
    - Handling input/output
    - Managing memory and storage
    - Scheduling tasks & sharing resources
- **Hardware**
  - Processor, memory, I/O controllers

Levels of Program Code

- **High-level language**
  - Level of abstraction closer to problem domain
  - Provides for productivity and portability
- **Assembly language**
  - Textual representation of instructions
- **Machine language (HW representation)**
  - Binary digits (bits)
  - Encoded instructions and data

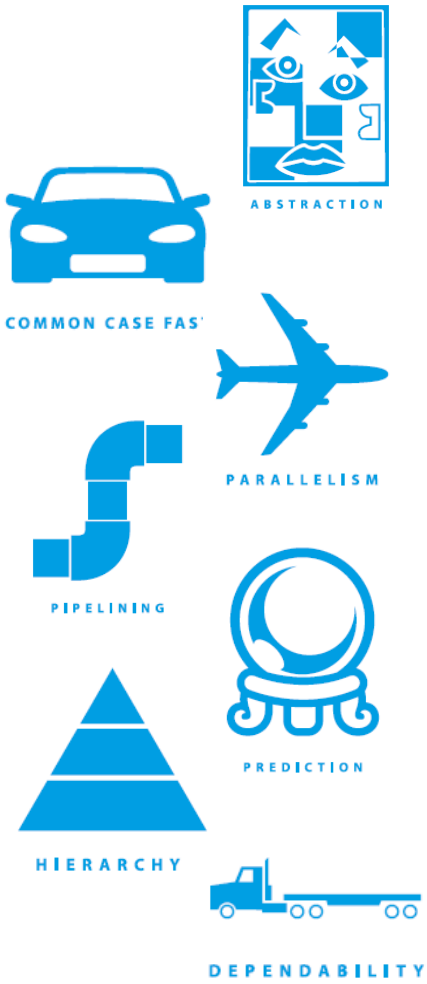


Understanding Performance

- **Algorithm**
  - Determines number of operations executed
- **Programming language, compiler, architecture**
  - Determine number of machine instructions executed per operation
- **Processor and memory system**
  - Determine how fast instructions are executed
- **I/O system (including OS)**
  - Determines how fast I/O operations are executed

Seven Great Ideas

- Use ***abstraction*** to simplify design
- Make the ***common case fast***
- Performance *via* ***parallelism***
- Performance *via* ***pipelining***
- Performance *via* ***prediction***
- ***Hierarchy*** of memories
- ***Dependability*** *via* redundancy



Course Schedule Overview

- One tutorial to refresh your memory on C++ programming and debugging
  - Basics of struct, union, pointers, and bit-level operations
  - Basics of debugging with gdb (cgdb) and Valgrind (optional)
- Intro to computer abstraction and technologies
  - One lecture, Chapter 1
- The hardware/software interface: RISC-V ISA
  - Four lectures, plus tutorial, Chapter 2
- The processor design
  - Five lectures, plus tutorial, Chapter 4

Course Schedule Overview

- The memory hierarchy design
  - Five lectures, plus tutorial, Chapter 5
- Integer and floating-point arithmetic for computers
  - Two lectures, Chapter 3
- Recent trends of parallel computing and domain-specific computing
  - Two lectures, Chapter 6
- Potential guest lecture on RISC-V processor design

Outline

- !!Course Format!!
- Why Care About This Course
- What You Should Already Know
- What You Will Learn
- !!Grading and Important Logistics!!
- More Information

Course Grade Breakdown

- 40 % Lab Assignments
- 20 % Homework Assignments
- 15 % Midterm Exam
- 25 % Final Project
- 5% Class Participation (Pop-up Quiz)
- 2 % Course Survey Bonus
- 1 % TA Evaluation Bonus (if applicable)
- Total marks: up to 108%

Lab Assignments

- **Five lab assignments in total (40% marks)**
  - Information on each lab will be posted on the course home page
- **Lab 1 weighs 4% of the final marks**
  - It is done and graded individually
  - This is a warmup lab to prepare you for lab 2 to 5, and also serves as a reference point for you to find your teammate for lab 2 to 5
- **For lab 2 to 5, each lab weighs 9% of the final marks**
  - Students need to form a 2-people lab group and work together
  - Both students should be in the same lab session, and both should attend the lab demo
  - To find the right teammate for your lab group, you could ask for their lab 1's marks as a reference point

Lab Assignments (cont.)

- **For lab 2 to 5, each lab weighs 9% of the final marks**
  - **Assume each lab has X points**
    - **At the end, this will be scaled to 9% of the final marks: (your_points / X) * 9% final marks**
  - **Both students get the same score except rare corner cases**
  - **In rare corner cases, one partner may not contribute much. Please document each of your own contributions and submit it along with your lab submission. TA will grade each of you individually.**

Lab Assignments (cont.)

- **You are required to demonstrate your work in lab session**
  - For each lab, TA will randomly select 1/3 to 1/2 of the groups (for lab 1, individuals) of students to do the demo
  - The lab has to be demonstrated on a FAS-RLA lab computer F2F
    - If you need some special accommodation, let TA know
  - If neither student understands the code, then both are considered as cheating
  - If student 1 performs good with the demo, and student 2 doesn't understand the code, then student 2 is considered as cheating
  - When a student is considered as cheating, he/she gets 0 mark for that lab and there are more consequences (more details later)
    - *If you didn‘t work on the lab, be honest to get a 0 instead of a cheating*

Lab Assignments (cont.)

- **Electronic submission of your code is required**
  - **You need to meet the deadline**
  - **Every 10 minutes late submission, you lose 10% of this lab's points; that is, 100 minutes late, you will get 0 point on this lab**
- **You *CANNOT* copy code from others or from website**
  - **This is considered as cheating, you will get 0 mark for the lab**
- **You can leverage ChatGPT for hints, but *CANNOT* directly copy code from ChatGPT without understanding**
- **You *CANNOT* give your lab code to other groups**
  - ***If you do so, both you and the group copying your code will be considered as cheating and awarded 0 mark for the lab***
- **You *CANNOT* post your code to the public (e.g., gitHub)**

F2F Lab Session

- **F2F Lab Session: ASB 9815**
  - Detailed schedule of lab sessions and TAs in each session will be released soon
  - TA will be available in the lab session for evaluating your lab demonstration
  - Please show up on time for your assigned lab demo time slot; being late, you will lose 10% of that lab's points
- **Except the lab demos, lab sessions will be used as TA office hours for Q&A**
- **Except lab sessions, Q&A can be posted on discussion board (highly encouraged), which can benefit all students**

Lab Equipment

- **Your SFU login will give you access to the lab machines**
  - **All access will be remote login using ssh/guacamole (Linux environment)**
- ***You are responsible for the well being and proper care of the equipment in the lab***
  - ***Destruction or vandalism or hacking the lab servers will result in FAILURE of the class***
- ***Observe the lab safety signs and follow the rules***

Final Project

- **Final project weighs 25% of the final marks**
  - You will build a cycle-accurate simulator for RISC-V CPU and its cache memory system, based on lab 2 to 4.
  - Information on each lab will be posted on the course home page
  - Similar rules to lab assignments
- **You will work in a 4-student project team (2 lab teams)**
  - You could ask for their lab 2's marks as a reference point
  - You will need to write a document and demo code to your TA
  - The project will be graded per team
  - In rare corner cases, members may not contribute equally. Please document each of your own contributions and submit it along with your lab submission. TA will grade each of you individually.

Homework Assignments

- **Eight homework assignments in total (20% marks)**
  - **Homework is for individual work, each student will be graded individually**
  - **Each homework weighs about 2-3% of the final marks**
    - **There might be some small variances, which will be detailed in each homework assignment**
  - **Information on each homework will be posted on the course home page**
  - **Similarly, electronic submission of your homework is required, and you need to meet the deadline**
  - **Similarly, no cheating is allowed**
  - **No demos for homework assignment**

Midterm Exam

- **Midterm exam (15% marks)**
  - **Time: Jul 10, 4:30pm-6:20pm**
  - **Location: TBD**
  - **Covers the topics in Lectures 2-13; we will have a midterm review before the exam**
- **The midterm is an open book exam**
  - **You can bring the paper-based material, e.g., textbook, slides, homework, paper notes, etc.**
  - **Electronic devices (e.g., phones, laptops, tablets, calculators, etc.) are NOT allowed**
- **NO final exam**

Academic Concession

- **If you encounter some difficulty due to medical reasons, please submit the FAS Academic Concession form: https://coursys.sfu.ca/forms/apsc-fas-academic-concession/**
- **We will try our best to make some accommodations**
  - **If you miss one lab assignment, talk with your teammate first. We can scale that lab mark for you based on the average marks of your other lab assignments**
  - **If you miss one or two homework assignments, we can scale that homework mark for you based on the average marks of your other homework assignments**
  - **We can only accommodate up to one lab assignment, two homework assignments.**
  - **You MUST provide a doctor's note to be eligible**

Class Participation Bonus

SFU

- **Class Participation: 5 Marks**
  - **Only applicable for in-person participation**

Course Survey *Bonus*

- **2 marks course survey bonus:**
  - **One survey around midterm time frame, 1 mark**
  - **The other survey around final time frame, 1 mark**
    - **Note for both surveys, it will be based on the participate rate. If it exceeds a threshold (e.g., 80%), all of you will get 1 mark; otherwise, nobody will get the bonus mark**

- **These surveys will be used to improve the course quality, both for the second half of the term, and for future offerings**
  - **Your valuable feedback is highly appreciated!**

TA Evaluation *Bonus*

- **1 mark for TA Evaluation bonus**
  - **ENSC usually has an in-class TA evaluation for each course, specifically asking feedback for course TAs**
  - **Only if ENSC asks for TA evaluation and you finish the TA evaluation; otherwise, 0 bonus**

Guest Lectures

- **Attendance of all guest lectures is mandatory.**
  - **If there is a guest lecture**
- **You will be required to sign in to indicate your attendance**
- **Failure to attend (without a medical note) will result in a 2% mark deduction off your final marks**

!! No Cheating Allowed !!

!! No Cheating Allowed !!

- Any cheating in a *lab* will lead to *zero* for that lab
- Any cheating in a *homework* will lead to *zero* for that homework
- Any cheating in an *exam* will lead to *zero* for that exam
- Any cheating in final project will lead to *zero* for final project
- Any cheating will lead to *zero* bonus

- *Two cheatings in any combination* will lead to *failure* of the course. *Furthermore, the instructor will recommend to the director that an FD (failure for academic dishonesty) be awarded*

- More details in "Statement of Expected Professional Integrity Standards and Consequences", which you have to sign before your lab assignments will be marked

ChatGPT (or Similar AI Bots) Policy

- **The usage of ChatGPT or similar AI bots is allowed in this course to**
  - **Help students learn the course concepts, and**
  - **Help students find clues to the solutions of homework and lab assignments.**
- **However, students MUST take full responsibility of leveraging the hints and answers from these AI bots**
  - **Your marks are graded based on what you submitted**
  - **More importantly, students need to master the concepts and skills to finish the assignments after they leveraged the help**
  - **For lab assignments, there will be lab demos: if you cannot confidently explain the code that you submitted (whether you leveraged AI bots or not), you will be considered as CHEATING.**

Tentative Grade Award Breakdown

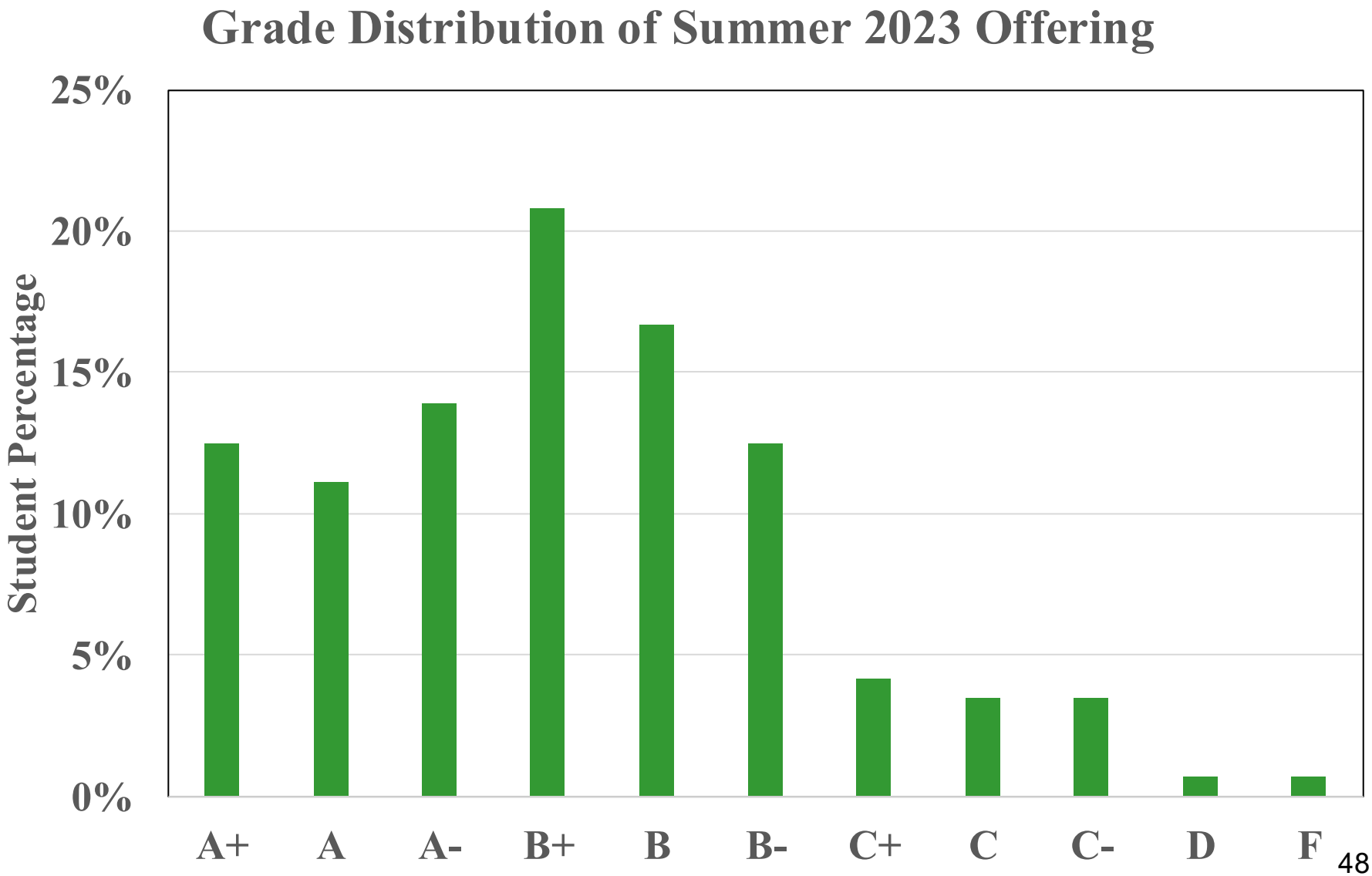
SFU

- **> = 95%** **A+**
- **90 – 94%** **A**
- **85 – 89%** **A-**
- **80 – 84%** **B+**
- **75 – 79%** **B**
- **70 – 74%** **B-**
- **65 – 69%** **C+**
- **60 – 64%** **C**
- **55 – 59%** **C-**
- **50 – 54%** **D**
- **< 50%** **F**

Note: the total marks are up to 103%

Note: I reserve the right to scale the grade

Grade Distribution from Summer 2023



A Secret Weapon to Succeed in This Course and in Your Career!

SFU

Practice, Practice, Practice!

Outline

- !!Course Format!!
- Why Care About This Course
- What You Should Already Know
- What You Will Learn
- !!Grading and Important Logistics!!
- More Information

Contact Info

SFU

- **Email: yingjie_li@sfu.ca**
- **Online Instructor Office Hours**
  - **Recommended: please bring your questions to live class or post your questions on discussion board anytime. This can benefit all other students as well.**
  - **If you prefer one-to-one office hours, please send me an email to book a meeting.**
- **Teaching Assistants**
  - **Rashid Zamanshoar Heris <rza93@sfu.ca>**
  - **Reyhaneh Ahani Manghotay <raa112@sfu.ca>**
  - **Lester He <wha86@sfu.ca>**
  - **Mohammad Reza Sadeghian <mrs27@sfu.ca>**
  - **Qihan Xu <qxa18@sfu.ca>**
  - **Zhenyu Zhang <zza320@sfu.ca>**

Course Discussion Board Policy and Disclaimer

SFU

Please do not use these forums to post any material that is knowingly false and/or defamatory, abusive, vulgar, hateful, obscene, threatening, invasive of a person's privacy, or otherwise in violation of any law. Do not post any copyrighted material unless the copyright is owned by you. I reserve the right to remove any messages posted and to reveal your identity (or whatever information is known about you) in the event of a complaint or legal action arising from any message posted by you.

By posting your message, you agree to indemnify me, my employees, agents and representatives, and to hold them harmless from any and all claims and liabilities (including lawyers' fees) resulting from any material posted to these forums, or from any acts resulting from participants' use of these forums.

Course References

- **Course textbook:**
  - David Patterson, John Hennessy. Computer Organization and Design RISC-V Edition: The Hardware Software Interface. 2nd Edition, Morgan Kaufmann, 2020 (ISBN: 9780128203316).
- **Other recommended books:**
  - Randal E. Bryant and David R. O'Hallaron, Computer Systems: A Programmer's Perspective, 3rd Edition (CS:APP3e), Pearson, 2016 (ISBN 0-13-409266-X).
- **Other recommended references:**
  - !!Always Google first!!
  - C++ Reference: http://www.cplusplus.com/reference/
  - Stack overflow: https://stackoverflow.com/
  - ChatGPT: https://openai.com/blog/chatgpt

Acknowledgements

- **Special thanks to**
  - **Dr. Zhenman Fang**
  - **The textbook authors David Patterson and John Hennessy and the Morgan Kaufmann publisher**
    - **Many of their slides are used/extracted/adapted in this course**
  - **Yonghong Yan, University of North Carolina at Charlotte, and Caroline Liu, Justin Yokota, Peyrin Kao, UC Berkeley**
    - **Some of their slides are used/extracted/adapted in this course**
  - **Arrvindh Shriraman, Alaa R. Alameldeen, SFU, and Randal E. Bryant, David R. O'Hallaron, CMU**
    - **Some of their lab assignments are used/adapted in this course**

Thanks to all of them for supporting the redesign of ENSC254