CS 250: FOUNDATIONS OF COMPUTER SYSTEMS
[INTRODUCTION]

The Systems Arsenal
Looking to solve a problem?
Bring systems thinking to the fore

Vault to different vantage points
For a holistic problem view

Know where you need to be
Before you set off

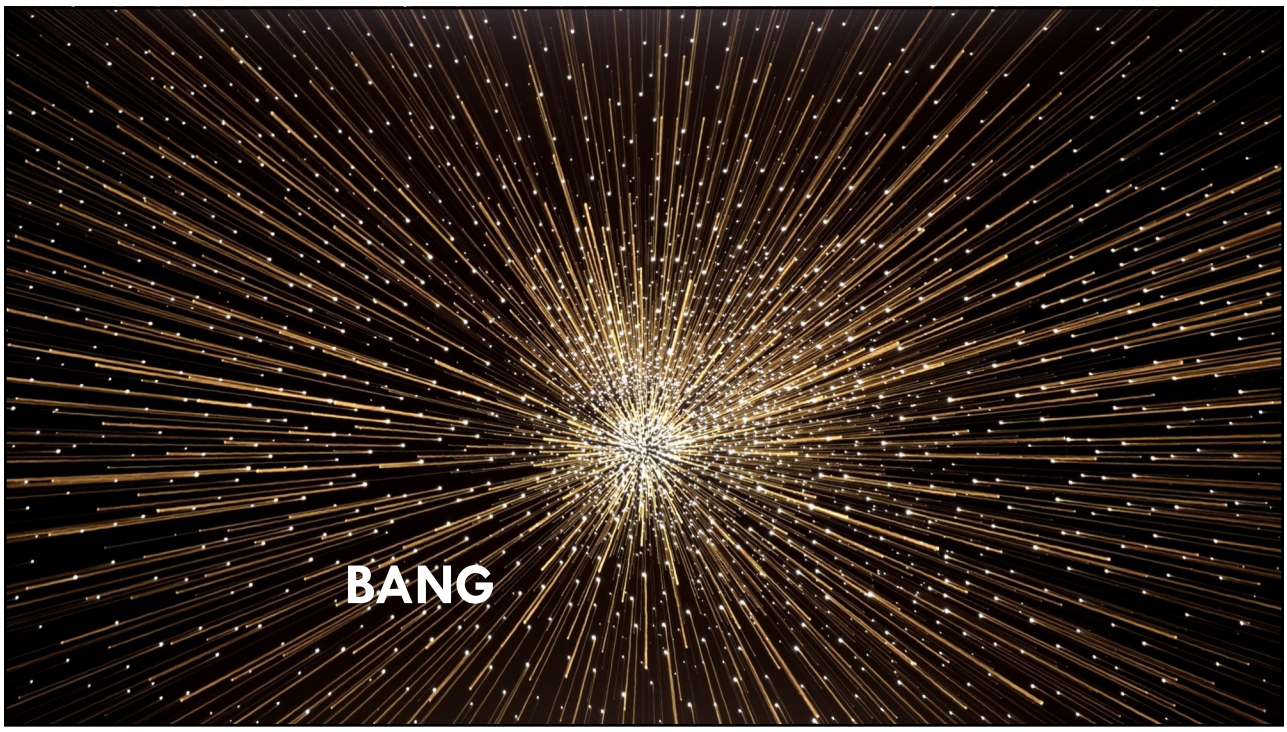
SHRIDEEP PALLICKARA
Computer Science
Colorado State University

1

Topics covered in this lecture

- Introduction
- Course overview and expectations
- Communications

2



3

BANG

- It's been argued that 21st century folks ought to familiarize themselves with the key ideas underlying **BANG**
  - Bits, Atoms, Neurons, and Genes (BANG)
- Science has been remarkably successful in uncovering their core ideas
  - Quite possible we may never fully grasp how atoms, neurons, and genes *actually* work
- A consoling exception?
  - Bits and computing systems at large

4

Gaining a gestalt understanding of how the machine works

- The interactions between hardware and software were simple and transparent enough to produce a *coherent* picture
- Alas, as digital technologies have become increasingly more complex, this clarity is all but lost
  - Hidden under many layers of obscure interfaces and proprietary implementations

Inevitable consequence of this complexity?

- **Specialization**
  - Pursuit of many *niche* courses, each covering a single aspect of the field
- Many computer science students are missing the forest for the trees
  - Marshaled through a series of courses in programming, theory, and engineering
    - Without pausing to appreciate the beauty of the picture at large
- We will strive to do this
  - Allowing you to view problems (and devise solutions) from multiple vantage points

ABOUT ME

COMPUTER SCIENCE DEPARTMENT COLORADO STATE UNIVERSITY

7

About me

- I do research in the area of large-scale computing systems, Big Data, and GeoAI
- My research has been funded by agencies in the United States and the United Kingdom
  - These include the National Science Foundation, the Department of Homeland Security (including the *Long Range* program), the Environmental Protection Agency, the Department of Agriculture, the National Institute of Food & Agriculture, the National Endowment for the Humanities/Teagle and the U.K's e-Science program
  - Recipient of the National Science Foundation's CAREER Award
  - I direct the Center for eXascale Spatial Data Analytics and Computing (XSD) @ CSU [**https://spatial.colostate.edu**]

COLORADO STATE UNIVERSITY Professor: SHRIDEEP PALLICKARA COMPUTER SCIENCE DEPARTMENT INTRODUCTION L1.8

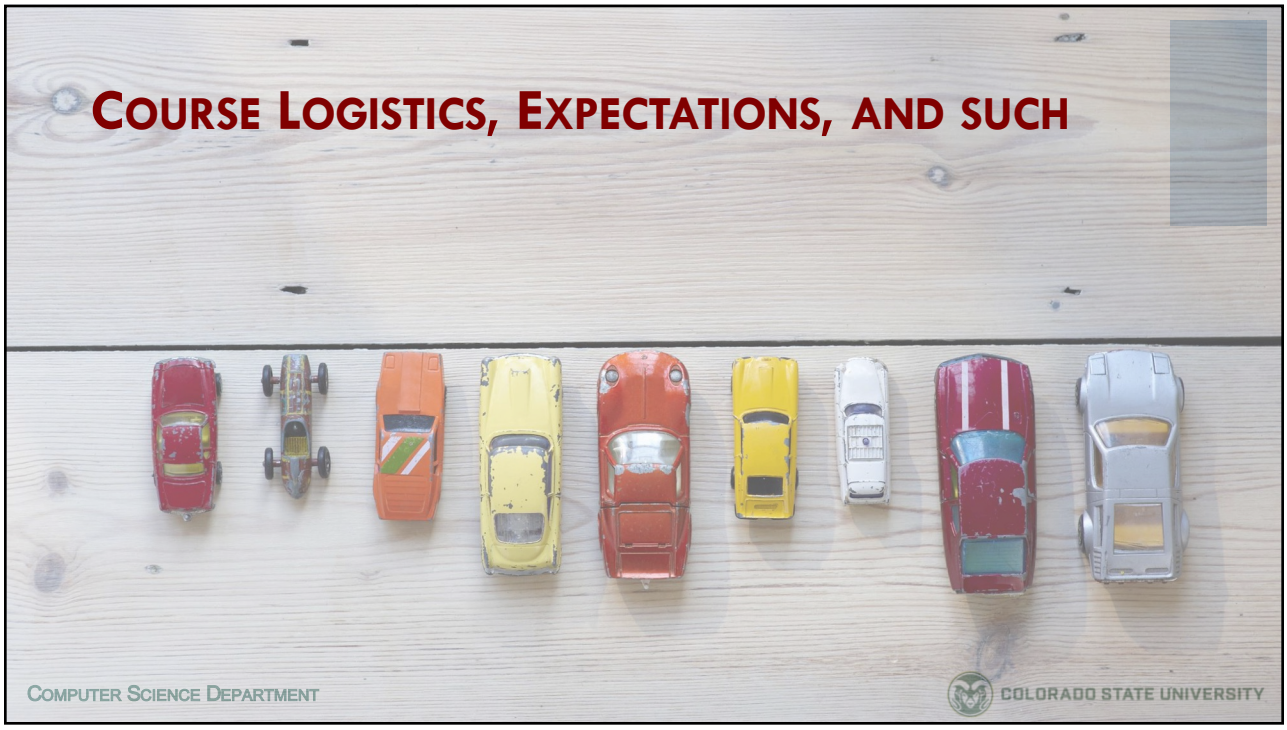
8

Areas that I have worked or actively work in include:

- Big Data
- GeoAI
- Cloud computing and analytics
- Internet-of-Things (IoT)
- Content dissemination and streaming systems
- Grid computing
- P2P systems
- Object Request Brokers

My research has been deployed in

- Urban sustainability
- Epidemic modeling
- Precision agriculture
- Commercial internet conferencing systems
- Defense applications
- Earthquake sciences
- Environmental monitoring
- Healthcare informatics
- High energy physics
- Visualizations



11

Course webpage

- All course materials will be on the course webpage **http://www.cs.colostate.edu/~cs250**
  - Schedule
  - Lectures
  - Assignments
  - Syllabus
- Grades will be posted on **Canvas**; assignment submissions will be via Canvas
- The course website, MS Teams Channel, and Canvas are all live now

12

Office Hours

- Professor: **Shrideep Pallickara**
  - Office Hours: 1:00 – 2:00 PM Friday (in CSB-364 and via Zoom)
  - Focused exclusively on **course concepts**
- TA: Office hours focused exclusively on **programming assignments**
  - Office Hours: CSB-120 and MS Teams
  - **Graduate Teaching Assistants**: Sofia Catalan, Jason Curcio, and Cameron Seuss
  - **Undergraduate Teaching Assistants**: Joshua Bridgham, Benito Encarnacion, Robert O'Neil

TA Office Hours: Almost finalized

****All changes will be reflected on the course webpage**

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday (Online)
Sofia Catalan	10:00am - 11:00am 1:00pm - 3:00pm	11:00am - 12:00pm 2:00pm - 3:00pm	10:00am - 12:00pm 2:00pm - 3:00pm	10:00am - 12:00pm 2:00pm - 3:00pm		
Jason Curcio	12:30pm - 3:00pm	12:00pm - 1:30pm 4:30pm - 6:00pm	12:00pm - 3:00pm	10:00am - 1:30pm		
Cameron Seuss	10:00am - 11:00am 12:00pm - 1:00pm	12:00pm - 1:00pm 5:00pm - 6:30pm	2:00pm - 3:00pm	5:00pm - 6:30pm	10:00am - 1:00pm 2:00pm - 4:00pm	
Joshua Bridgham		12:30pm - 1:45pm	11:00am - 2:30pm	12:30pm - 1:45pm		10:00am - 1:00pm
Benito Encarnacion		11:00am - 12:00pm 5:30pm - 7:30pm	3:00pm - 4:30pm 5:45pm - 7:15pm	10:30am - 12:00pm	12:00pm - 1:30pm	
Robert O'Neil	2:00pm - 3:00pm	5:00pm - 7:30pm		5:00pm - 7:30pm		12:00pm - 3:00pm

Communications

- Please DO NOT use Canvas messaging for communications
  - Please send communications to **compsci_cs250@colostate.edu**
- The e-mail account is checked by the entire team and allows us to respond to communications in a timely fashion
- Send e-mails from accounts that match your name
  - **No pseudonyms please**
- Do not post code on the MS Teams Channel

Recitations: Will be in the CS Building

- **Recitation Schedule**
  - R2 Wed 10:00-10:50 AM CSB-315
  - R3 Wed 11:00-11:50 AM CSB-315
  - R8 Wed 12:00-12:50 PM CSB-315
  - R4 Wed 1:00-1:50 PM CSB-315
  - R5 Wed 2:00-2:50 PM CSB-315
  - R6 Wed 3:00-3:50 PM CSB-315
- Recitations will be helpful to prepare you for the assignments
- Recitation grading is based on **attendance** and **completion** scores

Course textbook

- This class has three **optional** textbooks
  - The Elements of Computing Systems, second edition: Building a Modern Computer from First Principles. 2nd Edition. Noam Nisan and Shimon Schocken. ISBN-10/ ISBN-13: 0262539802 / 978-0262539807. MIT Press.
  - How Computers Really Work: A Hands-On Guide to the Inner Workings of the Machine. Matthew Justice. ISBN-10/ISBN-13 : 1718500661 / 978-1718500662. No Starch Press.
  - The Secret Life of Programs: Understand Computers - Craft Better Code. Jonathan E. Steinhart. ISBN-10/ ISBN-13 : 1593279701 / 978-1593279707. No Starch Press.

When I make slides …

- I usually refer to, and build them, from several texts
  - And technical papers and articles (with URLs)
- I always list my references at the end of every slide set

Textbooks that I will refer to during the course include … (1/2)

- The Elements of Computing Systems, second edition: Building a Modern Computer from First Principles. 2nd Edition. Noam Nisan and Shimon Schocken. ISBN-10/ ISBN-13: 0262539802 / 978-0262539807. MIT Press.
- How Computers Really Work: A Hands-On Guide to the Inner Workings of the Machine. Matthew Justice. ISBN-10/ISBN-13 : 1718500661 / 978-1718500662. No Starch Press.
- The Secret Life of Programs: Understand Computers -- Craft Better Code. Jonathan E. Steinhart. ISBN-10/ ISBN-13 : 1593279701 / 978-1593279707. No Starch Press.
- Crafting Interpreters. Robert Nystrom. ISBN-10/ ISBN-13 : 0990582930 / 978-0990582939. Genever Benning.

Textbooks that I will refer to during the course include … (2/2)

- Computer Networks: A Systems Approach. Larry Peterson and Bruce Davie. 4th edition. Morgan Kaufmann. ISBN: 978-0-12-370548-8.
- Alex Petrov. Database Internals. ISBN-10/13: 1492040347/978-1492040347 O'Reilly Media.
- Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. Martin Kleppmann. ISBN-10/ ISBN-13 : 1449373321 / 978-1449373320. O'Reilly Media. 2019.
- Shane Cook. CUDA Programming: A Developer's Guide to Parallel Computing with GPUs (Applications of GPU Computing). ISBN-10/ISBN-13: 0124159338/978-0124159334. 1st Edition. Morgan Kaufmann.
- Jeremy Kubica. Data Structures the Fun Way: An Amusing Adventure with Coffee-Filled Examples. No Starch Press. ISBN-10. /13: 1718502605/978-1718502604.
- Computer Organization and Design MIPS Edition: The Hardware/Software Interface. 5th Edition. David A. Patterson and John L. Hennessy. ISBN-10/ ISBN-13 0124077269/ 978-0124077263. Morgan Kaufmann.

On the Course schedule page

http://www.cs.colostate.edu/~cs250/schedule.html

- You will see the **topics** that will be covered and the **order** in which I will cover them
- The readings section will list the books (and the chapters therein) that form the basis for the materials
- You will also see the complete schedule for when the **assignments** are posted and when they are due

Infospaces (**https://infospaces.cs.colostate.edu**)

- A **knowledge repository** that my lab is building to enhance learning
- All videos are designed to be less than 2 minutes
- Improving Infospaces
  - Let us know what you would like to see
  - If you'd like to contribute to this repository let us know!

23

Assignments will target the following elements…

- Number representations and computing
- Memory
- Networking
- Storage systems

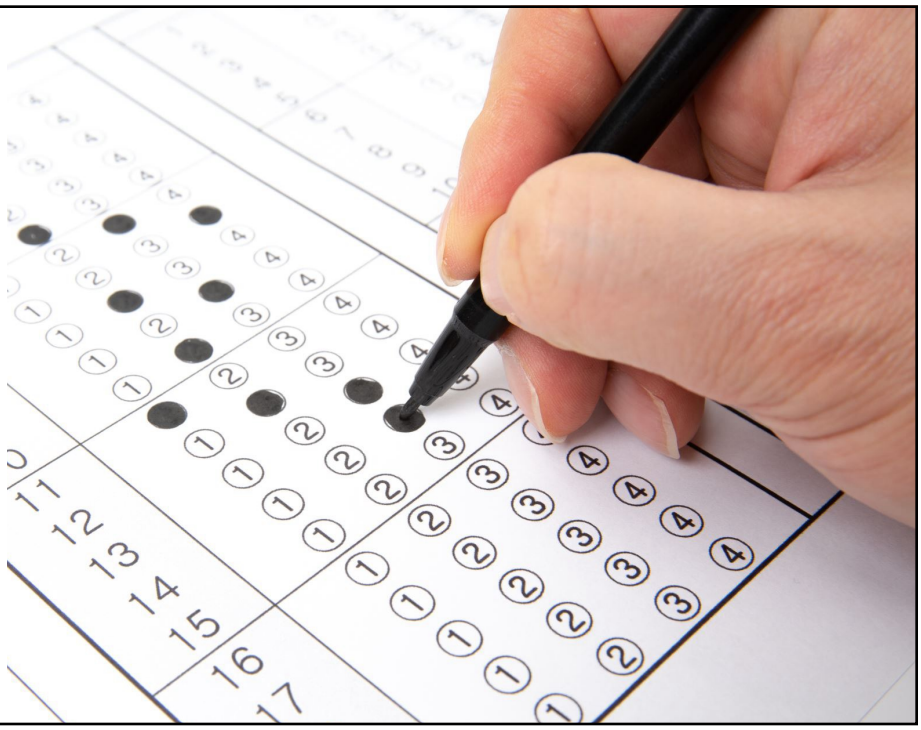
*** You will be developing complete programs individually and packaging your software for grading.*

24

Assignments: Logistics

- Assignments will be due **at 8:00 pm MT on Wednesdays**
- You are allowed to submit up to 2 days late
  - There is a **7.5%** deduction for each day that you are late
- All assignments are **individual** assignments
- Assignments should be submitted via **Canvas**

GRADING



Grading breakdown

- Assignments {4}: 15%
- Coding Exams {3}: 15%
- Recitations {attendance + completion}: 10%
- Quizzes (10 best) : 10%
- Mid Term I: 15%
- Mid Term II: 15%
- Comprehensive final exam: 20%
- Extra-credit (optional) opportunities
  - Programming exercises (2%)
  - .

Written Exams		
Midterm-I	2/24/26	3:30 PM
Midterm-II	4/7/26	3:30 PM
Final Exam	5/14/26	2:00-4:00 pm

There are 3 coding exams

- Each coding exam will have 2 problems
  - One worth 2 points, and the other worth 3 points
- February 17th 5:00 - 7:50 PM
- March 10th 5:00 - 7:50 PM
- April 14th 5:00 - 7:50 PM
- Coding exams are the norm for the vast majority of the internships/jobs
  - The coding exams in CS250 will help you hone that skill

To allow for some room to trip-up (not that you should plan to!)

- Each coding exam will be curved so that the average score is 80%
  - Only students who have received a minimum of 40% (i.e., 2/5)will be considered for (a) calculation of the average, and (2) receiving the extra points (if any are awarded, to preserve the 80% average)
- If you receive a 40% in each of the assignments; we will drop one assignment where you had your lowest score
  - That is, if you score at least a 40% (i.e., 2/5) in each of the 4 assignments, your top-3 assignments will account for programming component of the course grade

Assignment Schedule

	Released	**Due Date:** Multiple deliverables in case of HW3				
Exercises [Extra Credit]	21-Jan		Jan 28	Feb 4	Feb 11	Feb 18
HW1	28-Jan	11-Feb				
HW2	13-Feb	4-Mar				
HW3	5-Mar		25-Mar	1-Apr	8-Apr	
HW4	8-Apr	6-May				

Grading Policy [1/3]

- Letter grades will be based on the following standard breakpoints:
  - >= 90 is an A, >= 88 is an A-,
  - >= 86 is a B+, >=80 is a B, >=78 is a B-,
  - >= 76 is a C+, >=70 is a C,
  - >= 60 is a D, and <60 is an F
- I will not cut higher than this, but I *may* cut lower

Grading Policy [2/3]

- All exams (quizzes, midterms, and Final) are in-class, in-person
- If you have **SDC accommodations** the advisor will send me a letter outlining them

Grading Policy [3/3]

- Every assignment will be posted at least 2 weeks before the due date.
- Every assignment will include information about:
  - How much it will count towards the course grade
  - How it will be graded
- Late submission penalty: 7.5% per-day for the first 2 days
  - Submissions after the late submission period will have an automatic ZERO
  - Detailed submission instructions posted on the course website.
  - Assignments will be **graded within 30-60 seconds of submission**

Plagiarism detection

- Use of generative AI (GitHub co-pilot, ChatGPT, Maude and their ilk) is expressly disallowed at stage (include ideation) for programming assignments
  - Will be considered plagiarism and cheating
- All programming assignments will be subject to pair-wise comparisons
  - Collusions, copying from the same source on the internet, and using (un)paid-versions of GenAI for solutions will all be detected

Quizzes, mid term, and final

- I will only ask questions about what I teach
  - If I didn't teach it, I won't ask from that portion
- If the concepts were covered in my slides
  - You should be be able to answer the questions

Quizzes

- There will be 12-13 quizzes
  - We will take your 10 highest scores
  - If you miss class for some reason other than (family/medical) emergencies and university sanctioned reasons, you do not need to let me know and there will be NO makeup for quizzes (please don't ask to do this!)
    - University mandates that these require official documentation
    - Job/telephone interviews, waking up late, car-troubles, etc. do not constitute a university approved reason

Emergencies

- We will account for personal/family/medical emergencies
  - Please write!
- University regulations require that students submit documentation
  - We will need documentation to make accommodations

Late arrivals and quizzes

- Tardiness is not professional behavior
- 20% deduction on the quiz score for every 15-minute increment that you are late
  - Grace period of 15 minutes
  - E.g., 30 < tardy < 45 → 40% deduction
  - There will be no negative scores … scores will bottom out at zero

Expectations

- You will attend all classes
- You will focus on the discussions, and not on …
  - Other assignments
  - Social networking updates
- Assignments have to be done **individually**

WHAT IT TAKES TO SUCCEED

What it takes to succeed [1/3]

- You are required to work at least **9-10 hours** per-week outside of class
  - Coding and reviewing material from class
- If you miss a lecture
  - Add about 3 hours per missed lecture

What it takes to succeed [2/3]

- Work on the assignments **every day**
  - There is no such thing as waiting for inspiration to strike!
- **Reflect** about how you could have done things differently for better performance
  - Even after you have submitted an assignment
  - It will improve the choices you make in the next assignment

What it takes to succeed [3/3]

- Work in bigger-sized chunks
  - Too many short bursts = Too many context switches
    - You will be busy doing nothing
- Document your code

How to fail this course?

- Believing that you can learn via osmosis
- **Missing lectures**
  - If you don't have the discipline to show up, you will most likely not have the discipline to catch up
- **Procrastinating**
  - The assignments cannot be done in a week
  - *Organize your schedule* so that you can succeed
- Not attacking the problem and working on the fringes
  - Spend your time wisely on critical paths

You are not allowed to take away learning opportunities from other students

- You can **use a laptop ONLY IF you have a University accommodation** that expressly requests to allow you to use a laptop
  - You will have to sit in the last row starting at the corners
  - Turn off wireless and use it only for taking notes
  - Students not using a laptop should not sit in the last row starting at the corners
- Tablet use with a stylus/pencil is allowed
  - Tablet use in keyboard mode is not allowed
- When the class is in session, put away your cell-phones!
- Also, please no cross talking when the class is in-session

Why attend lectures if all the slides are posted?

- Slides are only part of the story
  - They anchor the discussion
- Any field has a *language* associated with it
- People who have worked in an area for a long time speak the language
  - Sitting in classes helps you learn how to frame questions and responses
- Often there are surprising questions
  - Some of these may be asked by interviewers

Help me help you

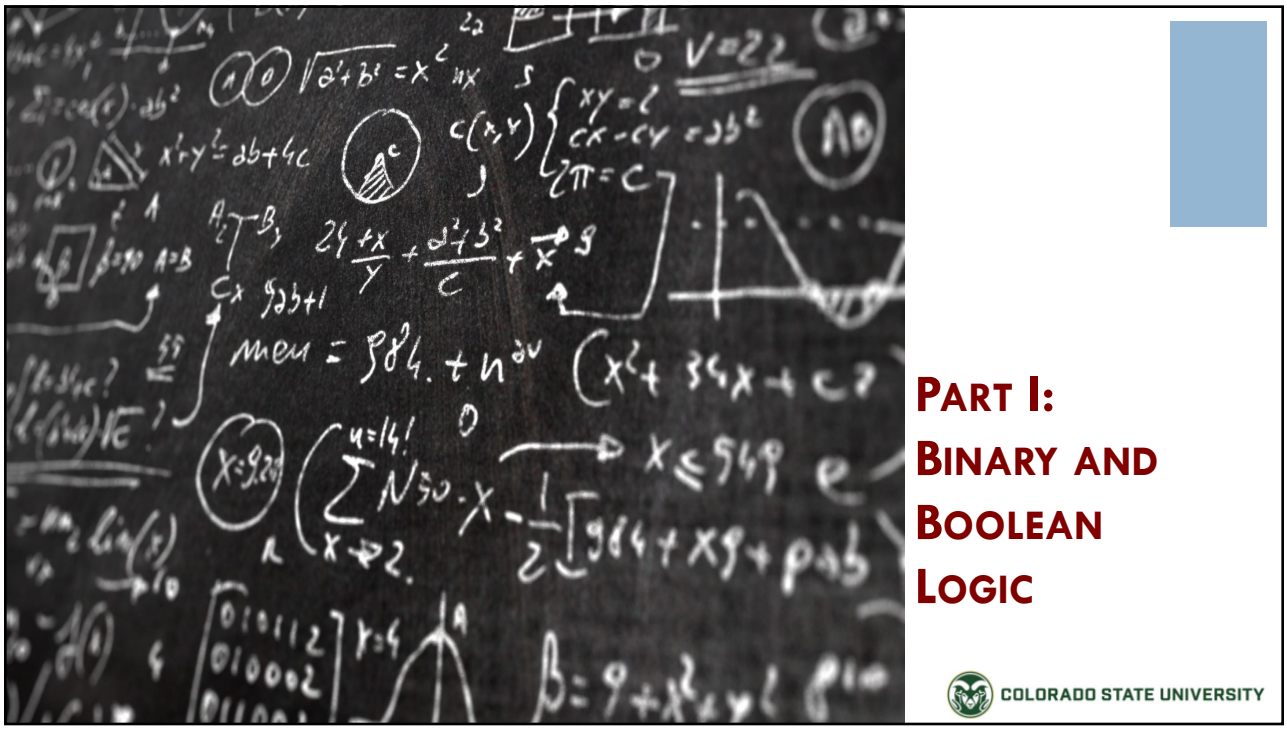
- We will have **surveys** at the end of every class
- You will provide a list of
  - 3 concepts you followed clearly
  - 3 concepts you had problems keeping up with
- Problem areas for the majority of the class will be addressed in the next class

Interactions

- You can have discussions with me, the TAs, and your peers
- There are **two constraints** to these discussions
  1. No code can be exchanged under *any* circumstances
  2. No one takes over someone else's keyboard
- Bumps are to be expected along the way
  - But you should get over this yourself
  - It will help you with the next problem you encounter


Fairy tales are more than true: not because they tell us that dragons exist,
but because they tell us that dragons can be beaten.
G.K. Chesterton by way of Neil Gaiman, Coraline
TOPICS COVERED IN CS250
L1.49

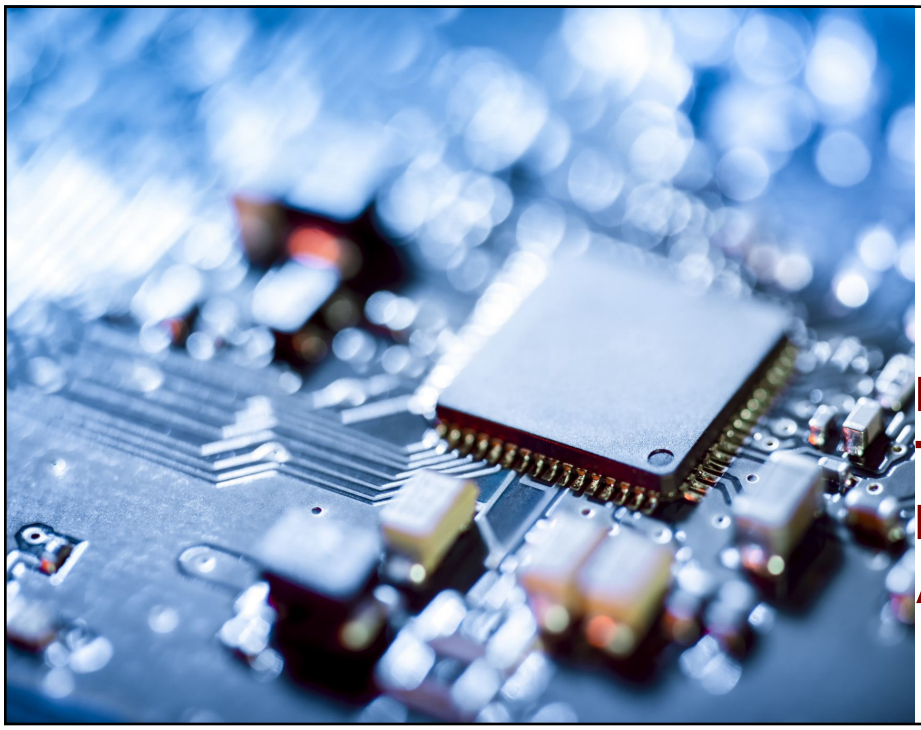
49


PART I:
BINARY AND
BOOLEAN
LOGIC
COLORADO STATE UNIVERSITY

50

Binary and Boolean Logic

- Number representations
  - Boolean Algebra, Boolean Logic
- Gates
- Signed numbers and floating-point representations
  - Two's and One's complement number representations
- Synthesize Boolean functions from Truth Tables
- Prove how all Boolean functions can be constructed using only NAND gates



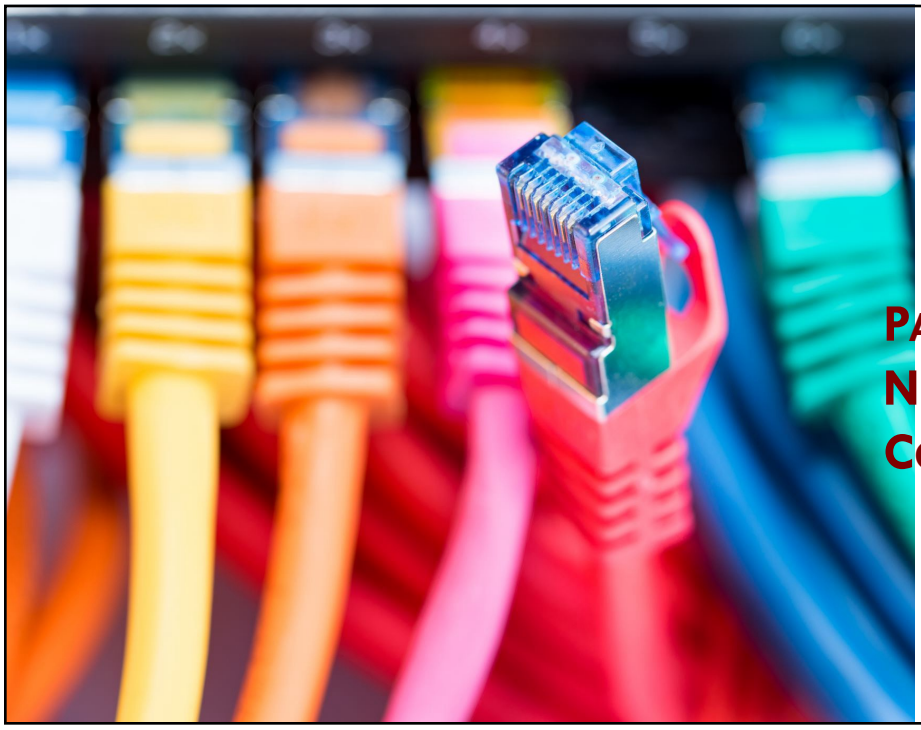
PART II: THE VON NEUMANN ARCHITECTURE

The von Neumann Architecture

- Processors, cores and hyperthreading
  - Mapping threads to execution pipelines
- Memory hierarchy and its impact on performance
  - Cache organization: L1, L2 and L3 caches
  - Associative memory and direct-mapped caches
  - Main Memory: CPU RAM (addressing and organization)
- Why miniaturization works?

CPU meet the GPU (TPU, etc.)

- GPU: Design, operations, and concurrency
- The right tool for the right job
  - How the GPU waxed while the CPU waned
  - GPU Limitations: general purpose computations and memory management



PART III
NETWORKED COMPUTER SYSTEMS

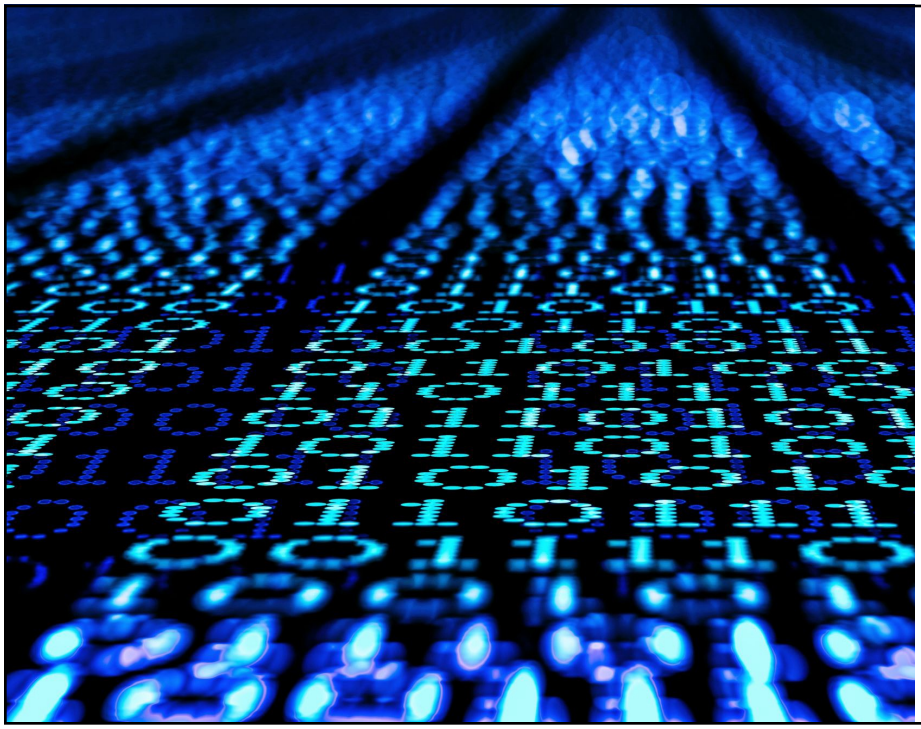
COLORADO STATE UNIVERSITY

55

Networking

- IPv4 and IPv6
  - Encapsulation, packet header formats, fragmentation, and extension headers
- TCP
  - Sliding window, buffering, reliable and ordered delivery
- UDP
- DNS, private IP addresses, and NAT

56



PART IV: STORAGE SYSTEMS

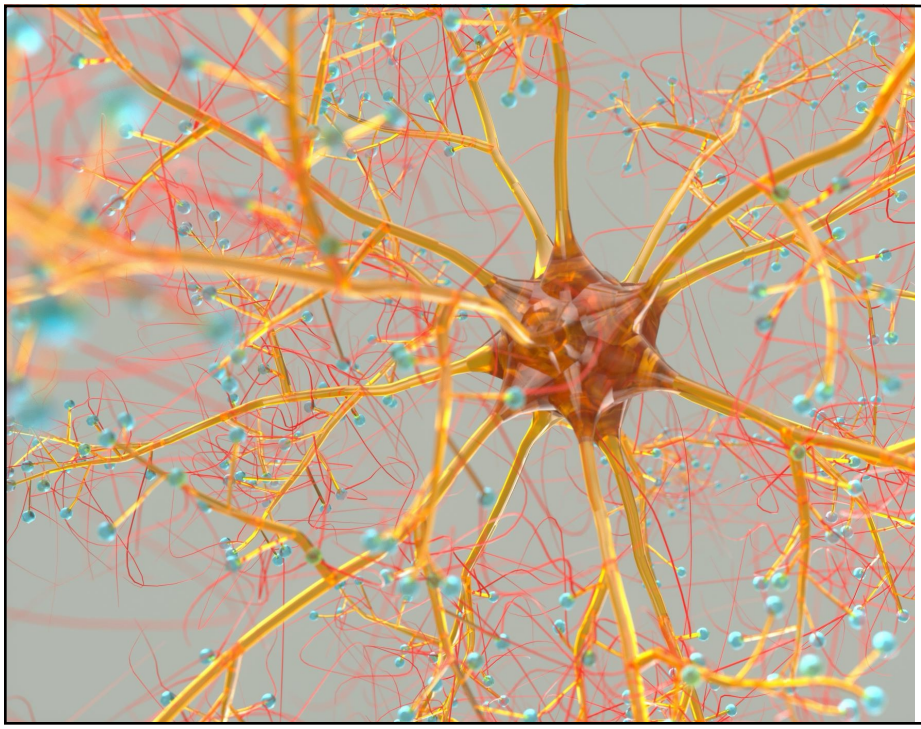
COLORADO STATE UNIVERSITY

57

Storage Systems

- Pitfalls of using in-memory data structures for storage systems
- Binary search, dynamic data structures, BSTs
- B-Trees (and variants)

58



PART V: FUTURE COMPUTING SYSTEMS

COLORADO STATE UNIVERSITY

59

Future Computing Systems

- The von Neumann bottleneck
- Where to from here?

60

61

Algorithm

- **Sequence of steps** to accomplish desired objectives
  - E.g., sorting numbers, factorizing numbers, processing graphs, etc.
- Humans can carry out algorithms, but we are much slower than machines
  - Modern computers are a trillion times faster!

62

Computations and computers

- Computations are complex series of numerical calculations
- Computers are agents that carry out operations within a computation

63

COMPUTATIONS & COMPLEXITY

64

Computation complexity

- Measured in terms of time, space, and (increasingly energy)
- Impacts responsiveness and how you can scale
- Other implications
  - Cryptocurrency: Systems like Bitcoin, where the currency represents a solution to a problem and reflects the *amount of computational work* that needs to (and has been) performed

Perils of wishful thinking [Part I]

- Most common wish is that we can get computers to do **any** job we can conceive of
- Many jobs are impossible for computers
  - No algorithm that will inspect another algorithm and tell us whether it terminates or loops forever
    - It was logically impossible in 1936 when Alan Turing proved it so, and it is still impossible today

A problem for you to solve

- UPS truck driver must deliver packages to 10 cities
  - Wants to travel the **least** distance, but visit each city *exactly once*

67

Perils of wishful thinking [Part II]

- Even if we stick to logically possible jobs, there are many that cannot be done in a reasonable time — they are **intractable**
- Example: Shortest tour of a set of cities that visits each city just once
  - The simplest way to find the shortest tour is to enumerate all possible tours and select the shortest

68

What if there were 50 cities?

- How many paths would you need to enumerate?
- 50!
- Which is?
  - 30,414,093,201,713,378,043,612,608,166,064,768,844,377,641,568, 960,512,000,000,000,000

Intractability and Heuristics

- For a small set of 100 cities, this would take 10^{130} years on the world's fastest supercomputer
  - The age of the universe is 10^{10} years
  - Even the "simplest way" can be impossible!
- The picture gets even more confusing because in most cases there are fast algorithms to find an **approximate** answer
  - They are called **heuristics**

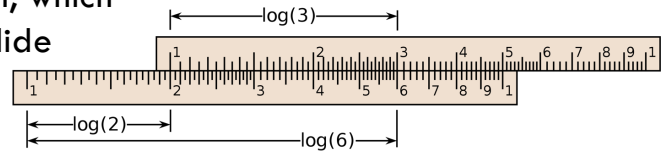
Wishful thinking is also when you believe the computer is smart

- If you are not precise in translating the algorithm into program steps, your computation will contain errors
- The computer amplifies your intelligence but has none of its own!

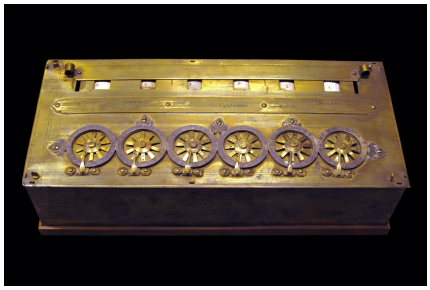
A (VERY) BRIEF HISTORY OF COMPUTERS

A Brief History of Computers: Part I

- Napier invented the logarithm, which became the principle of the slide rule (invented circa 1620)
  - Could not add or subtract

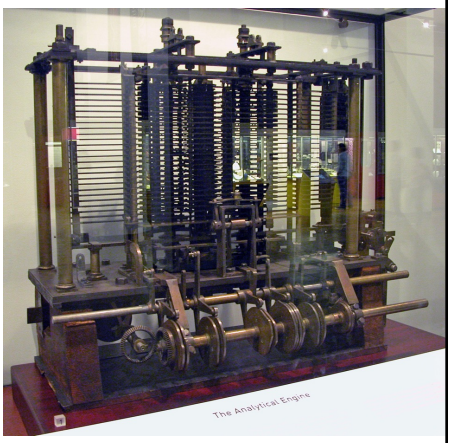


- Blaise Pascal designed and built an arithmetic machine in 1642 to add and subtract



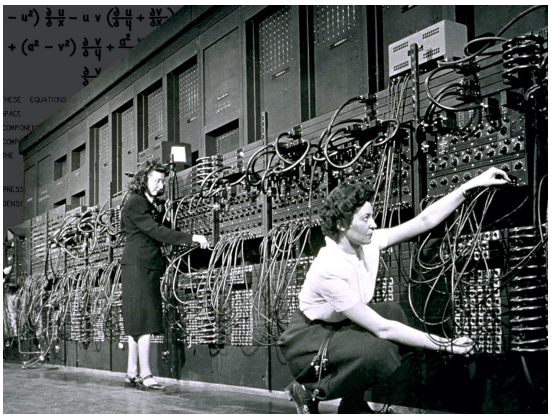
A Brief History of Computers: Part II

- In 1819, Babbage designed a machine of gears, shafts, and wheels that could calculate tables of arithmetic numbers such as logarithms
- 1890 US census, Hollerith's punched card machines tabulated large amounts of data
  - Jacquard's loom (1801) was the first place where punched cards were used



Modern computer systems design

- Has it origins dating back to1947 as part of work on ENIAC
- Breaks up a computing machine into three main subsystems:
  - The central processing unit (**CPU**) for performing arithmetic operations
  - The **memory** for storage of instructions and data
  - The input-output (**I/O**) unit for communicating with the external world
- This way of organizing a machine became known as the "**von Neumann architecture**"



Modern computers: The secret sauce?

- At its core modern computers harnesses the **movement of electrons** in circuits to carry out computations
- Computer circuits deal only with voltages, currents, switches, and malleable materials
  - Internally the computer does not process numbers and symbols

No information without representation

- To be processable, data must be represented as
  - Signals in the machine or
  - As measurable disturbances in the structure of storage media

77

The contents of this slide-set are based on the following references

- Noam Nisan and Shimon Schocken. *The Elements of Computing Systems: Building a Modern Computer from First Principles*. 2nd Edition. ISBN-10/ ISBN-13: 0262539802 / 978-0262539807. MIT Press. [Preface]
- Peter J. Denning and Matti Tedre. *Computational Thinking*. Essential Knowledge series. The MIT Press. ISBN-10:ISBN-13 0262536560/ 978-0262536561. 2019. [Chapter 2]

78