

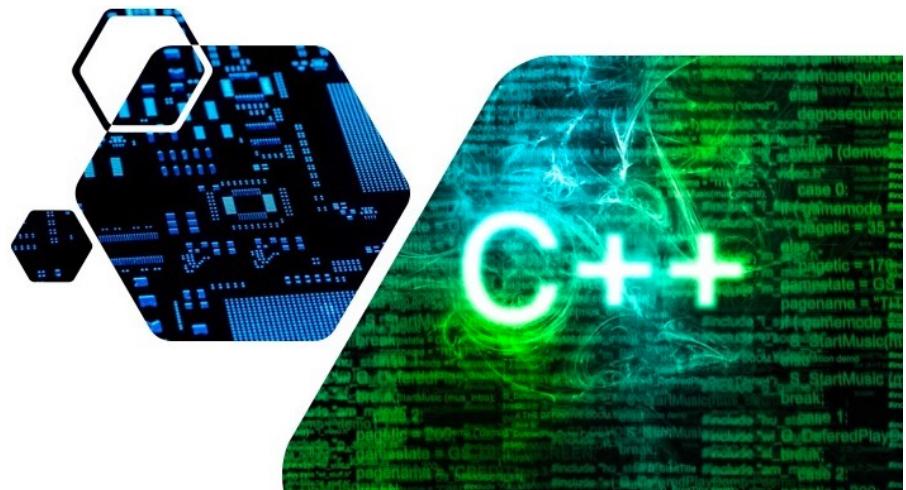
# Lecture 1: Introduction to Programming

Dr. Tsung-Wei Huang

Department of Electrical and Computer Engineering  
University of Utah, Salt Lake City, UT



*The course teaches you how to write basic computer programs with a specific focus on the C++ programming language.*



# Class Logistics

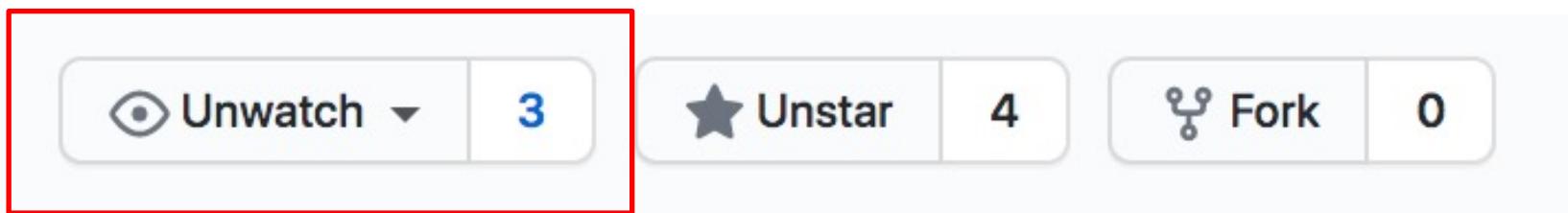
---

- Staff**
  - Instructor: Dr. Tsung-Wei Huang ([tsung-wei.huang@utah.edu](mailto:tsung-wei.huang@utah.edu))
- Main class**
  - 9:00 – 11:50 AM Friday (excluding holiday)
  - Zoom: <https://utah.zoom.us/j/2468214418>
- Lab is right after class**
  - Friday: 12:00– 12:50
- Office hour**
  - By appointment
- Web:** <https://github.com/tsung-wei-huang/cs1420>

# Watch the GitHub Repository

---

- <https://github.com/tsung-wei-huang/cs1420>
- You can receive all updates
  - New homework
  - New check-in data
  - New updates
  - ... (please register for a GitHub account)



- Canvas will be auxiliary

# Scoring

---

- Total 100 points**
  - Weekly lab programming assignments (PA): 60%
  - Take-home exams: 40%
- Academic integrity**
  - We trust you but don't take it for granted
  - Violation will be recorded in your transcript
  - <https://regulations.utah.edu/academics/6-400.php>
- Textbook**
  - No need to buy any textbook

# Learning Materials

- You **DON'T NEED** to buy any textbook
  - Slides will be available and are enough
  - Google can help you find literally all answers
  - I never use any textbook in learning C++
  - Programming requires “practice” not just reading
- Useful C++ reference
  - Cpp reference: <https://en.cppreference.com/w/>
  - CppCon: <https://www.youtube.com/user/CppCon>

C++ reference			
Compiler support	Freestanding implementations	Concepts library (C++20)	Iterators library
Language	Standard library	Diagnostics library	Ranges library (C++20)
Predefined concepts	General utilities library	General utilities library	Algorithms library
Keywords	String library	String library	Numerics library
Parsers	Date and time	Common math functions	Common math functions
Expressions	Filesystem library	Numeric special functions (C++17)	Numeric special functions (C++17)
Declarative	Localization library	Numeric algorithms	Periodic random number generation
Initialization	String conversion (C++11)	Random number generation	Fast random number generation (C++11)
Finalizers	Utility functions (C++11)	Complex $\times$ valarray	Complex $\times$ valarray
Statements	optional (C++17) = any (C++17)	Streams-based I/O	Streams-based I/O
Closures	variant (C++17) = forward (C++20)	I/O manipulators	I/O manipulators
Templates	String library	Input output library	Input output library
Exception	basic_string_view (C++17)	Stream-based I/O	Stream-based I/O
Headers	Nullpointer (C++11)	I/O manipulators	I/O manipulators
Named requirements	basic_string (C++17)	Regular expressions library (C++11)	Regular expressions library (C++11)
Program test macros	multibyte = wide	basic_regex = algorithms	basic_regex = algorithms
Language support library	Containers library	Atomic operations library (C++11)	Atomic operations library (C++11)
Type support – traits (C++11)	map = unordered_map (C++11)	atomic = atomic_flag	atomic = atomic_flag
Resource management	pair = unordered_pair (C++11)	atomic_flag = atomic_flag	atomic_flag = atomic_flag
Relational comparators (C++20)	Other containers	Sequence associative	Sequence associative
Memory management	sequence = associative	unordered associative = adaptors	unordered associative = adaptors
Initializer_list (C++11)	Thread support library (C++11)	Thread support library (C++11)	Filesystem library (C++17)
Technical specifications			
Standard library extensions	Standard library extensions	Library fundamentals TS1	
Concurrency library extensions	Concurrency library extensions	Library fundamentals TS2	
Standard library extensions v2	Standard library extensions v2	Library fundamentals TS2	
Transactional memory	Transactional memory	Library fundamentals TS3	
Standard library extensions v3	Standard library extensions v3	Library fundamentals TS3	
Concurrency library extensions	Concurrency library extensions	Library fundamentals TS4	
Concepts	Concepts (concepts TS)	Library fundamentals TS5	
Ranges	Ranges (ranges TS)		
Transactional Memory (TM TS)	Transactional Memory (TM TS)		



# Class Philosophy

---

- ❑ **Focus on software programming**
  - ❑ With applications on ECE problems
  - ❑ With focus on the C++ programming language
- ❑ **NOT to trouble you with**
  - ❑ Difficult homework
  - ❑ Tricky exams
  - ❑ Unreasonable learning and scoring curve
- ❑ **At the end of the class, I want you to**
  - ❑ Understand important ECE programming problems
  - ❑ Improve your coding skills and algorithm knowledge
  - ❑ Have more job opportunities in software companies

# To this end ...

---

- **Each class is organized as two parts**
  - Lecture
  - Programming practice
- **We will start from very basic stuff and then move up**
  - Important
  - Basic data structure and algorithms
  - C++ coding techniques and best practice
- **Solve real-world computer design problems in C++**
  - Program on microcontroller
  - Program circuit design problems
  - ...

# **LAB is Right After Class**

---

- We have one LAB section**
  - 12:00-12:50, Friday
- LAB gives you hands-on programming experience**
  - Lab assignment counts 60% of your score

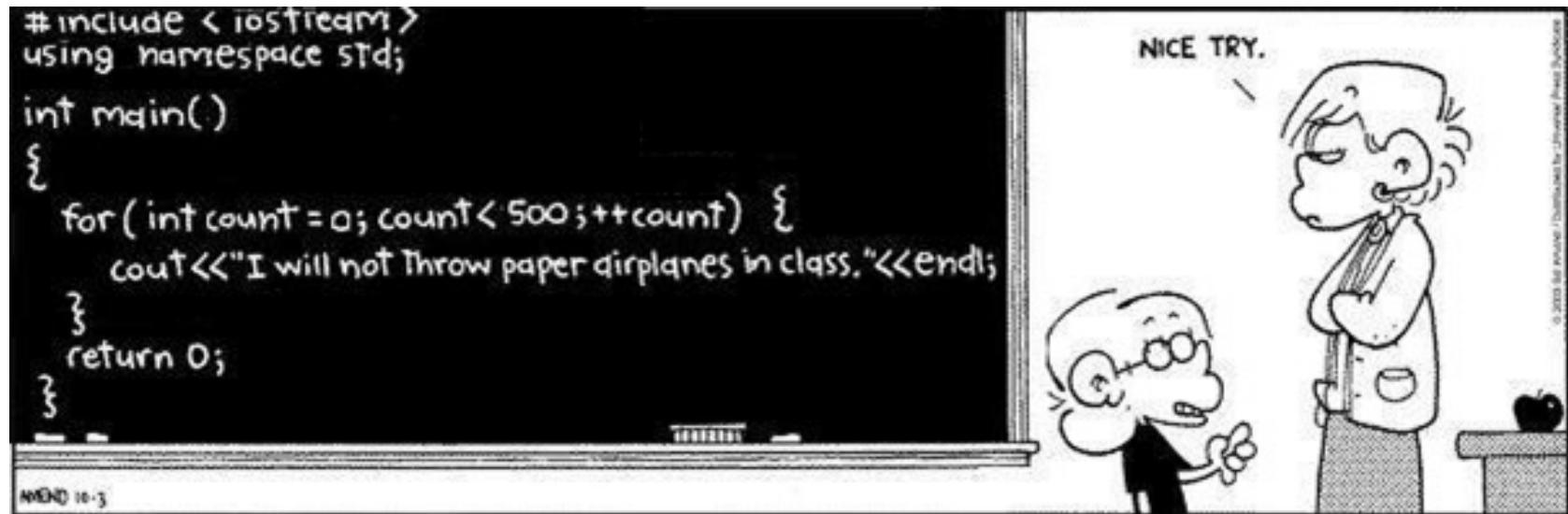
# Utah CADE Machines

---

- ❑ All of you should have access to CADE servers
  - ❑ If not, go apply for one at: <https://www.cade.utah.edu/>
- ❑ CADE machines have all the facilities we need
  - ❑ Domain: labX-Y.eng.utah.edu
    - X = 1, 2, 3, ... (# of lab space)
    - Y = 1, 2, 3, 4, 5, ... (machine # of each lab space)
    - For example, lab2-20.eng.utah.edu
  - ❑ Account: your uid
  - ❑ Password: your uid login password
- ❑ Remote login using ssh (the easiest way)
  - ❑ ssh -x u6024634@lab2-20.eng.utah.edu

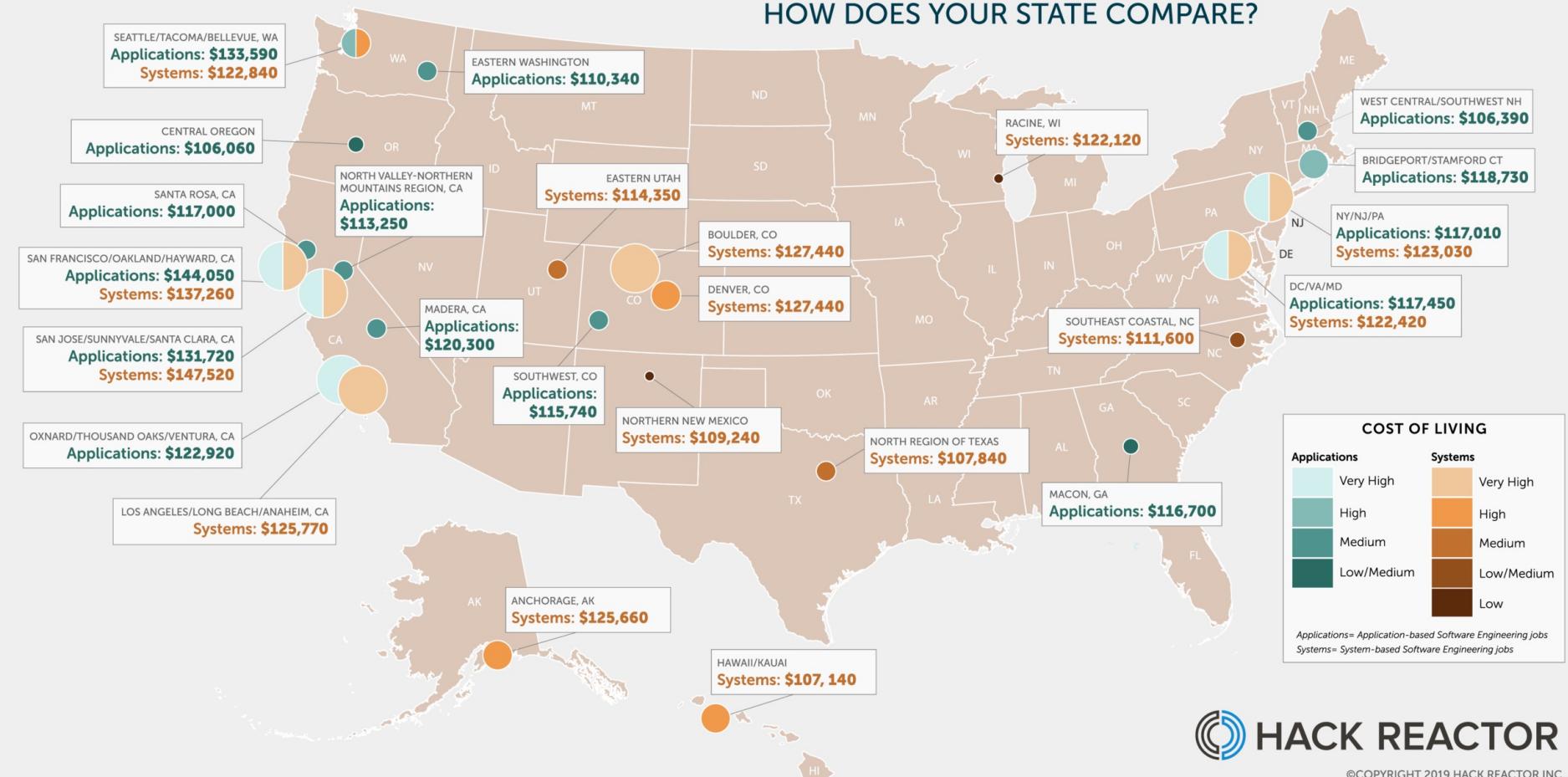
# Online Editor (Web-based Interface)

- Online C++ editor: <https://www.onlinegdb.com/>

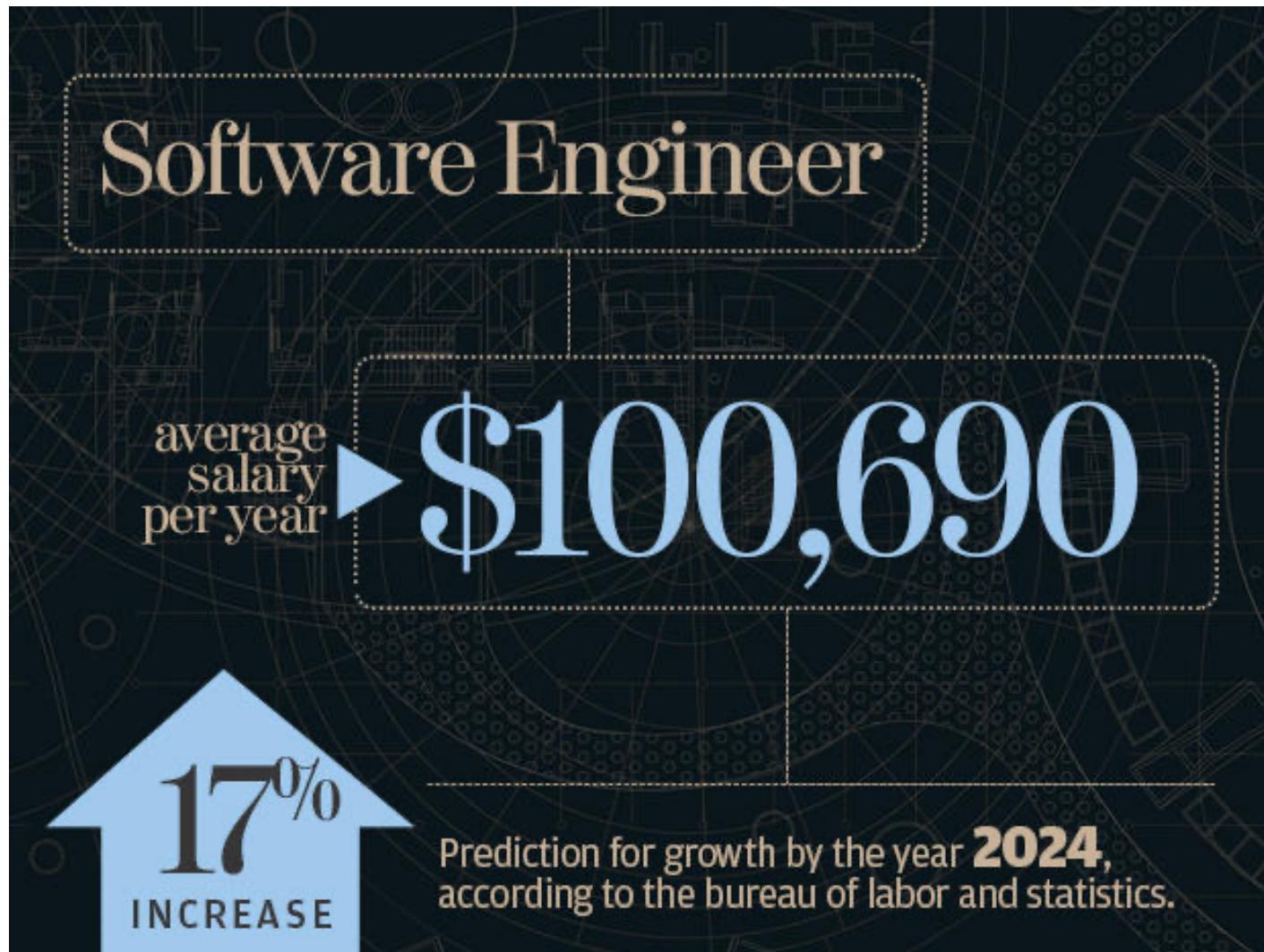


# Software Engineer Job Landscape

## SOFTWARE ENGINEER SALARY REVIEW 2019: HOW DOES YOUR STATE COMPARE?



# Software Engineer Jobs are Growing!



# Silicon Slope is right here in Utah!!!

**A rainbow sampling of companies that make Utah Valley the #1 place for business and careers according to Forbes magazine.**



**"Topping our 14th annual list of The Best Places for Business and Careers is Provo, Utah. The \$16 BILLION ECONOMY is thriving largely on the back of BRIGHAM YOUNG UNIVERSITY ... Job growth was a robust 3% in the Provo metro in 2011, THIRD BEST IN THE U.S. It enjoys – by far – the LOWEST VIOLENT CRIME RATE IN THE U.S. and ranks tenth lowest overall for crime, including property crime. The METRO POPULATION has doubled over the past two decades to 542,700."**

**– Forbes magazine**

# We live in a highly connected world ...

Transmission and Interface

***Connecting***



Screen Phone



Mobile Phone

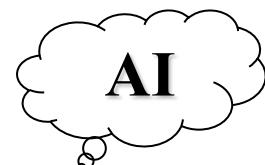
Smart Mobile Phone



PDA



Palm PC



AI



Set Top Box



Auto PC



Handheld PC

***Computing***

Thin Client



MP3



Data Processing

***Consumer***



Handheld Game



DSC

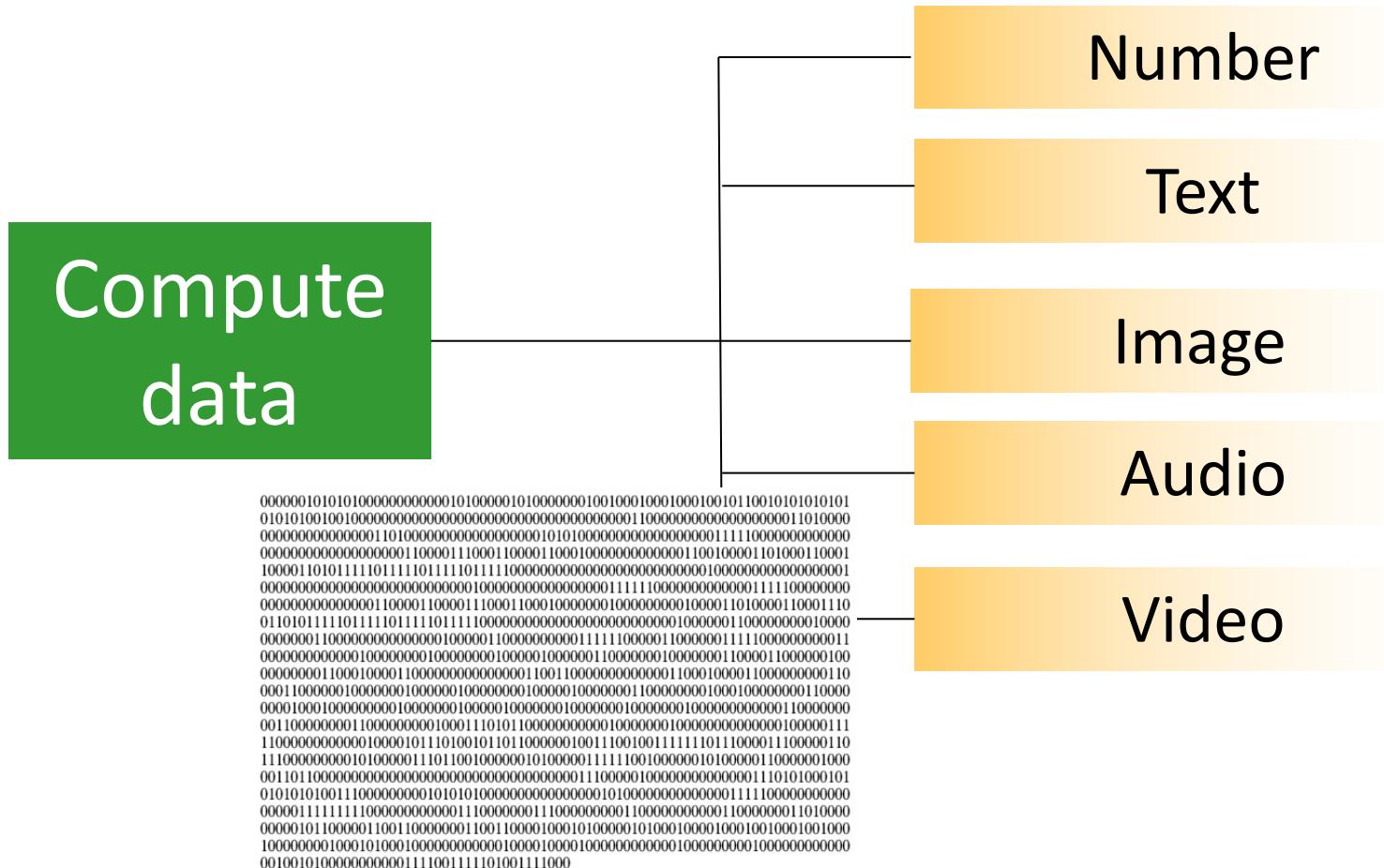


Net TV



Net TV

# Computing Plays a Key Role



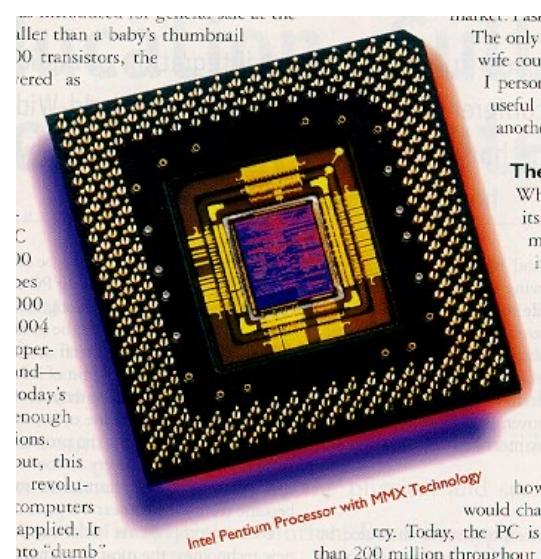
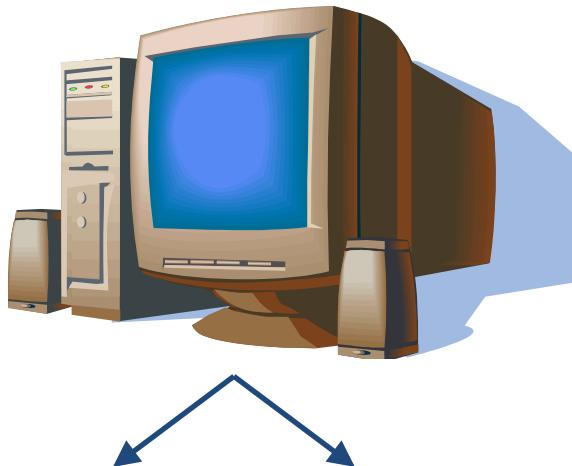
Computer see everything in binary (0 and 1)

# Hardware vs Software

The focus  
of this  
course

```
main ()  
{  
    say hello;  
    go to internet;  
    goodbye;  
}
```

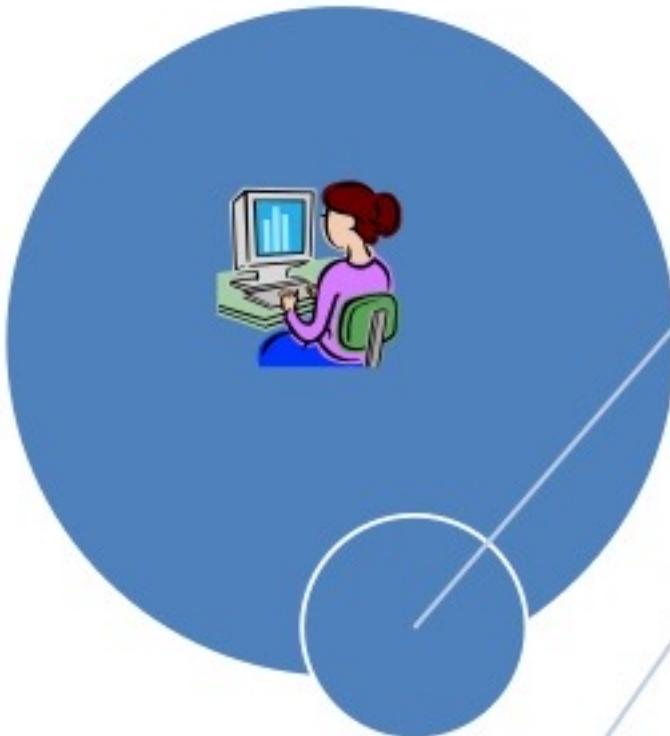
# Software



# Hardware

# What is Software?

---



**Software is a general term for the various kinds of programs used to operate computers and related devices. (The term hardware describes the physical aspects of computers and related devices.)**

# How do We Talk to Computers?

- Like we talk to human through “language”
  - English, Mandarin, Hindi, ... => depending on countries
  - C++, PHP, Python, JS, ... => depending on applications



# Top 10 Programming Languages

Rank	Language	Type	Score
1	Python	🌐💻⚙️	100.0
2	Java	🌐📱💻	96.3
3	C	📱💻⚙️	94.4
4	C++	📱💻⚙️	87.5
5	R	💻	81.5
6	JavaScript	🌐	79.4
7	C#	🌐📱💻⚙️	74.5
8	Matlab	💻	70.6
9	Swift	📱💻	69.1
10	Go	🌐💻	68.0

1	Java	
11	MATLAB	
2	c	
12	R	
3	Python	
13	Perl	
4	C++	
14	Assembly Language	
5	Visual Basic .NET	
15	Swift	
6	Javascript	
16	Go	
7	C#	
17	Delphi/Object Pascal	
8	PHP	
18	Ruby	
9	SQL	
19	PL/SQL	
10	Objective-C	
20	Visual Basic	

# An Often Asked Question

---



# From an ECE undergraduate's perspective

---

- It doesn't matter
  - Unless you just want to get hacky programming experience and find a job quickly ...

**Learning “how computers think” and  
“how you talk to computers through  
one language” is more important!**

# Let's Do an Example

---

- Find a power of a number
  - Input: a, b ( $1 < a, b < 2147483647$ )
  - Output:  $x = a^b$ 
    - $a=3, b=4, x=3^4=81$
    - $a=2, b=5, x=2^5=32$
  - Assume you can only do multiplication one at a time
- Naïve method
  - $2^{16} = 2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2$  total 15 calculations
- Can we do better?

# Divide and Conquer

---

- Naïve method

- $2^{16} = 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2$  total 15 calculations

- A better way as follows:

$$2^{16} = 2^8 * 2^8$$

$$2^8 = 2^4 * 2^4$$

$$2^4 = 2^2 * 2^2$$

$$2^2 = 2 * 2 \quad \text{We need only 4 calculations!!!}$$

# How Efficient is it to Compute $a^b$ ?

---

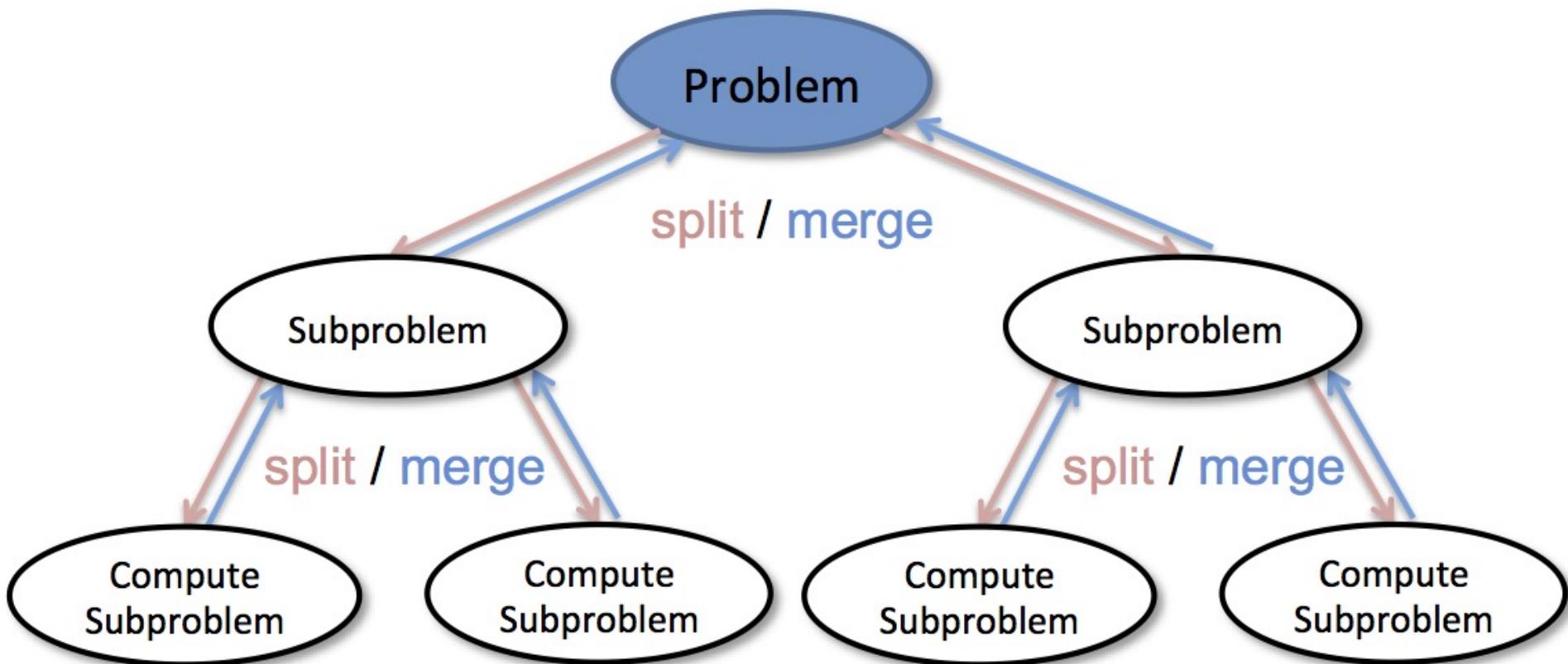
- Naïve method
  - # calculations: linear to b
- Divide and Conquer
  - # calculations:  $\log_2(b)$
- Let's say  $n = 2147483648$ 
  - Naïve method takes **2147483647** calculations ( $\sim 10\text{-}30\text{s}$ )
  - Divide and Conquer takes only **31** calculations ( $\sim 1\mu\text{s}$ )
    - 10000000x faster!
  - Indeed, this is a Goo\_\_ interview question

*In fact, there is a name for the refined method:*

*Divide and Conquer algorithm*

*(used to solve >50% computer science problems,  
including Google's distributed system solutions,  
e.g., MapReduce)*

**Divide and Conquer is the most basic building block to solve almost all difficult computer problems**



# Targeted Programming Language

---

- ❑ **We will be using C++ most of the time**
  - ❑ Most large-scale problems are written in C/C++
  - ❑ Most AI backend engines are written in C/C++
  - ❑ Most performance-critical blocks are written in C/C++
- ❑ **C++ is advantageous in**
  - ❑ Being both high-level and low-level
  - ❑ Linking to hardware and computer architecture
  - ❑ Understanding memory hierarchy and cache effect
- ❑ **We will talk a bit about Java at the end of the course**

# Once You Master C++ ...

---

C++ and JAVA developer: learning  
PYTHON



PYTHON developer: learning JAVA and  
C++



# Summary

---

- Class Logistics
- Software engineering
- Programming languages