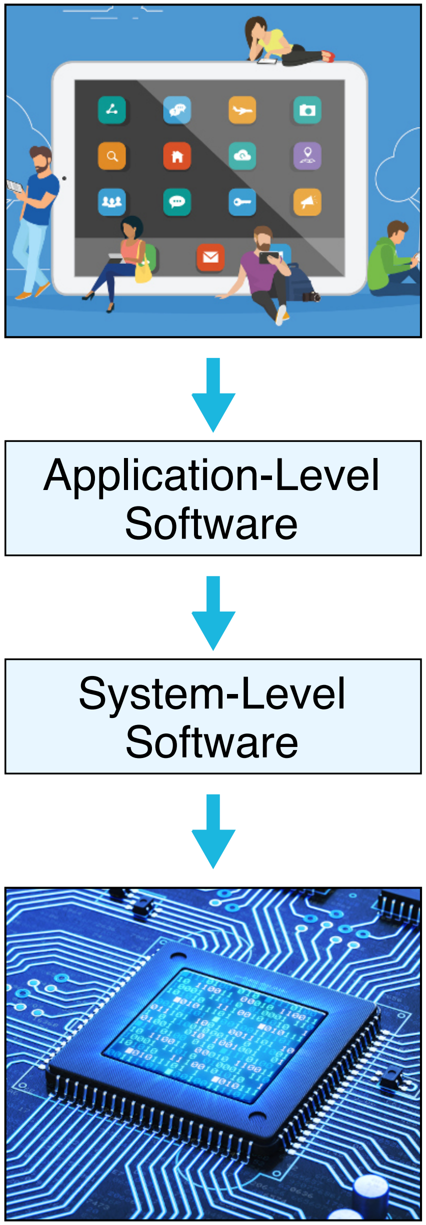
ECE 2400 / ENGRD 2140
Computer Systems Programming
Course Overview

Christopher Batten

School of Electrical and Computer Engineering
Cornell University

`http://www.csl.cornell.edu/courses/ece2400`



Application-Level Software

System-Level Software

ECE 2400 / ENGRD 2140 Computer Systems Programming

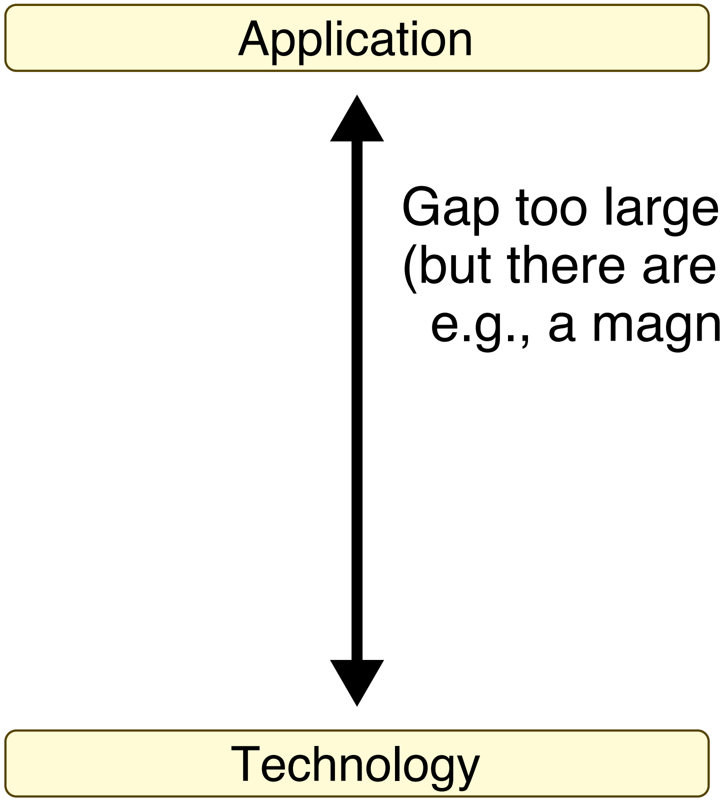
What is Computer Systems Programming?

Activity: Comparing Algorithms

Trends in Computer Systems Programming

Course Logistics

Applications vs. Technology

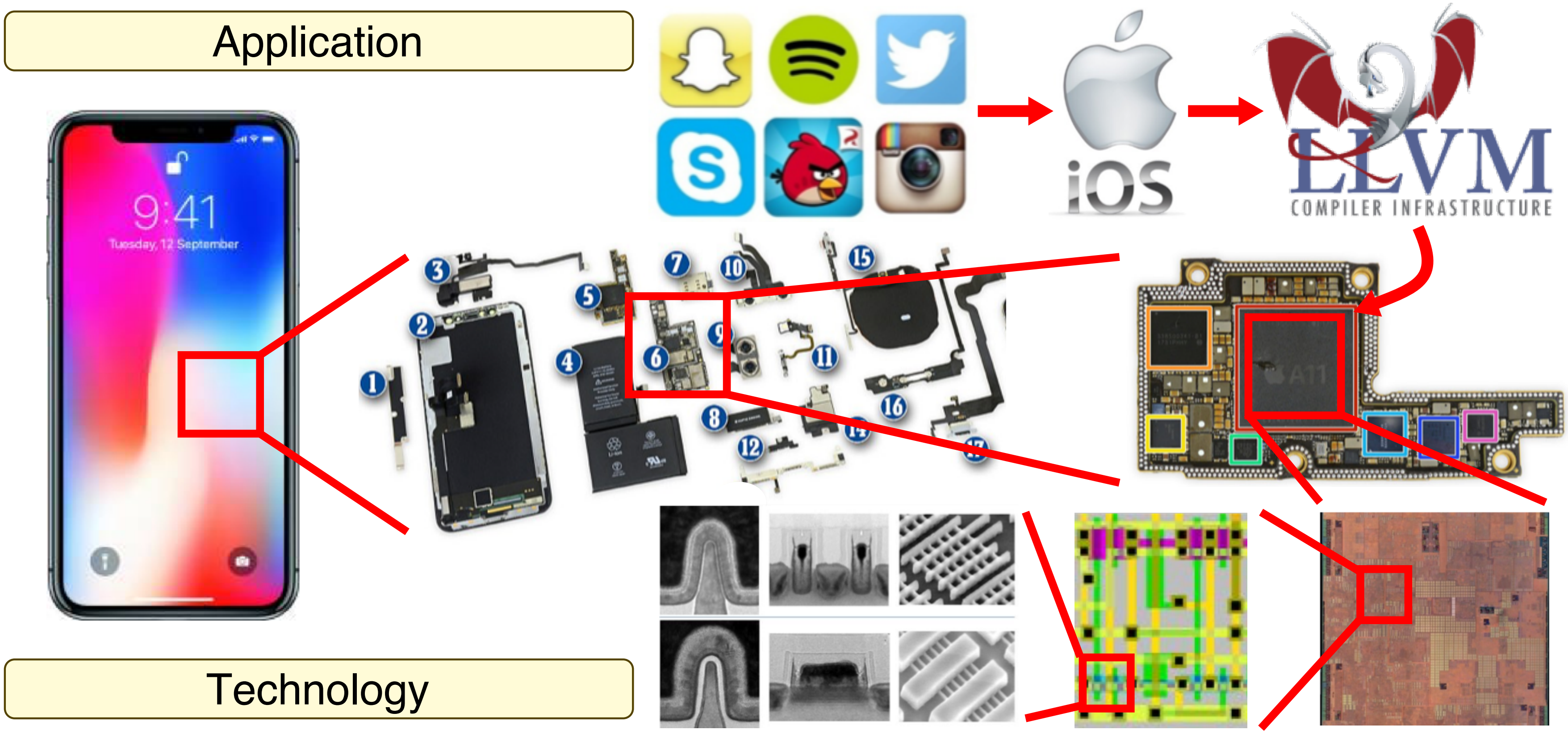


Application

Technology

Gap too large to bridge in one step
(but there are exceptions,
e.g., a magnetic compass)

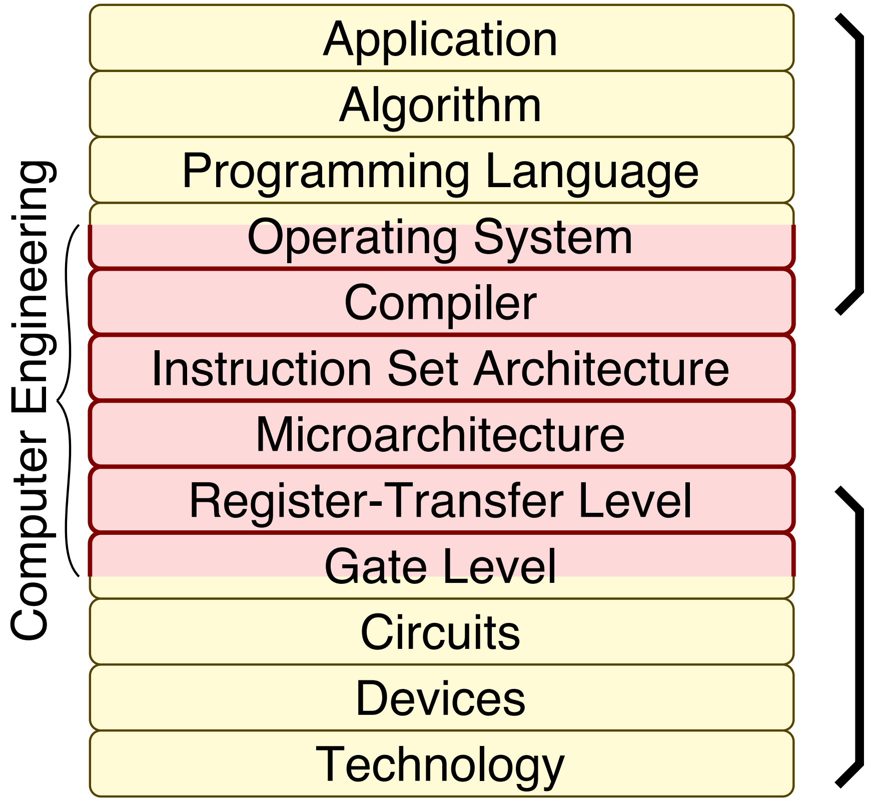
Applications vs. Technology



Application

Technology

The Computer Systems Stack



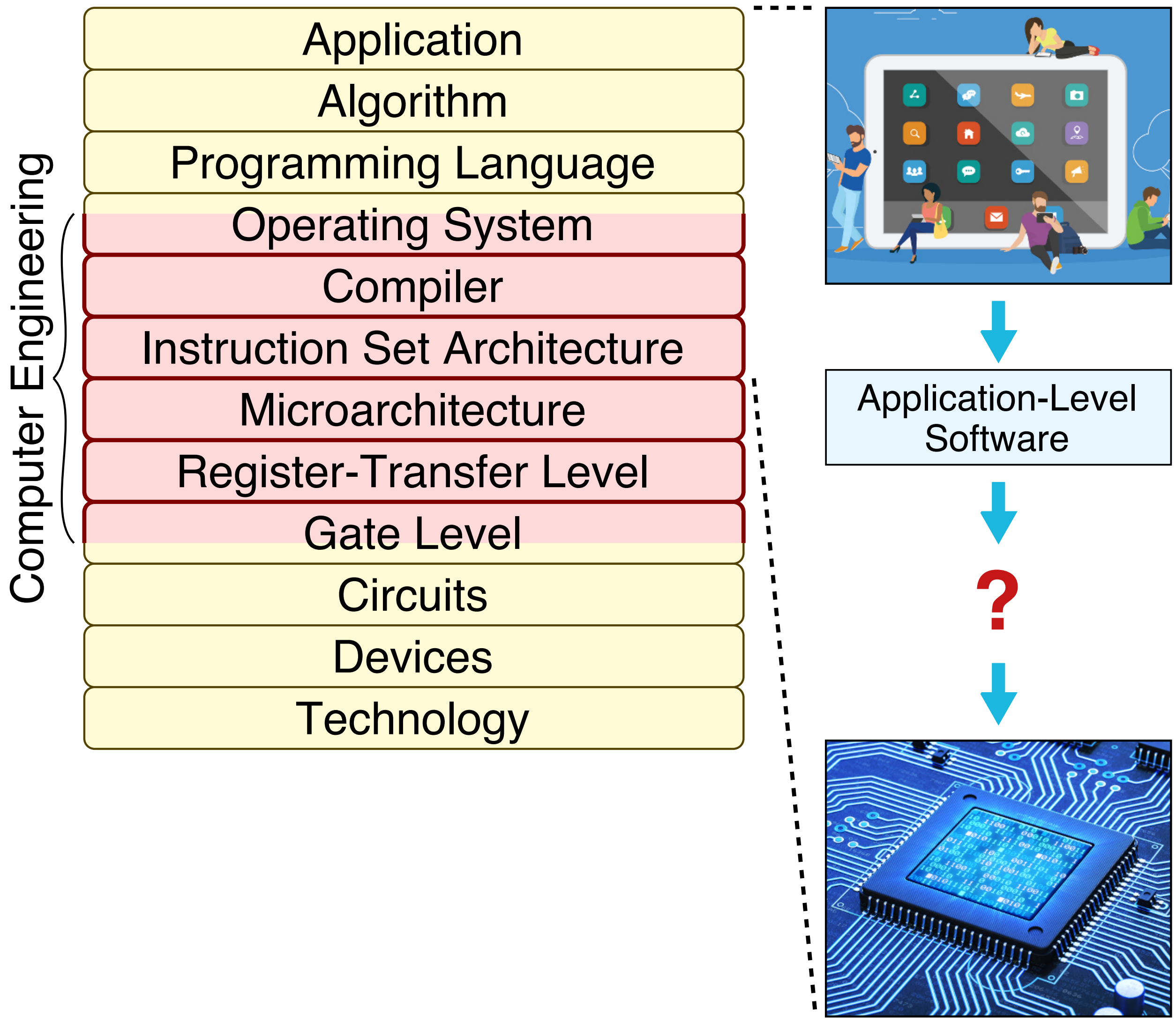
Traditional Computer Science

Computer Engineering is at the interface between hardware and software and considers the entire system

Traditional Electrical Engineering

In its broadest definition, computer engineering is the development of the abstraction/implementation layers that allow us to execute information processing applications efficiently using available manufacturing technologies

Python for Application-Level Programming



Computer Engineering

Application

Algorithm

Programming Language

Operating System

Compiler

Instruction Set Architecture

Microarchitecture

Register-Transfer Level

Gate Level

Circuits

Devices

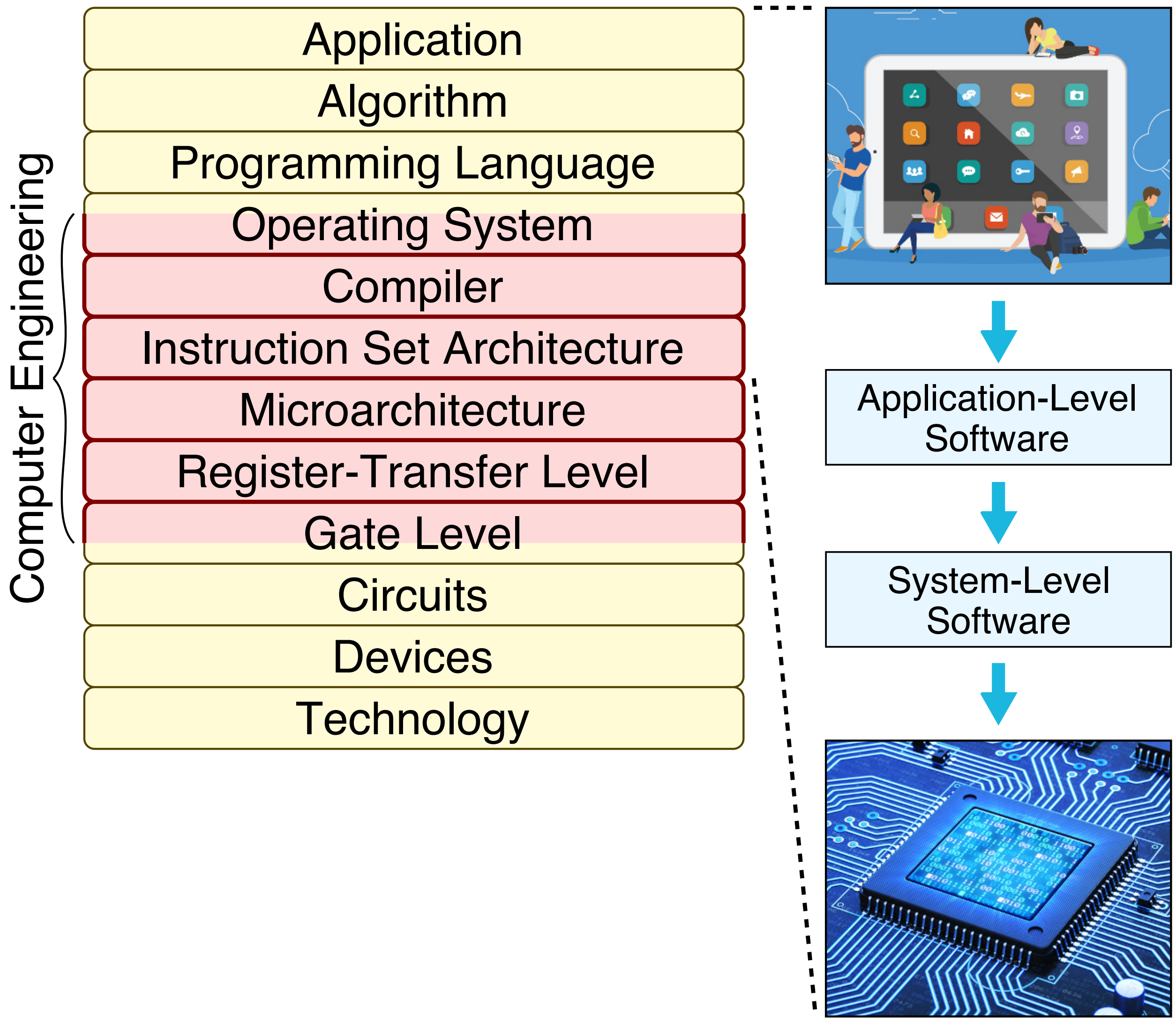
Technology

Application-Level Software

?

- High-level, user-facing software
- Enable productively developing applications that provide new functionality to users
- Enable productively collecting, analyzing, visualizing data
- Sometimes called a productivity-level language

C/C++ for System-Level Programming



Computer Engineering

Application
Algorithm
Programming Language
Operating System
Compiler
Instruction Set Architecture
Microarchitecture
Register-Transfer Level
Gate Level
Circuits
Devices
Technology

Application-Level Software

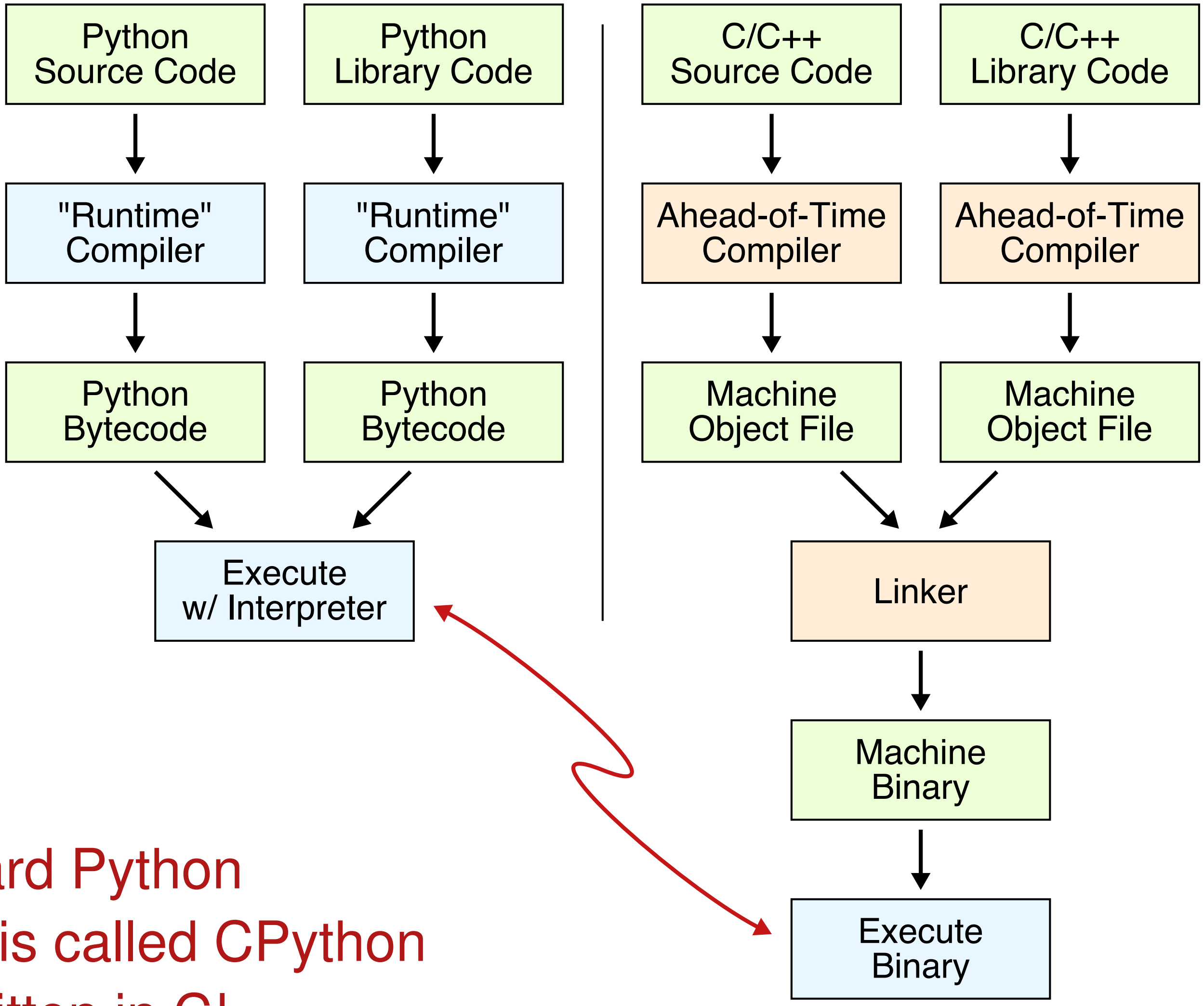
System-Level Software

- Connects application software to the low-level computer hardware
- Enables carefully managing performance and resource constraints
- Sometimes called an efficiency-level language

Dynamically Interpreted vs. Statically Compiled

```
def min(a,b):
  if a < b:
    c = a
  else
    c = b
  return c
```

```
LOAD_FAST
LOAD_FAST
COMPARE_OP
POP_JUMP_IF_F
LOAD_FAST
STORE_FAST
JUMP_FORWARD
LOAD_FAST
STORE_FAST
LOAD_FAST
RETURN_VALUE
```



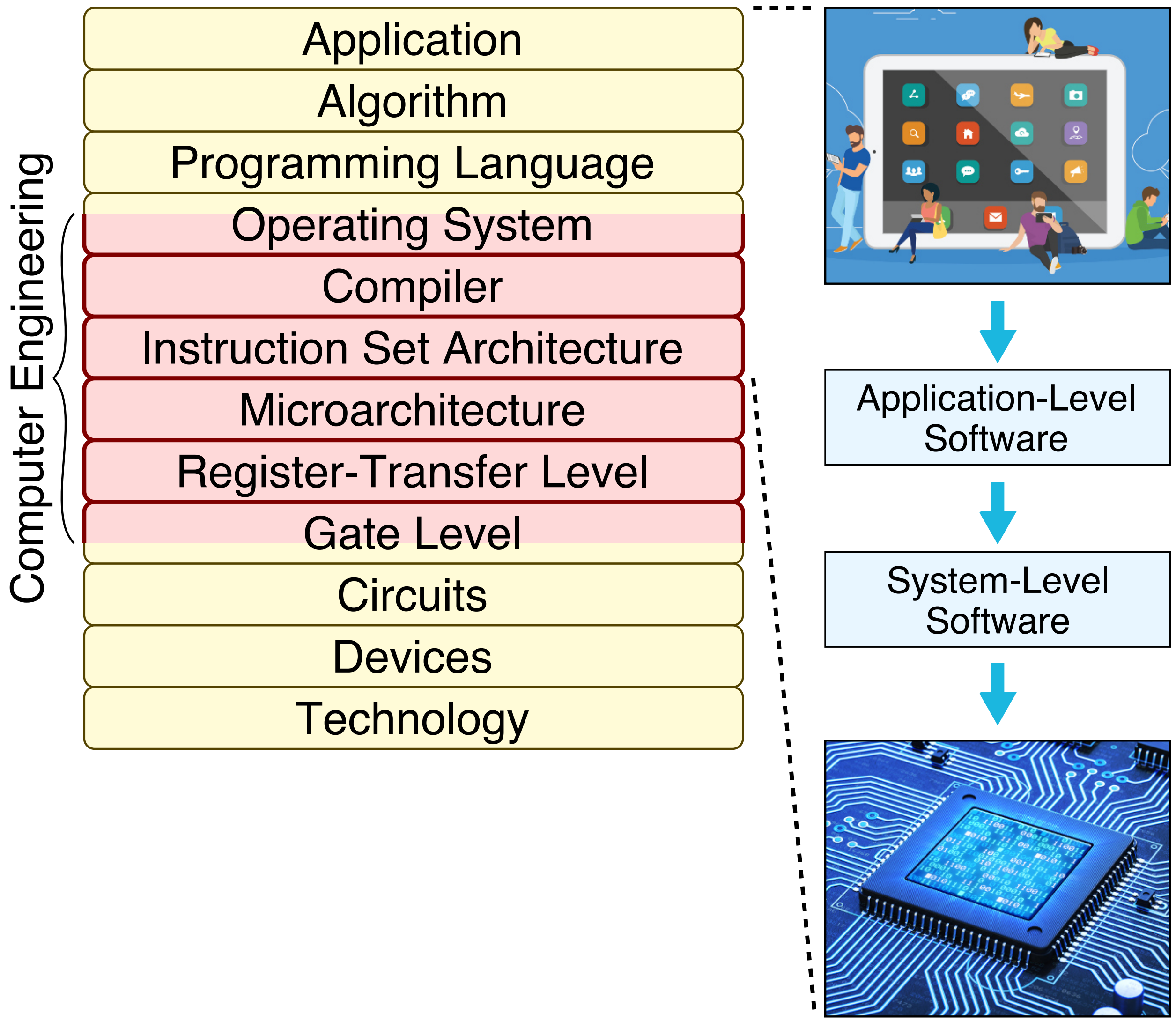
```
int min( int a,
         int b )
{
  int c;
  if ( a < b )
    c = a;
  else
    c = b;
  return c;
}
```

```
pushq %rbp
movq  %rsp,%rbp
cmpl  %esi,%edi
cmovl %edi,%esi
movl  %esi,%eax
popq  %rbp
retq
```

```
1010101
100100010001001
11100111110111
111101001110111
100010011111000
1011101
11000011
```

The standard Python interpreter is called CPython and it is written in C!

Computer Systems Programming is Diverse



Computer Engineering

Application

Algorithm

Programming Language

Operating System

Compiler

Instruction Set Architecture

Microarchitecture

Register-Transfer Level

Gate Level

Circuits

Devices

Technology

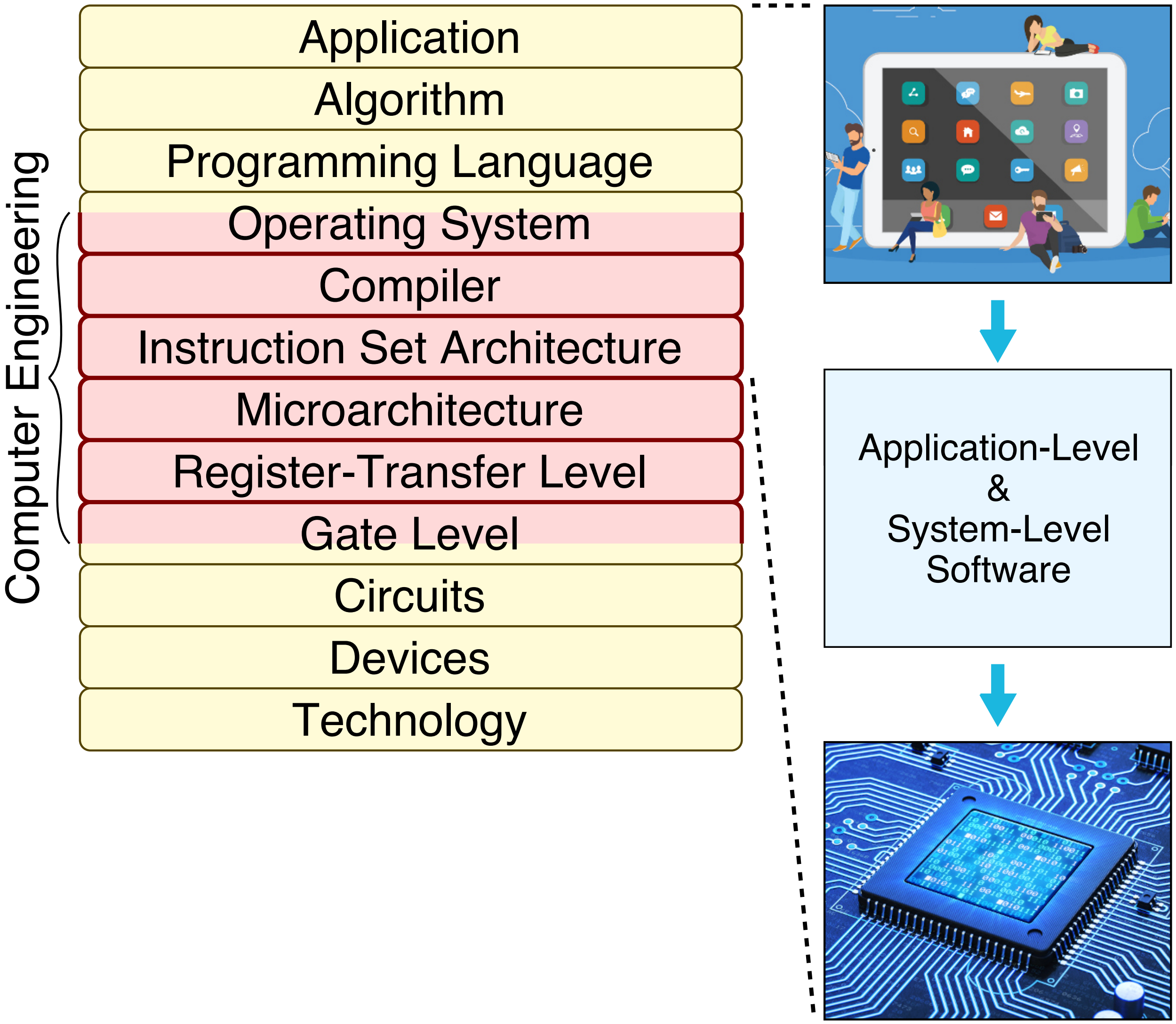
Application-Level Software

System-Level Software

- Python, MATLAB
- Ruby, Javascript
- SQL, LINQ
- NumPy
- GUI frameworks

- Interpreters
- Compilers
- Databases
- Numerical libraries
- Operating systems
- Embedded control

Aside: C/C++ for Application-Level Software



Application

Algorithm

Programming Language

Operating System

Compiler

Instruction Set Architecture

Microarchitecture

Register-Transfer Level

Gate Level

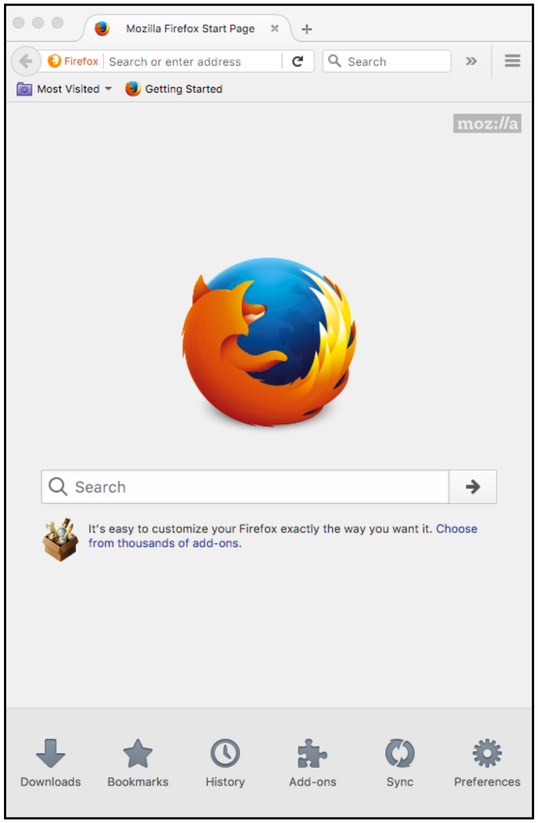
Circuits

Devices

Technology

Computer Engineering

Application-Level & System-Level Software



It's easy to customize your Firefox exactly the way you want it. Choose from thousands of add-ons.

A Tale of Two Programming Languages

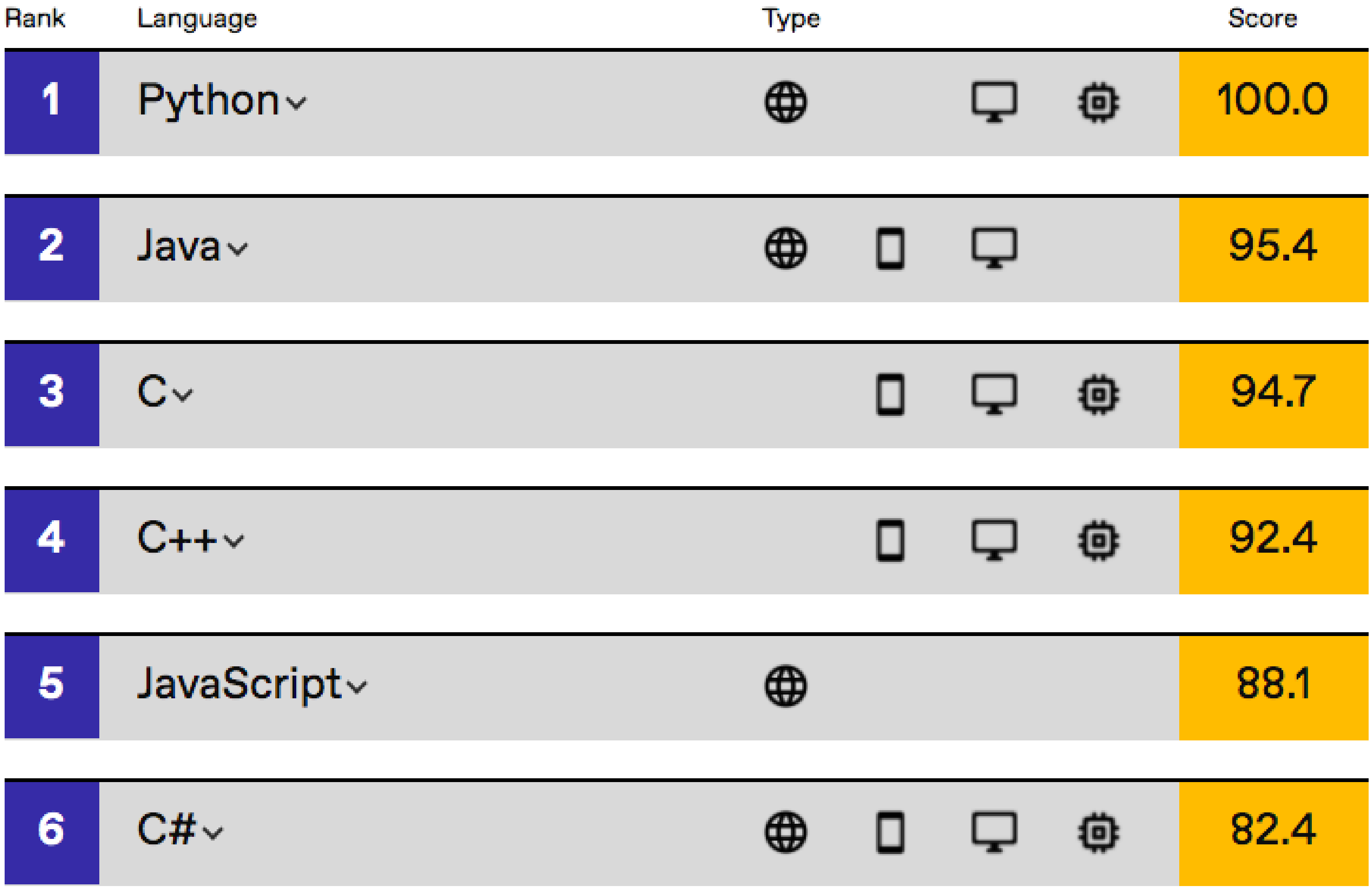
Python Programming Language

- Introduced: 1991
- Most of the machine details are hidden from programmer
- Programmer gives up some control for improved productivity
- Easily supports multiple programming paradigms
- Extensive standard library is included
- Slow and memory inefficient

C/C++ Programming Language

- Introduced: 1972(C), 1979(C++)
- Most of the machine details are exposed to the programmer
- Programmer is in complete control for improved efficiency
- Easily supports multiple programming paradigms
- More limited standard library is included
- Fast and memory efficient

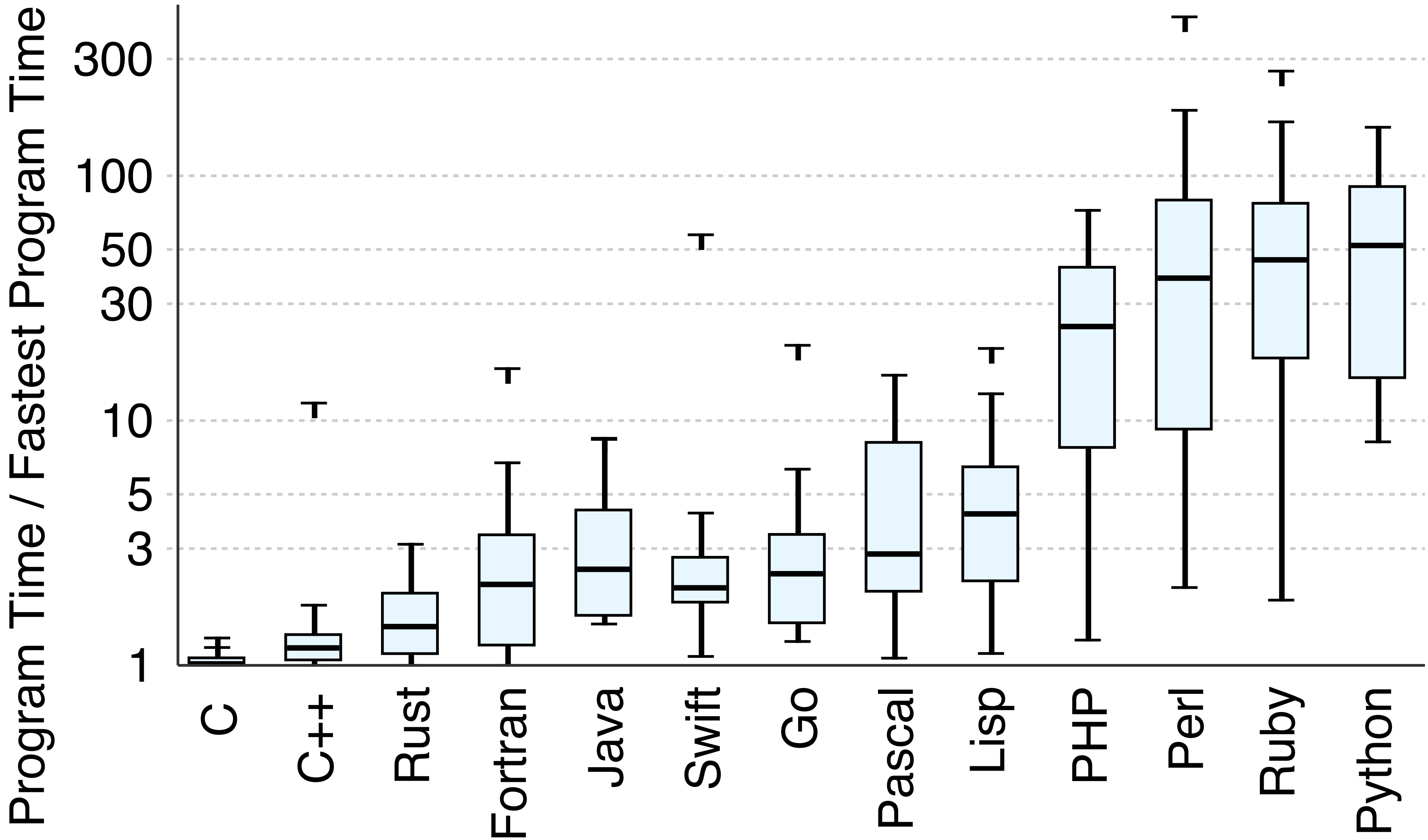
Comparing the Popularity of Python vs. C/C++



Rank	Language	Type	Score
1	Python	🌐 🖥 ▣	100.0
2	Java	🌐 📱 🖥	95.4
3	C	📱 🖥 ▣	94.7
4	C++	📱 🖥 ▣	92.4
5	JavaScript	🌐	88.1
6	C#	🌐 📱 🖥 ▣	82.4

The 2021 Top Programming Languages, IEEE Spectrum

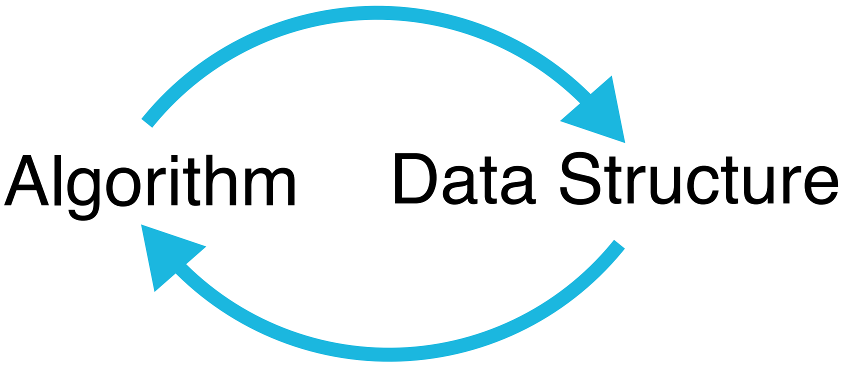
Comparing the Performance of Python vs. C/C++



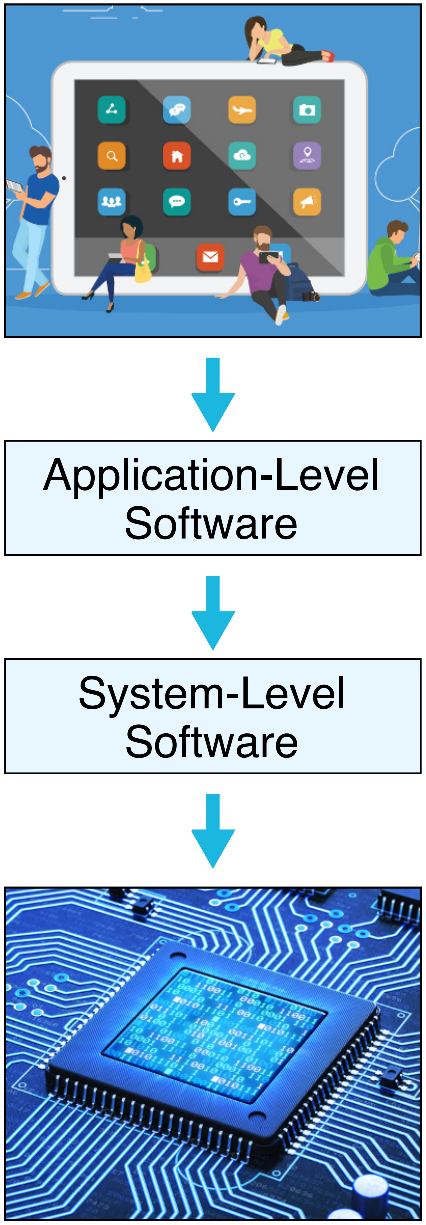
The Computer Language Benchmarks Game

Program = Algorithm + Data Structure

While this course covers C/C++ and system-level programming, this course also builds off of your prior programming experience to further develop your understanding of algorithms and data structures



- **Algorithm:** Clear set of steps to solve any problem instance in a particular class of problems
- **Data Structure:** Way of efficiently organizing and storing data along with operations for accessing and manipulating this data



Application-Level Software

System-Level Software

ECE 2400 / ENGRD 2140 Computer Systems Programming

What is Computer Systems Programming?

Activity: Comparing Algorithms

Trends in Computer Systems Programming

Course Logistics

Activity: Comparing Algorithms

- **Application:** Sort 16 numbers
- **Activity Steps**
  - 1. Half the class will use Algorithm A, half uses Algorithm B
  - 2. When instructor starts timer, flip over worksheet
  - 3. Sort 16 numbers using assigned algorithm
  - 4. Lookup when completed and write time on worksheet
  - 5. Raise hand
  - 6. When everyone is finished, then analyze data
- **Algorithm A**

```
repeat 16 times
  find smallest number not crossed off in input list
  copy smallest number to next open entry in output list
  cross smallest number off input list
```

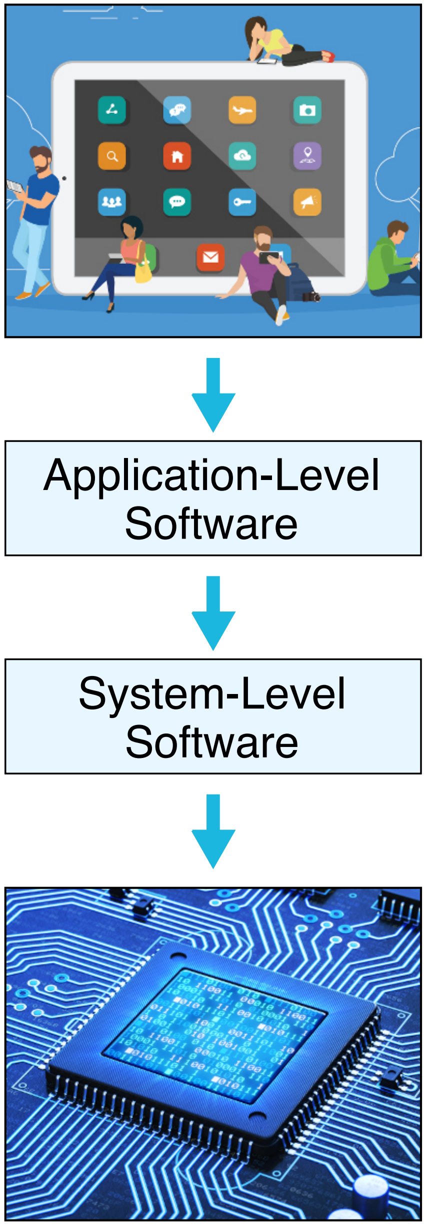
Activity: Comparing Algorithms

- **Algorithm B**

```
repeat 8 times, once for each pair in column 1
  copy smallest from input pair into next entry in column 1
  copy largest from input pair into next entry in column 1

repeat 4 times, once for group of 4 in column 2
  repeat 4 times
    compare top two numbers not crossed off in both groups
    copy smallest number to next open entry in column 2
    cross smallest number off input list

... and so on ...
```



Application-Level Software

System-Level Software

ECE 2400 / ENGRD 2140 Computer Systems Programming

What is Computer Systems Programming?

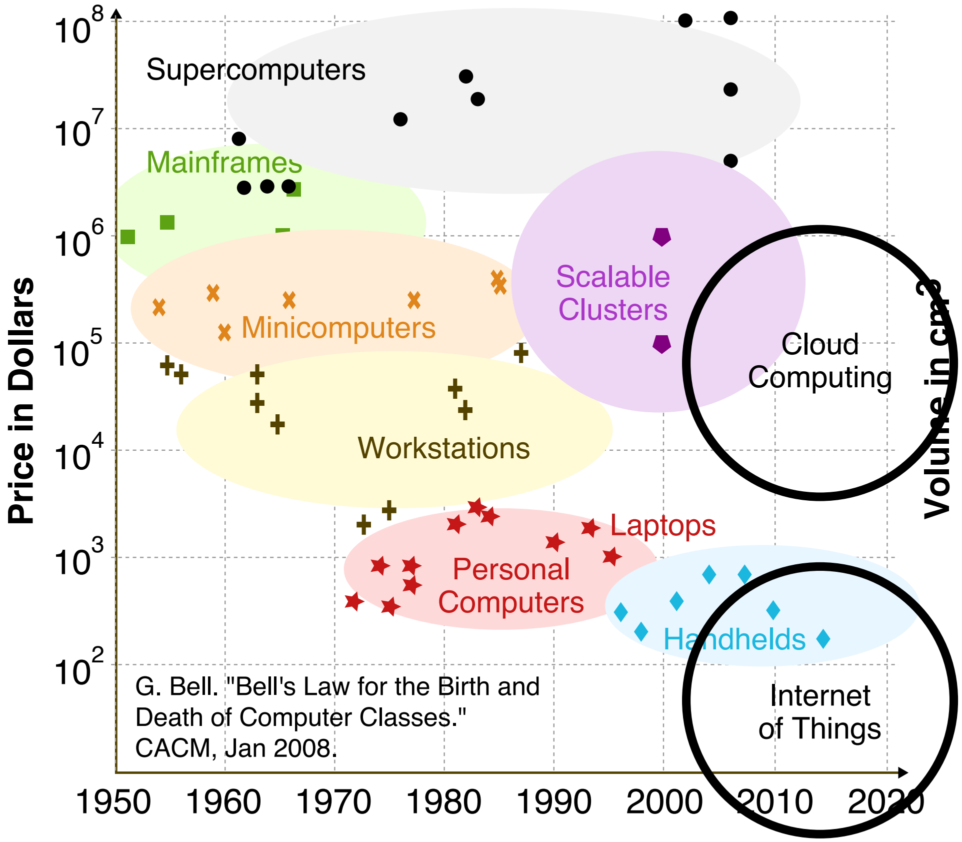
Activity: Comparing Algorithms

Trends in Computer Systems Programming

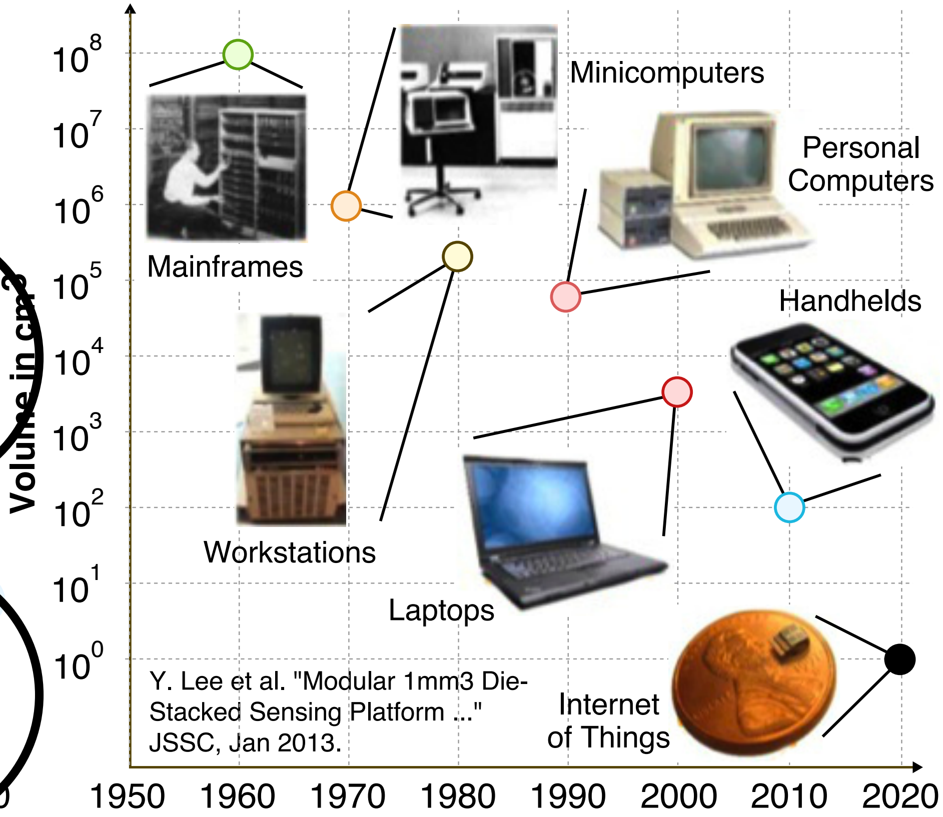
Course Logistics

Trend towards IoT and Cloud w/ Novel Hardware

Roughly every decade a new, smaller, lower priced computer class forms based on a new programming platform resulting in entire new industries



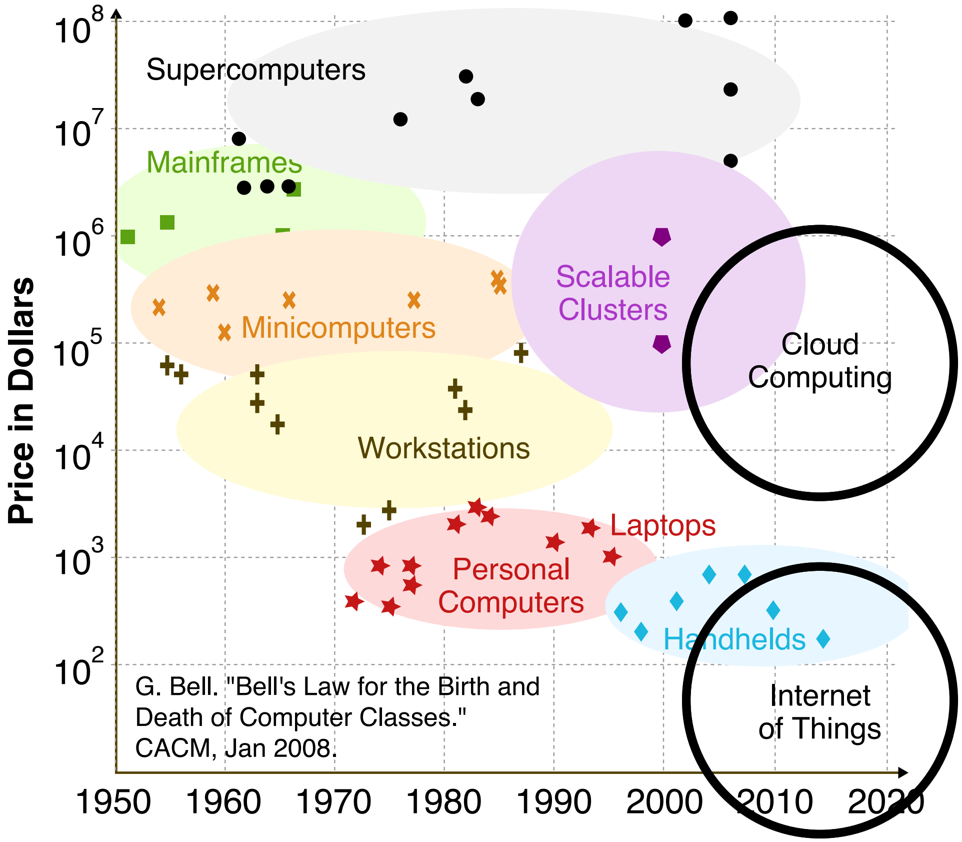
G. Bell. "Bell's Law for the Birth and Death of Computer Classes." CACM, Jan 2008.



Y. Lee et al. "Modular 1mm3 Die-Stacked Sensing Platform ..." JSSC, Jan 2013.

Trend towards IoT and Cloud w/ Novel Hardware

Roughly every decade a new, smaller, lower priced computer class forms based on a new programming platform resulting in entire new industries



G. Bell. "Bell's Law for the Birth and Death of Computer Classes." CACM, Jan 2008.

Cloud Computing

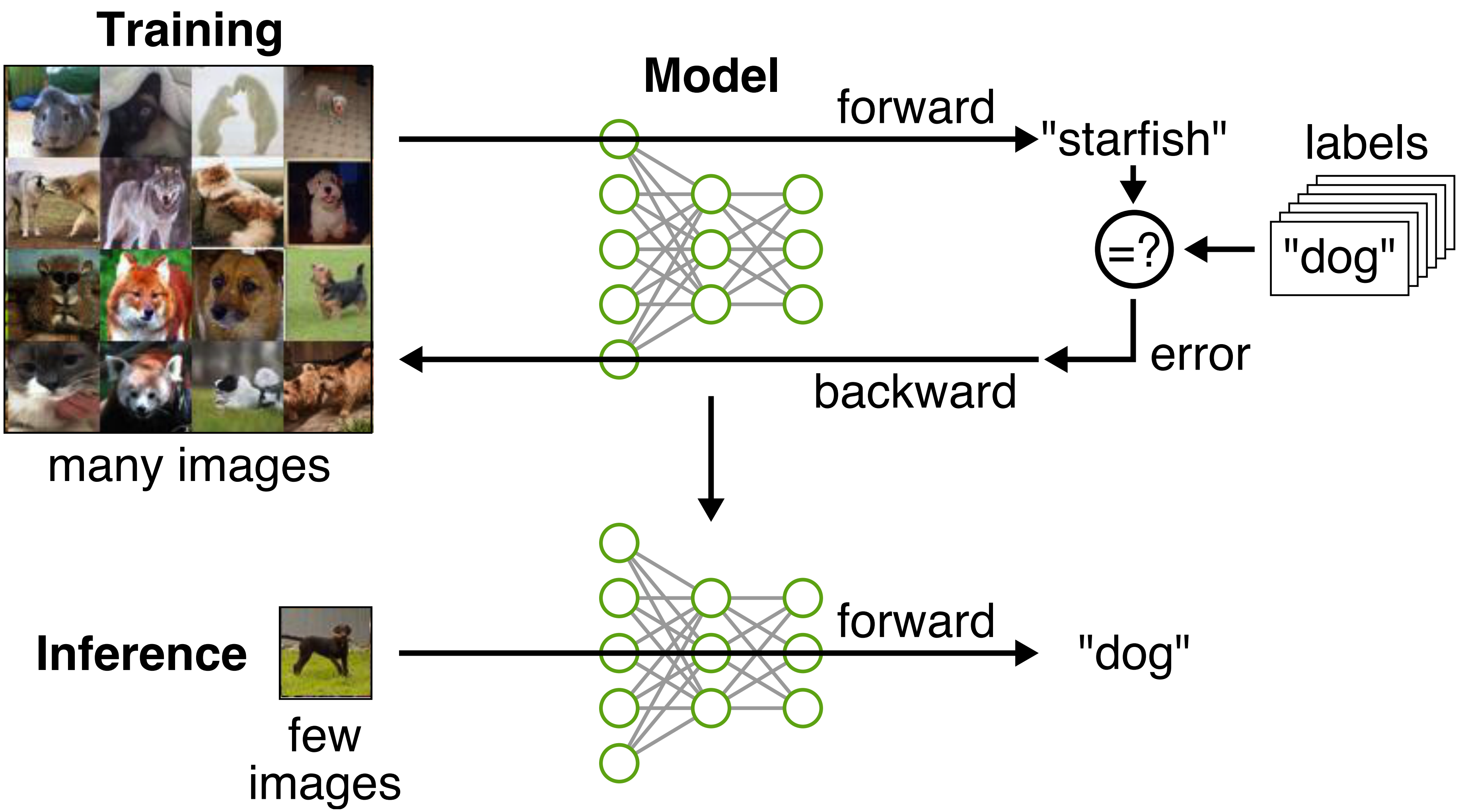
- Often requires low-latency, high-throughput to meet overall application requirements
- Increasingly w/ specialized HW

Internet-of-Things

- Very limited resource constraints (e.g., energy, memory)
- Requires carefully managing these resources to meet overall application requirements
- Increasingly w/ specialized HW

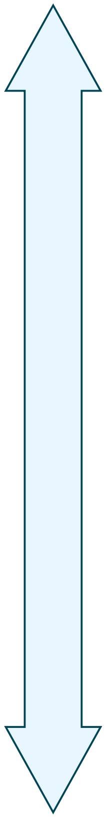
Example Application: Image Recognition

Machine Learning (ML): Training vs. Inference

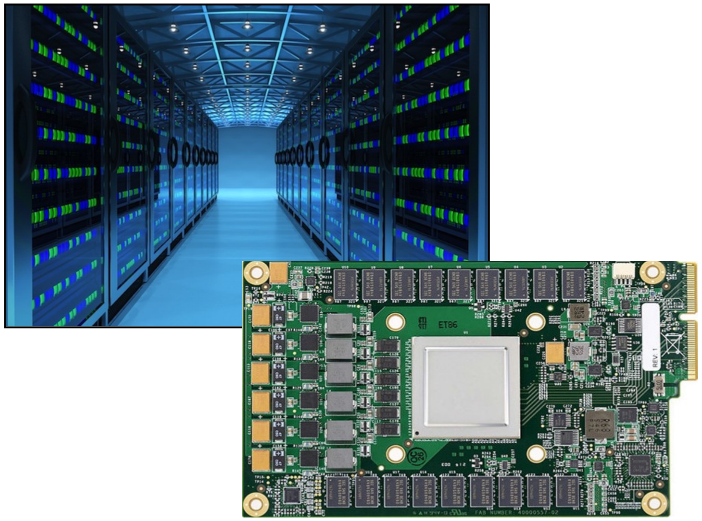


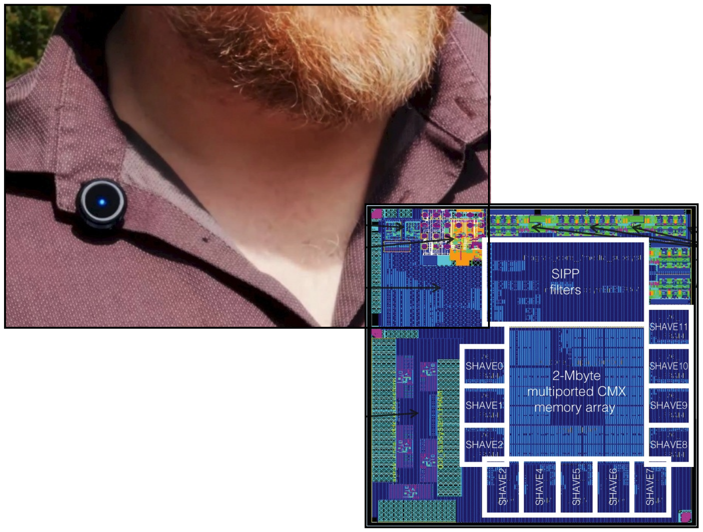
Computer Systems Programming in ML

Cloud Computing



Internet of Things



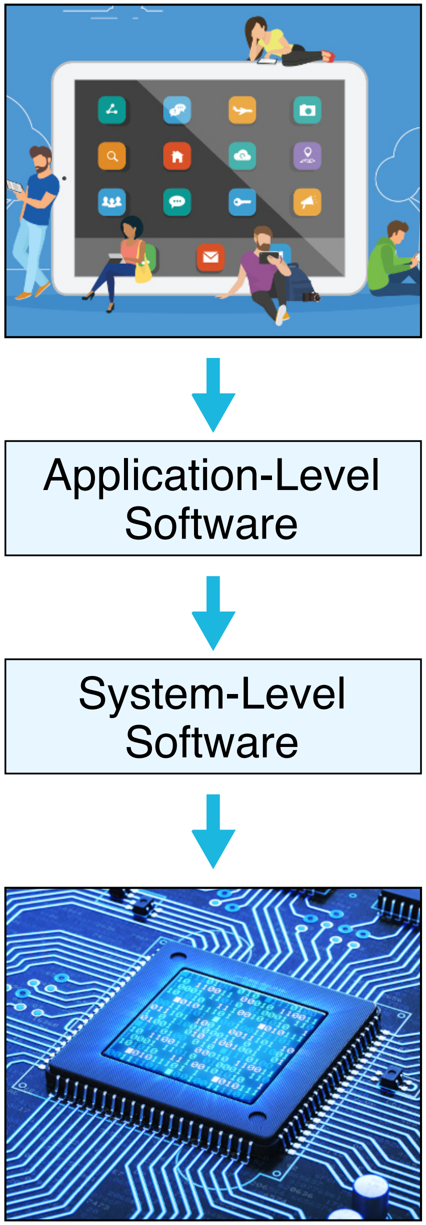


Google TPU

- Training is done using the TensorFlow C++ framework
- Training can take weeks
- Google TPU is custom chip
- High-level ML frameworks use C++ under the hood

Movidius Myriad 2

- Custom chip for ML on embedded IoT devices
- Carefully crafted C/C++ ML libraries for inference
- Embedded control also in C/C++



Application-Level Software

System-Level Software

ECE 2400 / ENGRD 2140 Computer Systems Programming

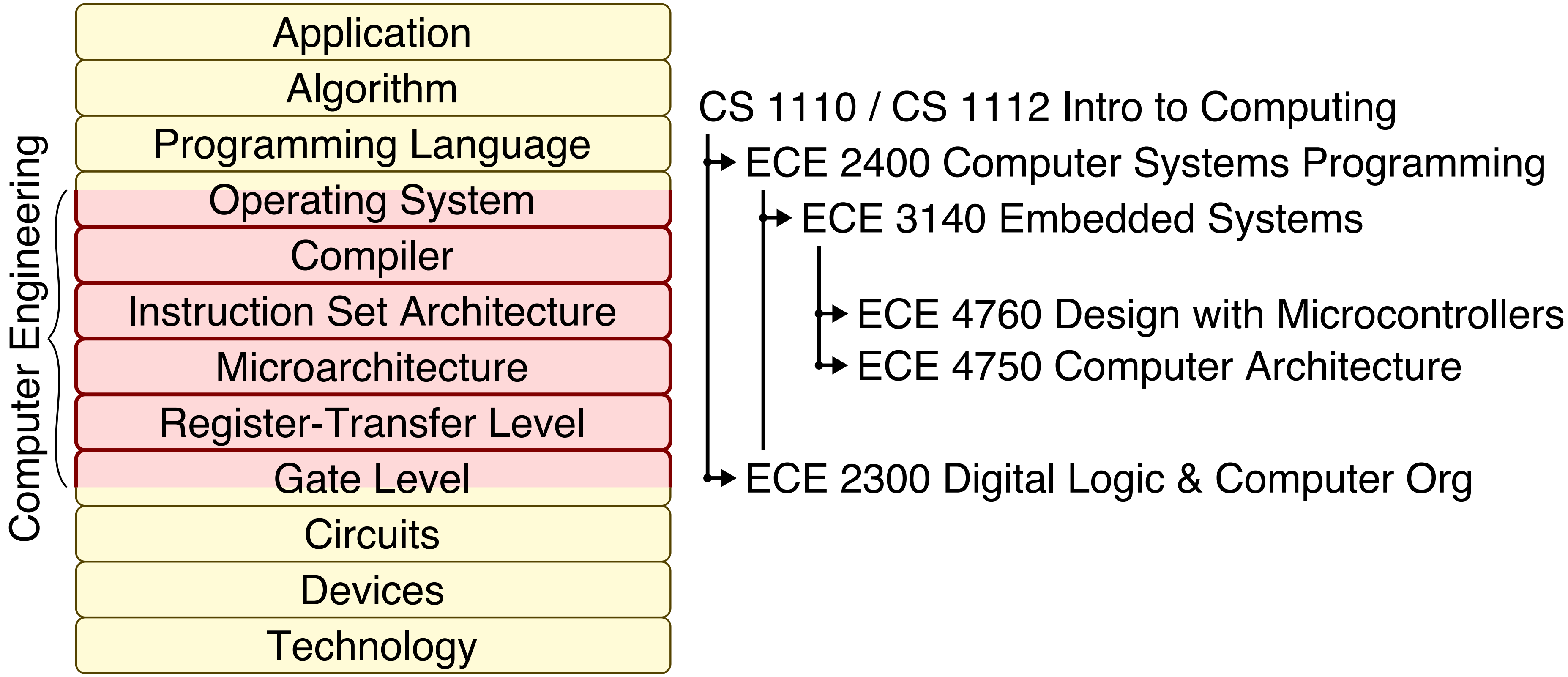
What is Computer Systems Programming?

Activity: Comparing Algorithms

Trends in Computer Systems Programming

Course Logistics

ECE 2400 Within the Engineering Curriculum



Application
Algorithm
Programming Language
Operating System
Compiler
Instruction Set Architecture
Microarchitecture
Register-Transfer Level
Gate Level
Circuits
Devices
Technology

Computer Engineering

CS 1110 / CS 1112 Intro to Computing
ECE 2400 Computer Systems Programming
ECE 3140 Embedded Systems
ECE 4760 Design with Microcontrollers
ECE 4750 Computer Architecture
ECE 2300 Digital Logic & Computer Org

ECE 2400 is also an ENGRD and thus satisfies the engineering distribution requirement

ECE 2400 can be an excellent way to generally incorporate programming into your non-ECE engineering curriculum

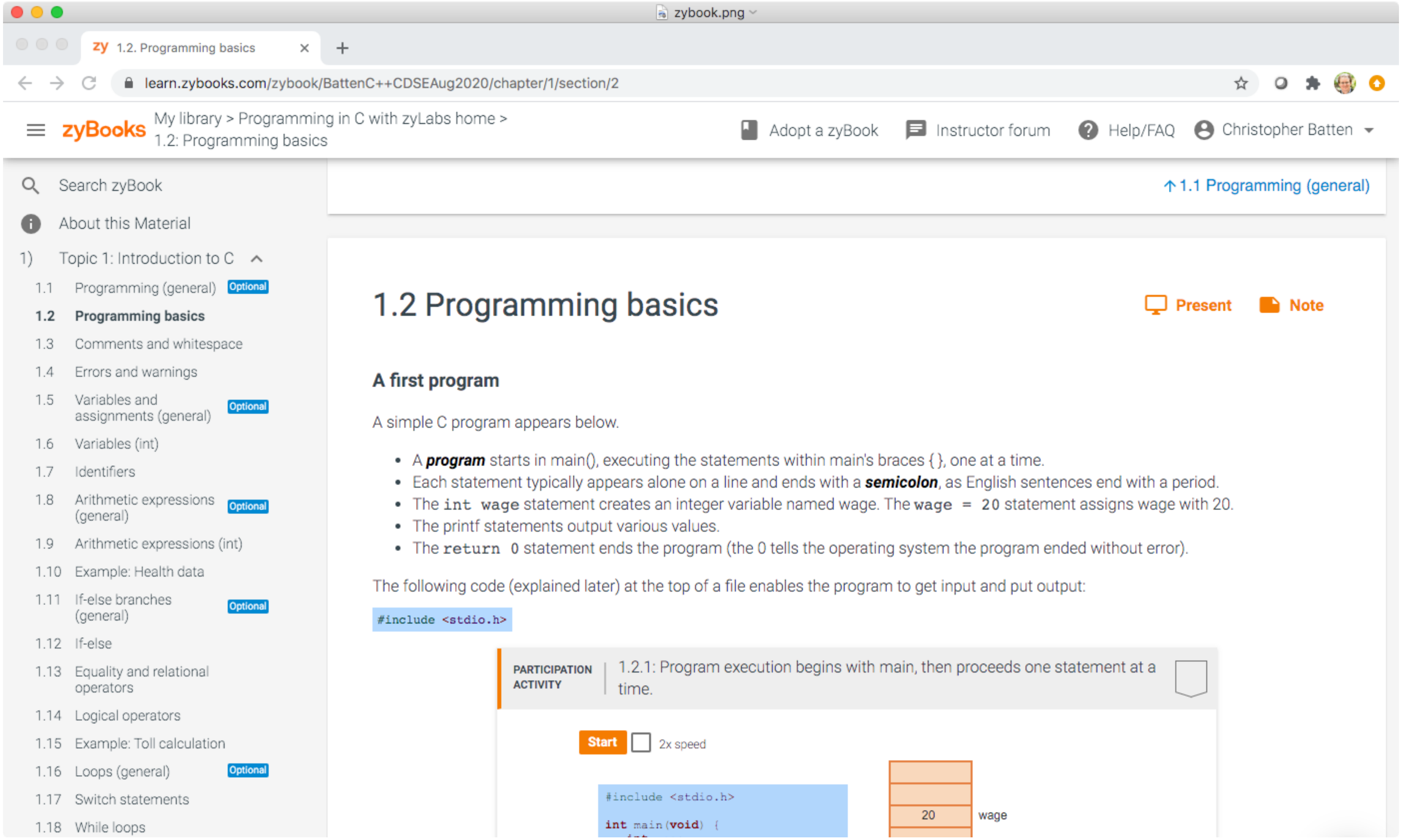
Course Objectives

- **describe** a variety of algorithms and data structures and how to analyze these algorithms and data structures in terms of time and space complexity
- **apply** the C/C++ programming languages to implement algorithms and data structures using different programming paradigms
- **evaluate** algorithm and data structure alternatives and make a compelling qualitative and/or quantitative argument for one approach
- **create** non-trivial C/C++ programs (roughly 1,000 lines of code) and the associated testing strategy from an English language specification
- **write** concise yet comprehensive technical reports that describe a program implemented in C/C++, explain the testing strategy used to verify functionality, and evaluate the program to characterize its performance and memory usage

Course Structure

- **Part 1: Procedural Programming**
  - introduction to C; variables; expressions; functions; conditional & iteration statements; recursion; static types; pointers; arrays; dynamic allocation
- **Part 2: Basic Algorithms and Data Structures**
  - lists; vectors; complexity analysis; sorting algorithms: insertion, selection, merge, quick, radix; ADTs: stacks, queues, priority queues, sets, maps
- **Part 3: Multi-Paradigm Programming**
  - transition to C++; namespaces; flexible function prototypes; references; exceptions; new/delete; *object oriented programming:* C++ classes and inheritance for dynamic polymorphism; *generic programming:* C++ templates for static polymorphism; *functional programming:* C++ functors and lambdas; *concurrent programming:* C++ threads and atomics
- **Part 4: More Algorithms and Data Structures**
  - trees (binary search trees; binary heaps); tables (lookup tables; hash tables); graphs (DFS, BFS, shortest path, minimum spanning trees)

zyBook: New Interactive Online Textbook


zybook.png
zy 1.2. Programming basics
learn.zybooks.com/zybook/BattenC++CDSEAug2020/chapter/1/section/2
zyBooks
My library > Programming in C with zyLabs home >
1.2: Programming basics
Adopt a zyBook
Instructor forum
Help/FAQ
Christopher Batten
Search zyBook
About this Material
↑ 1.1 Programming (general)
Present
Note
Start
2x speed
20
wage

1) Topic 1: Introduction to C

- 1.1 Programming (general) Optional
- **1.2 Programming basics**
- 1.3 Comments and whitespace
- 1.4 Errors and warnings
- 1.5 Variables and assignments (general) Optional
- 1.6 Variables (int)
- 1.7 Identifiers
- 1.8 Arithmetic expressions (general) Optional
- 1.9 Arithmetic expressions (int)
- 1.10 Example: Health data
- 1.11 If-else branches (general) Optional
- 1.12 If-else
- 1.13 Equality and relational operators
- 1.14 Logical operators
- 1.15 Example: Toll calculation
- 1.16 Loops (general) Optional
- 1.17 Switch statements
- 1.18 While loops

1.2 Programming basics

A first program

A simple C program appears below.

- A ***program*** starts in main(), executing the statements within main's braces { }, one at a time.
- Each statement typically appears alone on a line and ends with a ***semicolon***, as English sentences end with a period.
- The `int wage` statement creates an integer variable named wage. The `wage = 20` statement assigns wage with 20.
- The printf statements output various values.
- The `return 0` statement ends the program (the 0 tells the operating system the program ended without error).

The following code (explained later) at the top of a file enables the program to get input and put output:

```
#include <stdio.h>
```

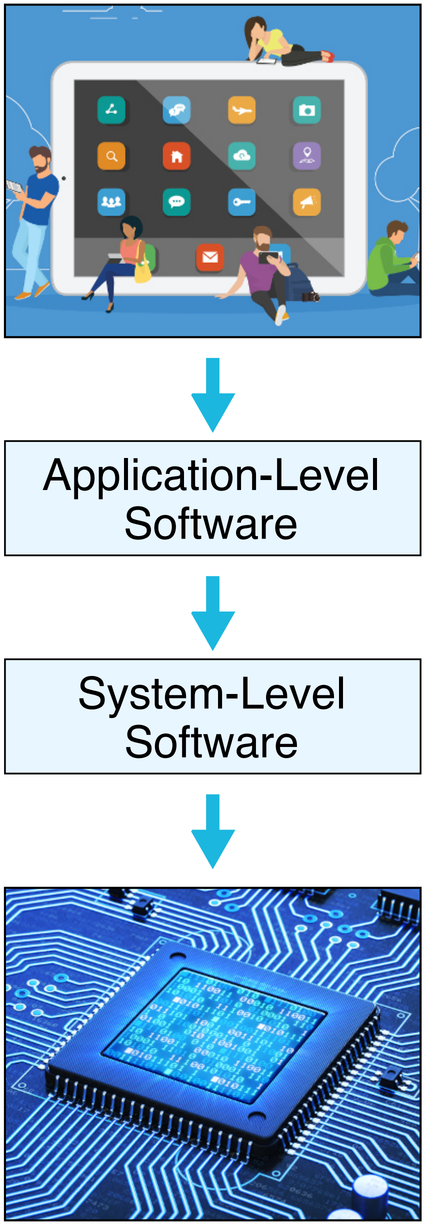
PARTICIPATION ACTIVITY | 1.2.1: Program execution begins with main, then proceeds one statement at a time.

```
#include <stdio.h>

int main(void) {
```

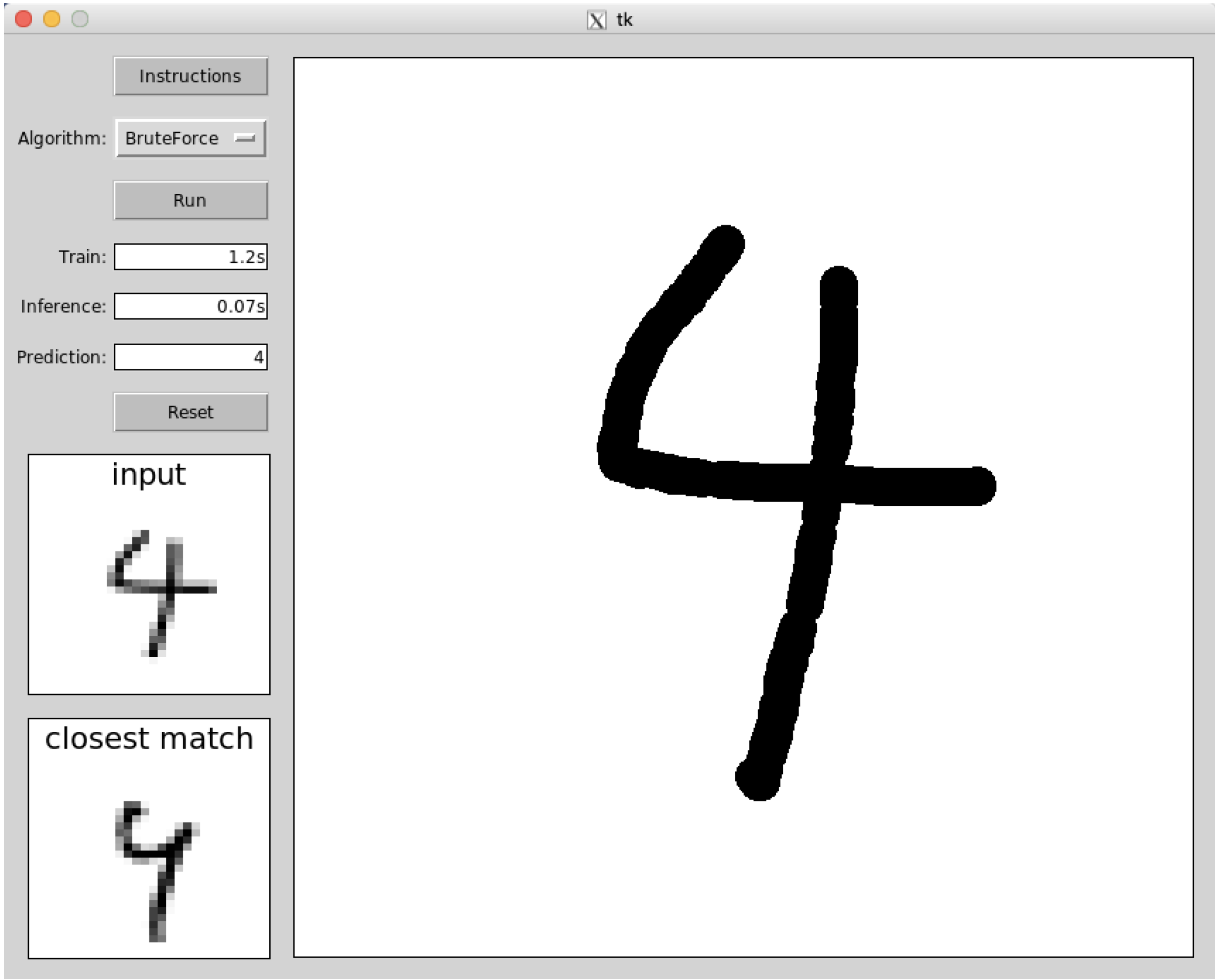
Programming Assignments

- **PA1–3: Fundamentals**
  - PA1: Math functions
  - PA2: List and Vector Data Structures
  - PA3: Sorting Algorithms
- **PA4–5: Handwriting Recognition System**
  - PA5: Linear vs. Binary Searching
  - PA5: Trees vs. Tables
- **Every programming assignment involves**
  - C/C++ "agile" programming
  - State-of-the-art tools for build systems, version control, continuous integration, code coverage
  - Performance measurement
  - Short technical report



Application-Level Software

System-Level Software

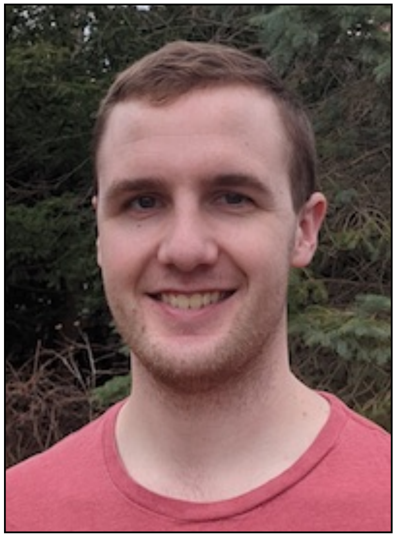


Algorithm: BruteForce

Train: 1.2s

Inference: 0.07s

Prediction: 4



Nick Cebry
ECE Phd

Ryan McMahon
ECE MEng

Guadalupe Bernal
ECE Junior

Michael Egbueze
CS Senior

Eric Hall
ECE Senior

Sonal Parab
CS Senior

Anya Prabowo
ECE Junior

Chidera Wokonko
ECE Junior

	Mon	Tue	Wed	Thu	Fri
10:00am	**Lecture**		**Lecture**		**Lecture**
	(219 Phillips)		(219 Phillips)		(219 Phillips)
11:00am					
:	:	:	:	:	:
2:00pm					
					Section
3:00pm					(225 Upson)
4:00pm					
		Office Hours			
5:00pm		(323 Rhodes)			
6:00pm					
7:00pm					
	Lab/Office	**Lab/Office**	**Lab/Office**	**Lab/Office**	
8:00pm	**Hours**	**Hours**	**Hours**	**Hours**	
	(225 Upson)	(225 Upson)	(225 Upson)	(225 Upson)	
9:00pm					
10:00pm					

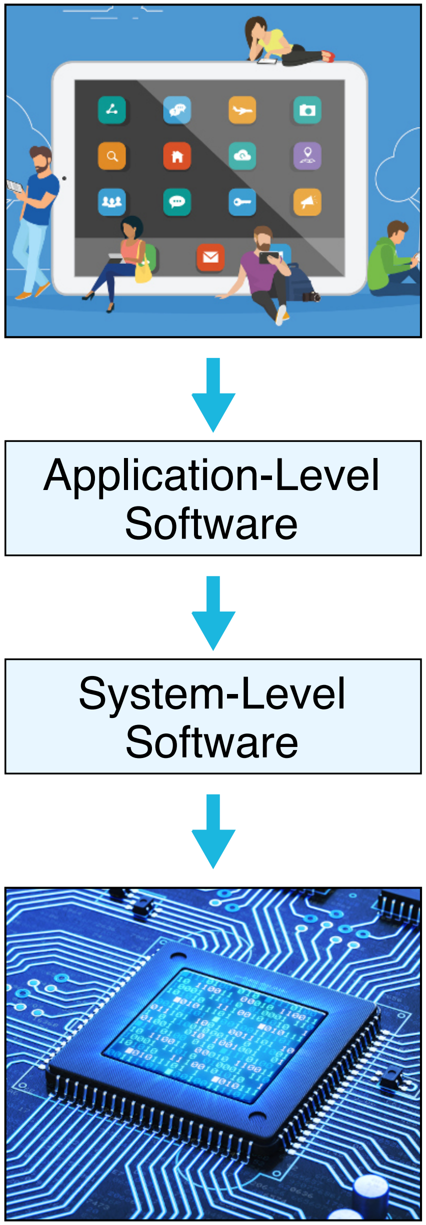
Extra zoom office hours for Prof. Batten are from 7:30–8:30pm on Tuesdays

Frequently Asked Questions

- I have not taken CS 1110 nor CS 1112, can I take this class?
  - We assume some basic programming experience, discuss with instructor
- **ECE Majors –** How does ECE 2400 satisfy degree requirements?
  - ECE 2400 can count as your second ENGRD course
  - ECE 2400 can count as an outside-ECE technical elective
  - ECE 2400 satisfies the ECE advanced programming requirement
- **CS Majors –** Can I use ECE 2400 in place of CS 2110?
  - Yes but you should probably take CS 2110
- **ECE/CS Dual Majors –** Can I use ECE 2400 in place of CS 2110?
  - Absolutely! (NEW)
- **CS Minors –** Can I use ECE 2400 in place of CS 2110?
  - Absolutely! (NEW)

Frequently Asked Questions

- **Other Majors –** How does ECE 2400 satisfy degree requirements?
  - ECE 2400 can count as one of your two required ENGRD courses
  - CS 2110 and ECE 2400 are in the same ENGRD category, so you cannot use both of them as your two ENGRD courses
- Can I take both ECE 2400 and CS 2110?
  - Sure! (recall popularity and performance data)



Application-Level Software

System-Level Software

Take-Away Points

- Computer systems programming involves developing software to connect the low-level computer hardware to high-level, user-facing application software and usually requires careful consideration of performance and resource constraints
- We are entering an exciting era where computer systems programming will play a critical role in enabling both cloud computing and the internet-of-things