CS 240 in context
How Computers Work

1. Hardware
   - Devices (transistors, etc.)
   - Digital Logic
   - Microarchitecture
   - Instruction Set Architecture
   - Operating System
   - Compiler/Interpreter
   - Programming Language
   - Program, Application, Algorithm

2. Software
   - Solid-State Physics

CS 240 in Context
Skills for Thinking and Programming

Few of you will build new HW, OS, compiler, but...

1. Effective programmers and computer scientists understand their tools and systems.
2. The skills and ideas you learn here apply everywhere.

Reason about computational models, translation.

Debug for correctness and performance (with tools to help).

Assess costs and limits of representations.

"Figure it out" via documentation, experiments, critical thinking.

Remember low-level implications of high-level choices.
Big Ideas in CS, Systems, and beyond

Abstraction
Do not start every project with transistors. Abstraction is beautiful and empowering, but real abstractions have leaks and wrinkles.

Translation
Between layers of abstraction. Structured computation.

Representation
No representation without taxation. Representations have costs.

Performance
Memory: clever, imperfect abstraction. Tiny code changes, huge impact.

Security + Reliability
Trickiest exploits & errors involve multiple layers, even hardware!

These things matter more every day.
The **GHOST vulnerability** is a buffer overflow condition that can be easily exploited locally or remotely, which makes it extremely dangerous. This vulnerability is named after the `GetHOST` function involved in the exploit.
Ariane 5 Rocket, 1996
Exploded due to cast of 64-bit floating-point number to 16-bit signed number. Overflow.

1998
Mars Climate Orbiter
Disintegrated due to mismatched units in Lockheed-Martin / NASA software components.
"... a Model 787 airplane that has been powered continuously for 248 days can lose all alternating current (AC) electrical power due to the generator control units (GCUs) simultaneously going into failsafe mode ... This condition is caused by a software counter internal to the GCUs that will overflow after 248 days of continuous power. We are issuing this AD to prevent loss of all AC electrical power, which could result in loss of control of the airplane." --FAA, April 2015
Toyota "Unintended Acceleration Events"

Oklahoma jury:
"Spaghetti Code" = "reckless disregard"

>10,000 global variables

81,514 violations of MISRA-C coding rules
   Expect 3 minor bugs + 1 major bug per 30 violations

Task/process monitoring failed to monitor tasks/processes
Memory corruption

(Wait, it was written in C?!?!?!)
How could we improve computer systems?

Efficiency
- Time, space, programmer time
- Cost, availability: https://opendatakit.org/about/deployments/

Programmability
- Maintainability, creativity, accessibility, inclusivity, debuggability, testability

Reliability
- Correctness, safety, predictability,

Auditability, provability, analyzability, transparency

Security, privacy

Ownership, control, openness, privacy, rights
- Who owns/controls computing infrastructure, computation/software, data?
- How can systems support personal rights or prevent their compromise?

...
Discussion

In groups of 3. Map ideas on board section. Share at end.

1. **How do computer systems design and implementation choices affect people? The environment?**

   Regardless of topic or application, please focus on impacts/implications of the systems/implementation.

2. **How could changes at the systems/implementation level better support positive impacts or mitigate negative impacts of computing applications on people? The environment?**

3. **What applications are too critical (not) to rely on computing? Why?**