Beyond CS240

How Computers Work

1. Devices (transistors, etc.)
2. Solid State Physics
3. Hardware
   - Digital Logic
   - Microarchitecture
   - Microarchitecture
4. Software
   - Operating System
   - Programming Language
   - Compiler/Interpreter
5. Beyond CS240

CS 240 in Context

2. Foundations
   - CS 240: Computer Systems/Organization
   - CS 343: Distributed Computing
   - CS 34X?: Computer Architecture (Olin ENGR3410 Computer Architecture)
   - CS 304: Databases with Web Interfaces
   - CS 310?: Cryptography (MIT 6.6510 Applied Cryptography and Security; 6.3620 Cryptography and Cryptanalysis)

3. 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.
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.