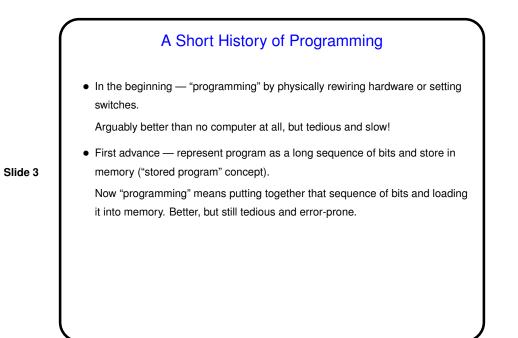
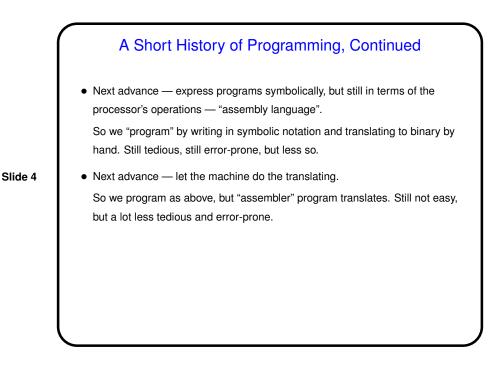
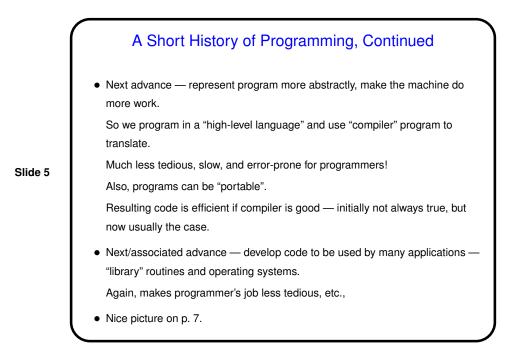


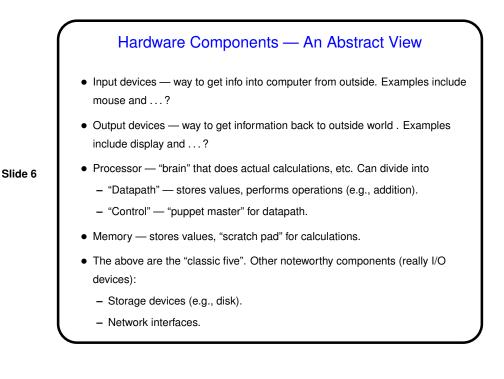
Slide 1

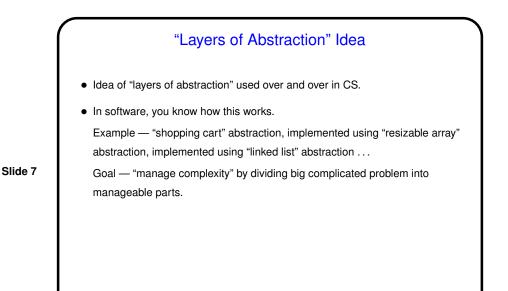
	"It's All Ones and Zeros"
Slide 2	 Computers only "understand" binary. Why? Easier to design hardware in terms of two distinct states (rather than, say, 10). Hence everything — data and instructions — must be represented in terms of 0 and 1.
	How to represent data?
	 Integers — binary numbers, as in CS 1320.
	- Non-integers (e.g., double) — "floating point" (later).
	- Characters — small binary numbers (ASCII, Unicode).
	- MS Word files, MP3s, etc., - more complex, but similar ideas.
	• How to represent instructions? "Machine language". Very briefly for now:
	 Processor has a small repertoire of operations, each with limited number of operands.
	 Each "instruction" is an operation plus operands. Can represent it using a number for the instruction and numbers for the operands.

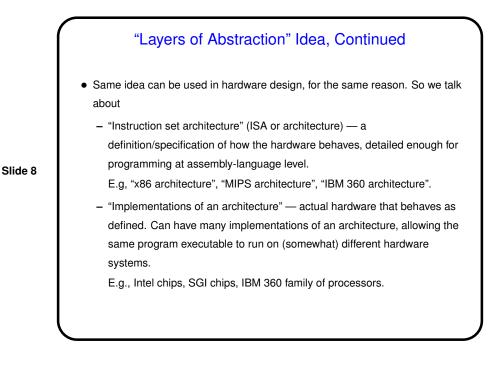


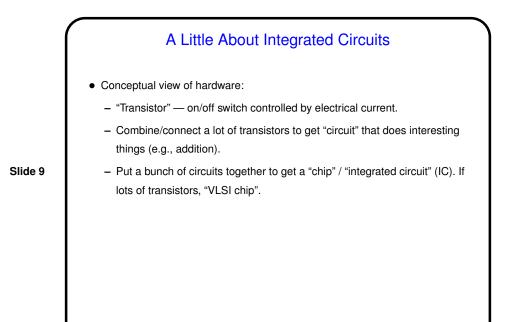


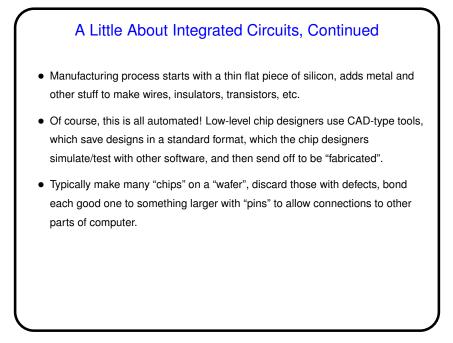




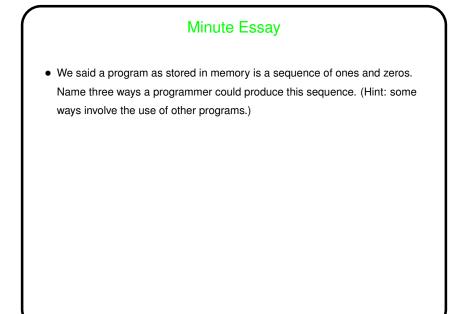








Slide 10



Slide 11