University of Illinois at Urbana-Champaign
Dept. of Electrical and Computer Engineering

ECE 120: Introduction to Computing

Hexadecimal, Text, and Terminology for Representations

Some Sugar-Coating for Humans

Bits are a bit of a pain. For example, try to memorize this pattern:

00010011010101100111

But **computers always use bits**!

Humans, on the other hand,

◦ can use base 16,
◦ usually called **hexadecimal**, or **hex**,
◦ to make dealing with bit patterns easier.

Have you memorized the pattern? Hurry up!

Convert Hex to/from Binary in Groups of 4 Bits

Hex includes A through F to get 16 digits:

0 1 2 3 4 5 6 7 8 9 A B C D E F

$16 = 2^4$, so **each hex digit represents four bits**.

Remember:

◦ Use of **hex only serves to help humans write and remember bits**!
◦ Digital systems just use bits.

Time for a **Pop Quiz**!

Ok, what is the bit pattern?

Seriously?

Maybe you remember a few of them?

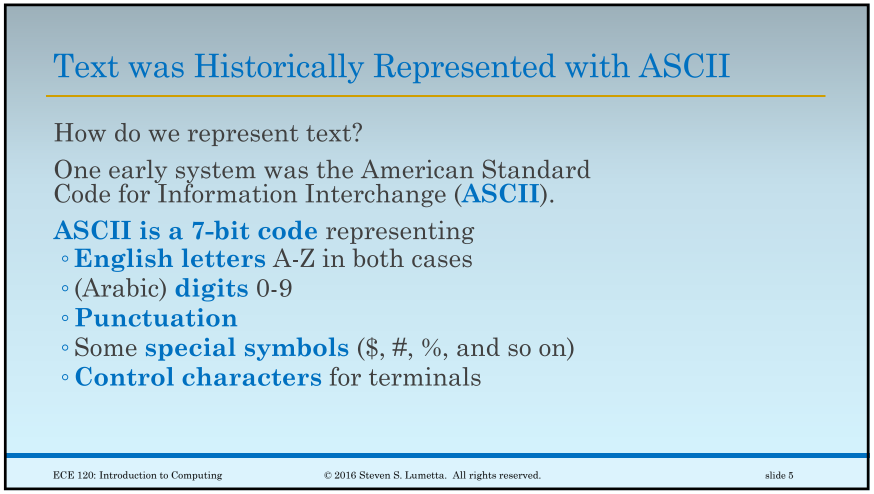
What if this is were an exam question?

Sigh.

Ok, it was **00010011010101100111**.

In hex, that's **x13567** (P&P/LC-3 hex notation—otherwise, 13567 is probably decimal!).

Can you remember that? **Please?**

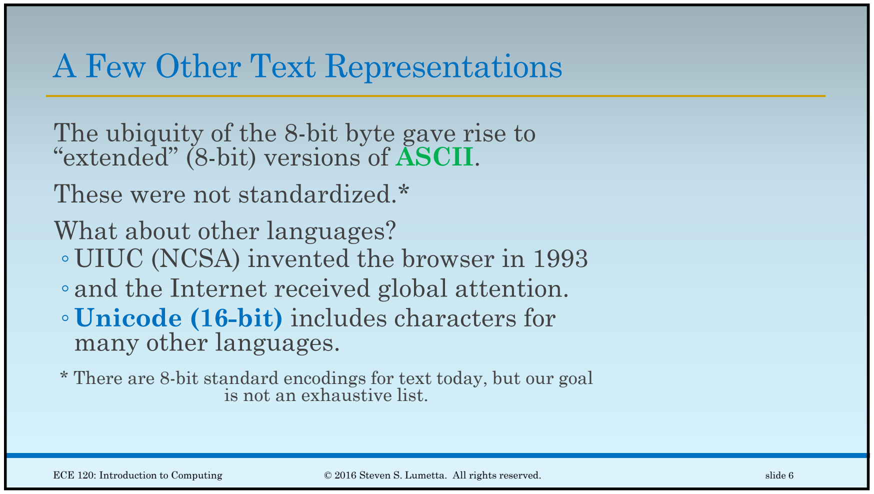


Text was Historically Represented with ASCII

How do we represent text?

One early system was the American Standard Code for Information Interchange (**ASCII**).

ASCII is a 7-bit code representing
- **English letters** A-Z in both cases
- (Arabic) **digits** 0-9
- **Punctuation**
- Some **special symbols** ($, #, %, and so on)
- **Control characters** for terminals



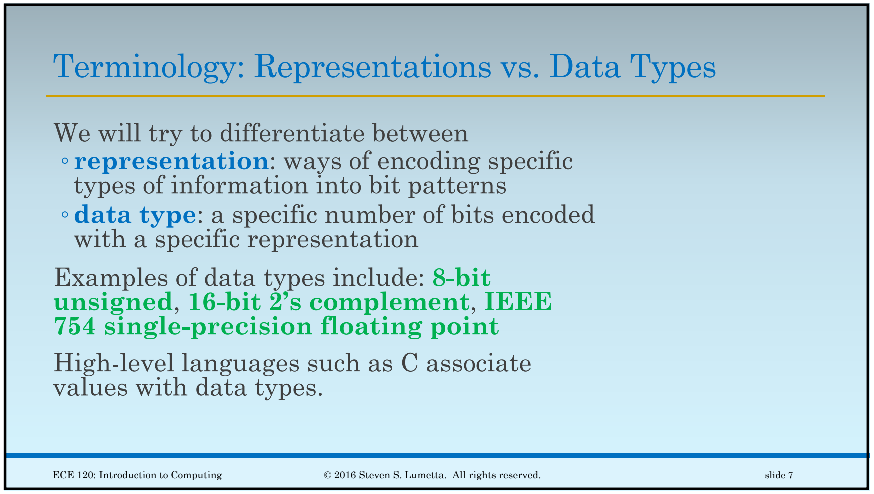
A Few Other Text Representations

The ubiquity of the 8-bit byte gave rise to “extended” (8-bit) versions of **ASCII**.

These were not standardized.*

What about other languages?
- UIUC (NCSA) invented the browser in 1993
- and the Internet received global attention.
- **Unicode (16-bit)** includes characters for many other languages.

* There are 8-bit standard encodings for text today, but our goal is not an exhaustive list.



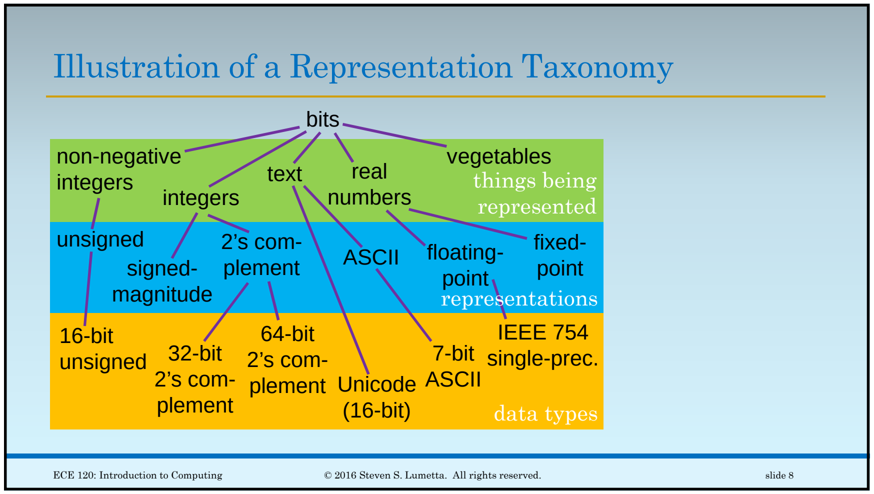
Terminology: Representations vs. Data Types

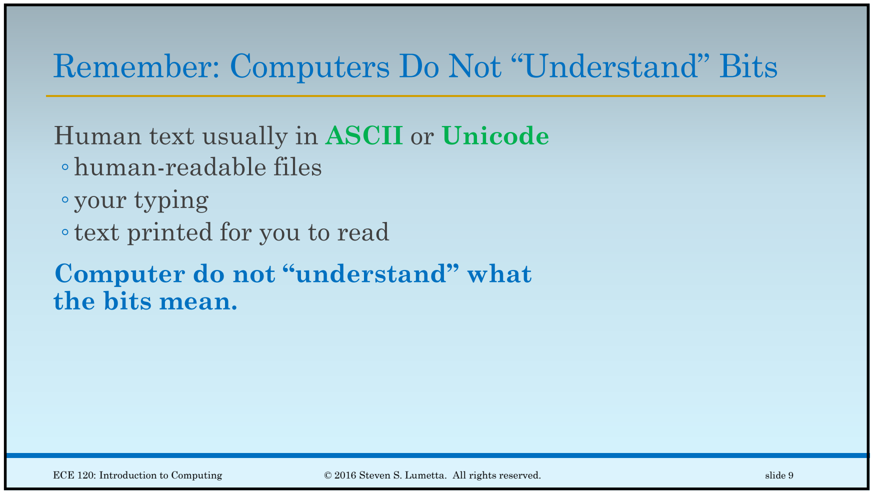
We will try to differentiate between
- **representation**: ways of encoding specific types of information into bit patterns
- **data type**: a specific number of bits encoded with a specific representation

Examples of data types include: **8-bit unsigned**, **16-bit 2’s complement**, **IEEE 754 single-precision floating point**

High-level languages such as C associate values with data types.

Illustration of a Representation Taxonomy



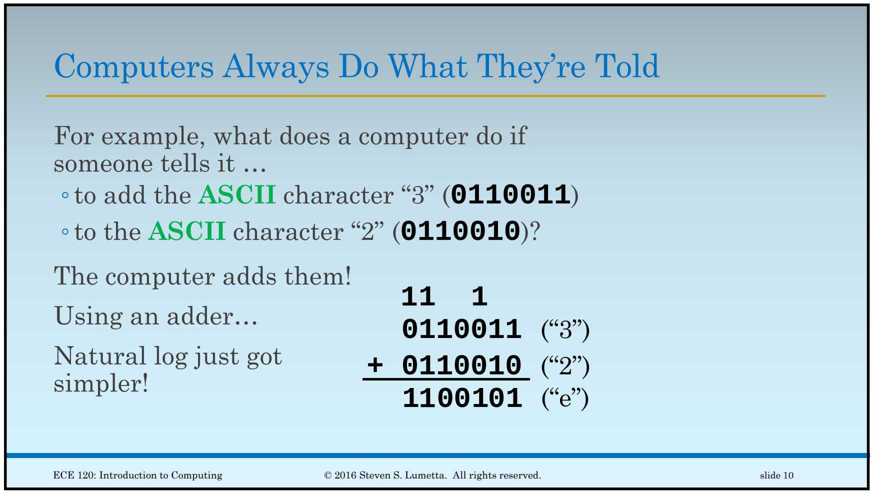


Remember: Computers Do Not "Understand" Bits

Human text usually in **ASCII** or **Unicode**

- human-readable files
- your typing
- text printed for you to read

Computer do not "understand" what the bits mean.



Computers Always Do What They're Told

For example, what does a computer do if someone tells it …

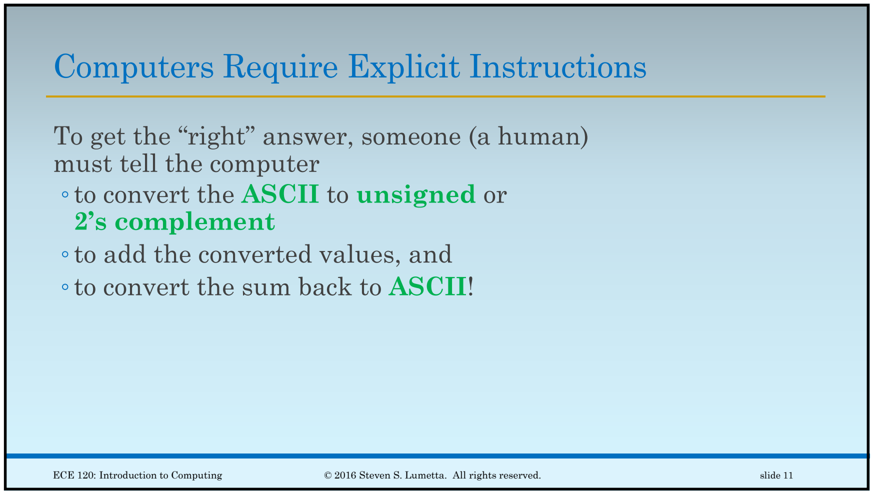
- to add the **ASCII** character "3" (**0110011**)
- to the **ASCII** character "2" (**0110010**)?

The computer adds them!

Using an adder…

Natural log just got simpler!

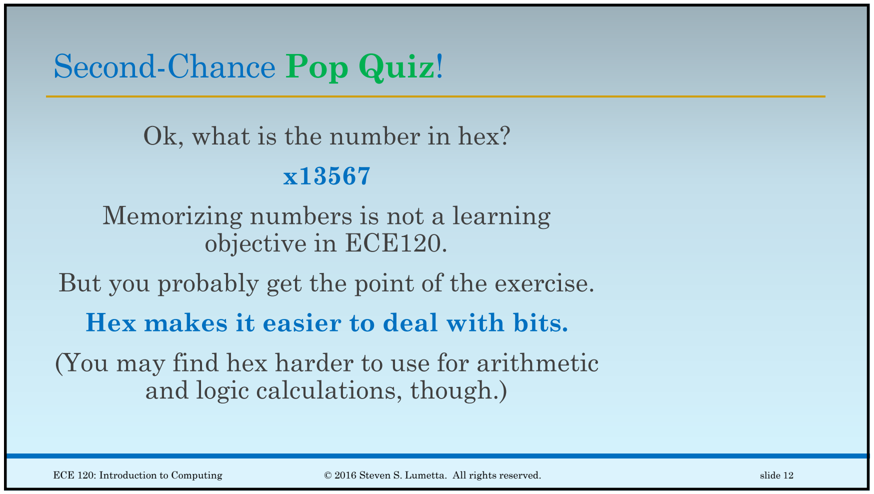
```
   11  1
   0110011  ("3")
 + 0110010  ("2")
 ---------
   1100101  ("e")
```



Computers Require Explicit Instructions

To get the "right" answer, someone (a human) must tell the computer

- to convert the **ASCII** to **unsigned** or **2's complement**
- to add the converted values, and
- to convert the sum back to **ASCII**!



Second-Chance **Pop Quiz!**

Ok, what is the number in hex?

x13567

Memorizing numbers is not a learning objective in ECE120.

But you probably get the point of the exercise.

Hex makes it easier to deal with bits.

(You may find hex harder to use for arithmetic and logic calculations, though.)