CSE 113 B
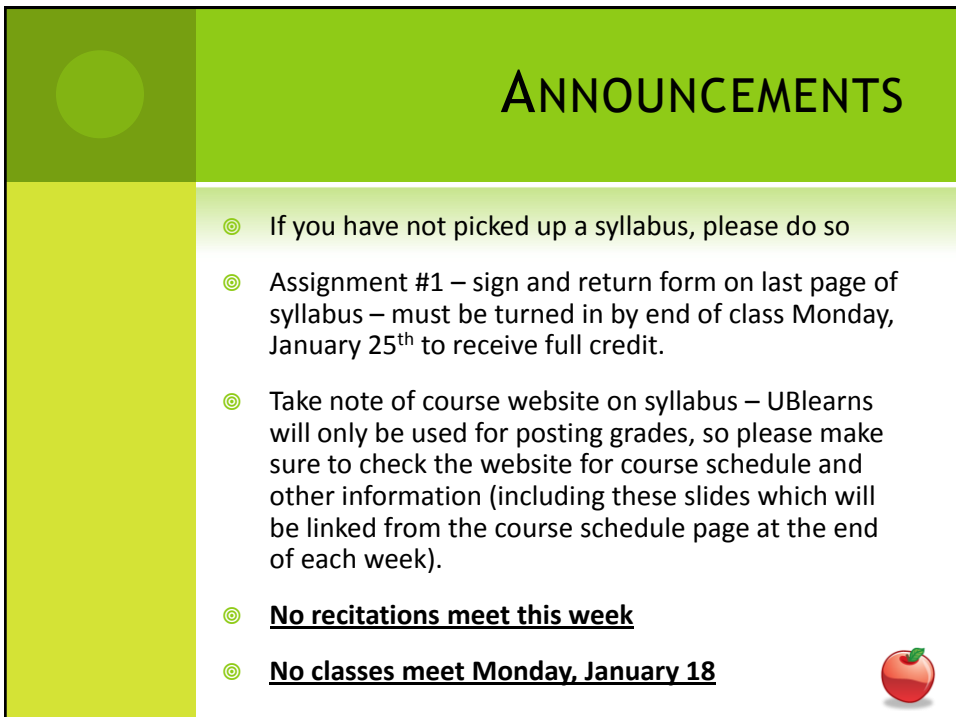
January 11 – 15, 2010

ANNOUNCEMENTS

◎ If you have not picked up a syllabus, please do so

◎ Assignment #1 – sign and return form on last page of syllabus – must be turned in by end of class Monday, January 25th to receive full credit.

◎ Take note of course website on syllabus – UBlearns will only be used for posting grades, so please make sure to check the website for course schedule and other information (including these slides which will be linked from the course schedule page at the end of each week).

◎ **No recitations meet this week**

◎ **No classes meet Monday, January 18**

WHAT DOES A COMPUTER UNDERSTAND?

◎ 0's and 1's (zeros and ones)

BITS AND BIT STRINGS

◎ The 0 or 1 is called a binary digit (bit).

◎ A sequence of bits is called a bit string.

◎ 0100101 is a bit string

⊙ What does it mean/represent?

37

91

%

turn the computer off

INTERPRETING BIT PATTERNS

- Binary (non-negative numbers)
- Two's complement (integers)
- IEEE 754 (approximate floating point numbers)
- ASCII/EBCDIC/Unicode (characters)

CONVERSION FROM DECIMAL TO BINARY

37

tens place

ones place

$3 \times 10 + 7 \times 1$

$3 \times 10^1 + 7 \times 10^0$

base

bitstring: 111

$1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

$1 \times 4 + 1 \times 2 + 1 \times 1$

$4 + 2 + 1 =$ "seven"

ENCODING MACHINE INSTRUCTIONS

- ◎ Use bits to encode those as well
- ◎ When we want the machine to follow those instructions:
  - ⊙ Fetch — *get the instruction from memory*
  - ⊙ Decode — *turn it into 1's & 0's*
  - ⊙ Execute — *do the instruction*

ASSEMBLY LANGUAGE

- ◎ ADD r1 r2
- ◎ STOR r2 r1
- ◎ SUB r3 r1

instruction *which registers* *places to store data*

HIGH-LEVEL LANGUAGES

◎ Step closer to natural language from machine language.

TOOLS

◎ Editor *– place to type your program's text*

◎ Compiler *– translator*

◎ Execution Environment
↳ to help us run the program

OUR LANGUAGE: JAVA

◎ High level programming language

◎ Object-oriented