Representing Java Values

Recitation 2

Decimal Integers

$$d_{n-1}d_{n-2}...d_0$$

$$\sum_{i=0}^{n-1} d_i(10^i)$$

131 = (100 * 1) + (10 * 3) + 1

= (10^2 * 1) + (10^1 * 3) + (10^0 * 1)

Binary Integers

$$b_{n-1}b_{n-2}...b_0$$

$$\sum_{i=0}^{n-1} b_i(2^i)$$

131 = (128 * 1) + (2 * 1) + 1

= (2^7 * 1) + (2^1 * 1) + (2^0 * 1)

= 10000011_2

Exercise

Convert 7 and 11 to binary

Two's Complement

Definition: It's a signed n-bit representation where the most significant bit stands for -2^{n-1} instead of 2^{n-1}

$$b_{n-1}b_{n-2}...b_0$$

$$-b_{n-1}(2^{n-1}) + \sum_{i=0}^{n-2} b_i(2^i)$$

Two's Complement Example

$$11000001_2 = -128 + 64 + 1$$

$$= -63$$

$$11111111_2 = -128 + 64 + 32 + 16 + 8 + 4 + 2 + 1$$

$$= -1$$

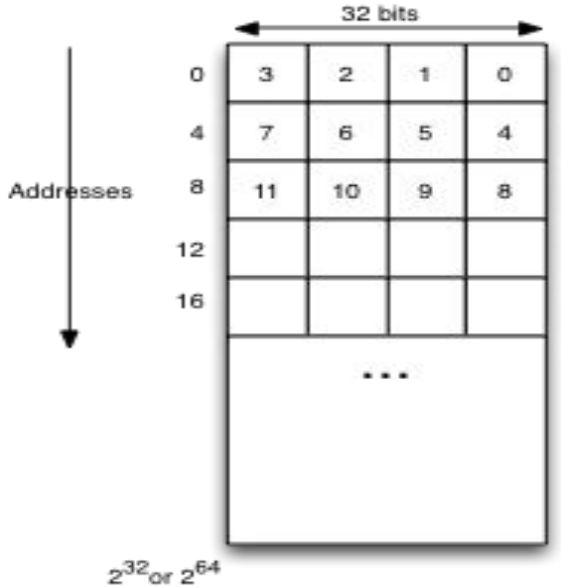
Exercise

Convert 3 and -3 to Two’s complement

(Assume that the numbers are represented using 4 bits)

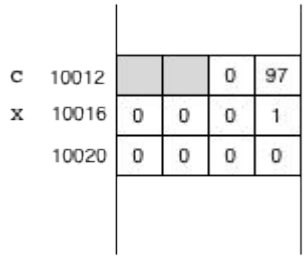
Memory

- Computer memory is a grid with a bit stored in every cell
- Each address names a group of 8 bits (a **byte**)
- Computer memory can read the four bytes beginning at an address. These four bytes are called a **word**



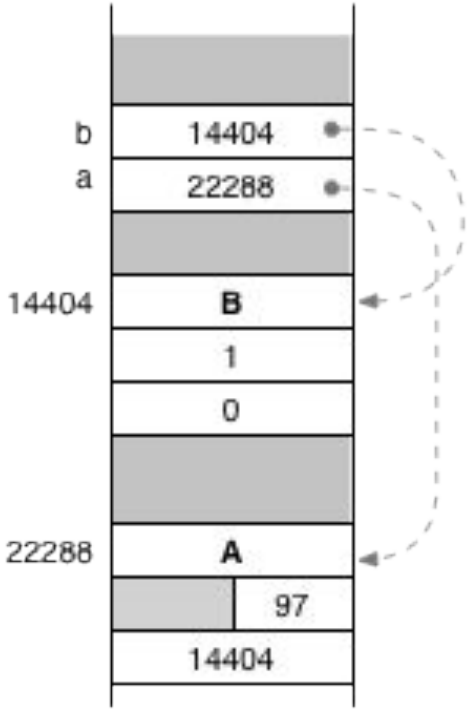
Variables

- Variables are assigned an integer number of words (even if it needs less space)
- char c = ‘a’; long x = 1;



Objects

```
class A {
    char c;
    B y;
}
class B {
    long z;
}
...
B b = new B();
b.z = 1;
A a = new A();
a.c = 'a';
a.y = b;
```



Decimal Scientific Notation

- Consists of an integer n between 0 and 9
- A rational number between 0 and 1
- A factor of 10 raised to an integer power

7.234 * 10^(-5)

Floating Point Representation

Consists of the following components:

- sign (s): 0 or 1
- exponent (exp): integer
- mantissa (m): sequence of binary digits such that $1 \leq 1.m < 2$

$(-1)^{s} \times 2^{exp} \times (1.m)$

Floating Point Pitfalls!!

- Due to precision related errors check for closeness instead of equality
  - f1 == f2
  - $|f1 - f2| < \varepsilon$
- Floating point addition is not commutative
  - $1 + 10^{-40} - 1 \neq 1 - 1 + 10^{-40}$