Winter 2016
COMP-250: Introduction to Computer Science

Lecture 4, January 21, 2016

Fractional Numbers

26.375

$=(11010.\_\_\_\_)_2$

0.375

$=1/4+1/8$

$=(0.011)_2$

26.375

$=(11010.011)_2$

Fractional Numbers

$$26.375 = 2*10^{1} + 6*10^{0} + 3*10^{-1} + 7*10^{-2} + 5*10^{-3}$$

$$0.375 = 3*10^{-1} + 7*10^{-2} + 5*10^{-3}$$

Fractional Numbers

$$0.375*8/8$$
$$=3/8$$
$$=(11)_2/(1000)_2$$
$$=(1.1)_2/(100)_2$$
$$=(0.11)_2/(10)_2$$
$$=(0.011)_2/(1)_2$$
$$=(0.011)_2$$

Fractional Numbers

$$(1/3)_{10} = (0.\underline{3})_{10}$$

$$= (1/11)_{2} = (0.\underline{01})_{2}$$

Fractional Numbers

1/5

$=(0.2)_{10}$

$=(1/101)_2$

```
1.          /101
 .101        0.0011
-----
 .011
 .0101
-----
 .0001
```

Check:

$0.\underline{0011}$

$=3*\sum_{i\geq 1}16^{-i}$

$=3*1/15$

$=1/5$

$=(0.\underline{0011})_2$

Fractional Numbers

$$19.247$$
$$=1*10^{1}$$
$$+9*10^{0}$$
$$+2*10^{-1}$$
$$+4*10^{-2}$$
$$+7*10^{-3}$$

$$(19)_{10}=(10011)_{2}$$

$$0.247$$
$$=2*10^{-1}$$
$$+4*10^{-2}$$
$$+7*10^{-3}$$

Fractional Numbers

$(0.247)_{10}$
$=(11110111/1111101000)_2$

```
1110111.          /1111101000
 1111101.000       0.0011…
-------------
 1111010.000
  111110.1000
-------------
    11101.1
       …
```

Fractional Numbers

0.247

$=0.247*64/64$

$=15.808/64$

$=(1111.\overline{——})_2/(1000000)_2$

$=(0.001111\overline{——})_2$

$\approx(0.001111)_2$ lower bound

$\approx(0.01)_2$ upper bound

Fractional Numbers

$(19.247)_{10} > (10011.001111)_2$ lower bound

$(19.247)_{10} < (10011.01)_2$ upper bound

Fractional Numbers

0.247
=0.494/2
=0.988/4
=1.976/8
=3.952/16
=7.904/32
=15.808/64
=31.616/128
=63.232/256
=126.464/512
=252.928/1024
=505.856/2048

=1011.712/2^{12}
=2023.424/2^{13}
=4046.848/2^{14}
=8093.696/2^{15}
=16187.392/2^{16}
=32374.784/2^{17}
=64749.568/2^{18}
=129499.136/2^{19}
=258998.272/2^{20}
=517996.544/2^{21}
=1035993.088/2^{22}
=2071986.176/2^{23}

Fractional Numbers

=4143972.352/2^{24} =———.792/2^{36}
=8287944.704/2^{25} =———.584/2^{37}
=16575889.408/2^{26} =———.168/2^{38}
=33151778.816/2^{27} =———.336/2^{39}
=66303557.632/2^{28} =———.672/2^{40}
=132607115.264/2^{29} =———.344/2^{41}
=265214230.528/2^{30} =———.688/2^{42}
=530428461.056/2^{31} =———.376/2^{43}
=1060856922.112/2^{32} =———.752/2^{44}
=2121713844.224/2^{33} =———.504/2^{45}
=4243427688.448/2^{34} =———.008/2^{46}
=8486855376.896/2^{35} =———.016/2^{47}

Fractional Numbers

=———.032/2^{48} =———.072/2^{60} =———.912/2^{72}
=———.064/2^{49} =———.144/2^{61} =———.824/2^{73}
=———.128/2^{50} =———.288/2^{62} =———.648/2^{74}
=———.256/2^{51} =———.576/2^{63} =———.296/2^{75}
=———.512/2^{52} =———.152/2^{64} =———.592/2^{76}
=———.024/2^{53} =———.304/2^{65} =———.184/2^{77}
=———.048/2^{54} =———.608/2^{66} =———.368/2^{78}
=———.096/2^{55} =———.216/2^{67} =———.736/2^{79}
=———.192/2^{56} =———.432/2^{68} =———.472/2^{80}
=———.384/2^{57} =———.864/2^{69} =———.944/2^{81}
=———.768/2^{58} =———.728/2^{70} =———.888/2^{82}
=———.536/2^{59} =———.456/2^{71} =———.776/2^{83}

Fractional Numbers

=———.552/2^{84}
=———.104/2^{85}
=———.208/2^{86}
=———.416/2^{87}
=———.832/2^{88}
=———.664/2^{89}
=———.328/2^{90}
=———.656/2^{91}
=———.312/2^{92}
=———.624/2^{93}
=———.248/2^{94}
=———.496/2^{95}
=———.992/2^{96}
=———.984/2^{97}
=———.968/2^{98}
=———.936/2^{99}
=———.872/2^{100}
=———.744/2^{101}
=———.488/2^{102}
=—W—.976/2^{103}
=2 504 877 586 050
981 297 357 485
533 822.976/2^{103}

$0.247 = (11110111/1111101000)_2$

$=1.976/8$

0.976=8*0.247-1

$= \text{—W—}.976/2^{103}$

$= (\text{—W—}+8*0.247-1)/2^{103}$

$= (\text{—W—}-1)/2^{103} + 0.247/2^{100}$

$= (\text{—W—}-1)/2^{103} + (\text{—W—}-1)/2^{203} + 0.247/2^{200}$

$= (\text{—W—}-1)/8 \sum_{i\geq 1} 2^{-100*i}$

=(0.00111111001110110110010
001011010000111001010110000001
000001100010010011011101001011
11000110101001111110)$_2$

Fractional Numbers

$(19.247)_{10}$ =

$(10011.001111110011101101100100$

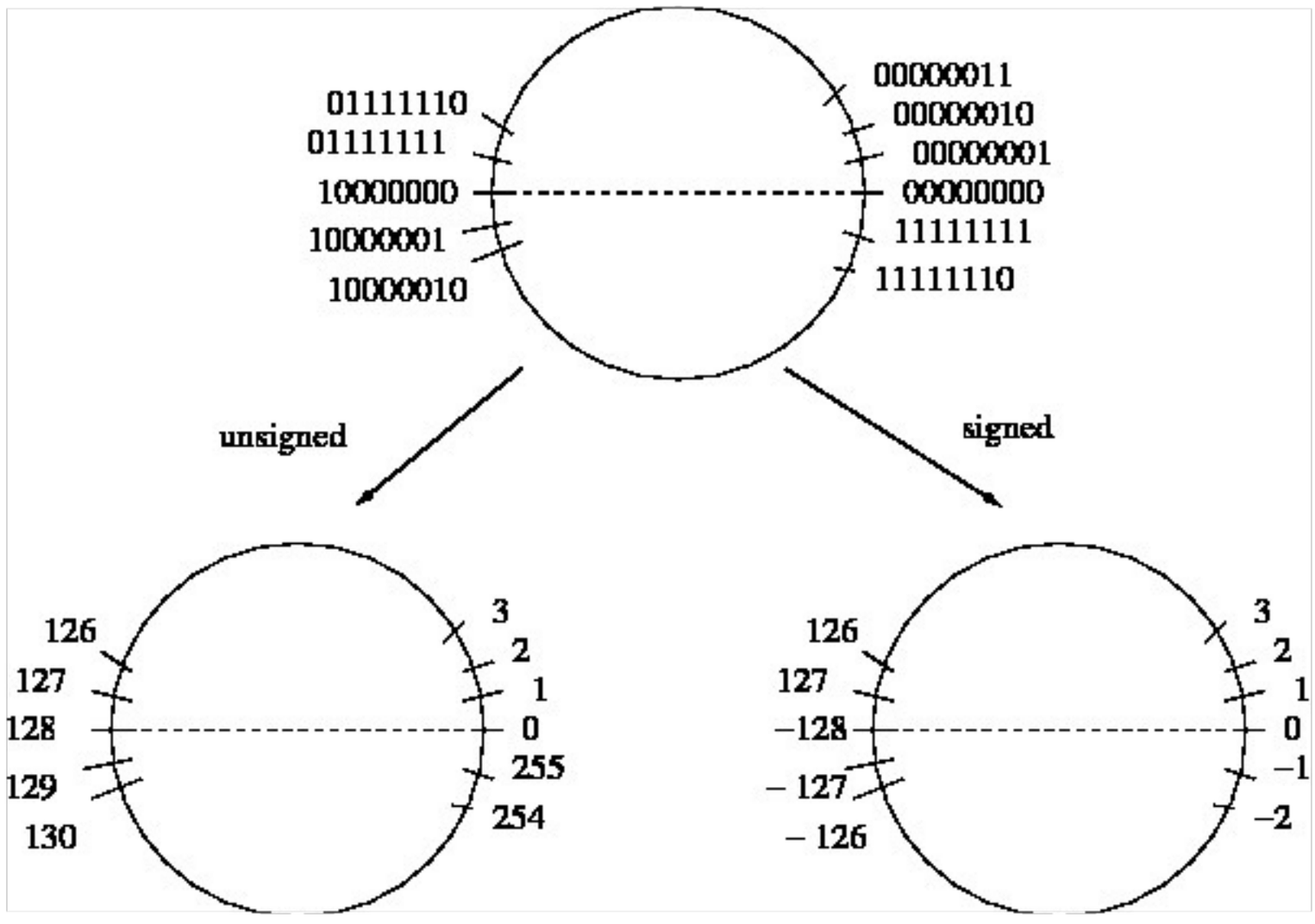
$0010110100001110010101100000001$

$0000011000100100110110100101111$

$1000110101001111110)_2$

More Binary Representation

More Binary Representation



Representation

binary	signed	unsigned
00000000	0	0
00000001	1	1
:	:	:
01111111	127	127
10000000	-128	128
10000001	-127	129
:	:	:
11111111	-1	255

Representation

binary	signed	unsigned
0000000000000000	0	0
0000000000000001	1	1
:	:	:
0000000001111111	127	127
0000000010000000	128	128
0000000010000001	129	129
:	:	
0111111111111111	$2^{15} - 1$	$2^{15} - 1$
1000000000000000	-2^{15}	2^{15}
1000000000000001	$-2^{15} + 1$	$2^{15} + 1$
:	:	
1111111101111111	-129	$2^{16} - 129$
1111111110000000	-128	$2^{16} - 128$
1111111110000001	-127	$2^{16} - 127$
:	:	
1111111111111111	-1	$2^{16} - 1$

Fun with Java !!!

```
for (short s = 32767; s < 32768; s++)
    System.out.println(s);
```

```
32767
-32768
-32767
…
-1
0
1
```

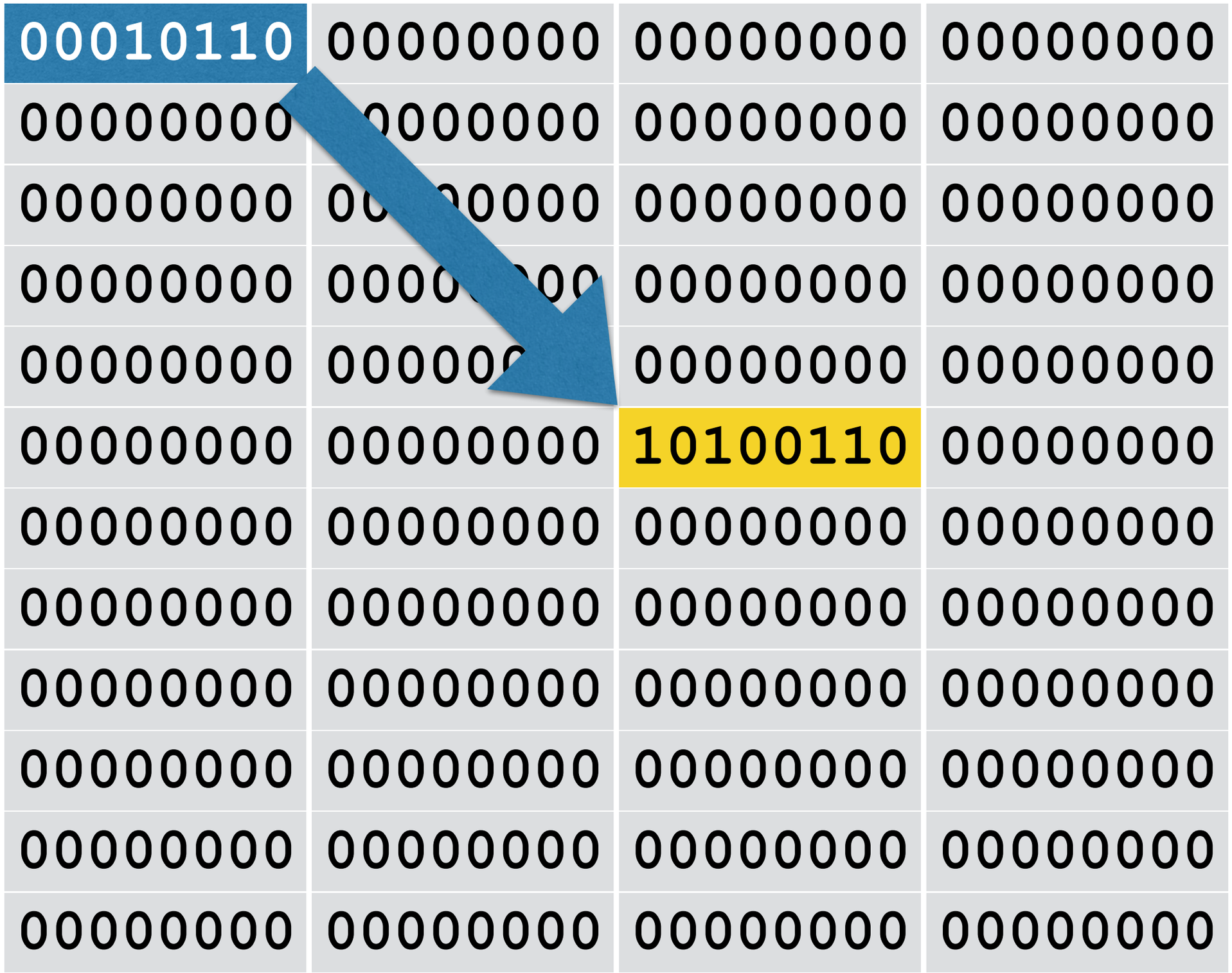
A Byte

10100110

A Byte

00010110	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	10100110	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000

An address



A (32-bit) address

00000000	00000000	00000000	00010110
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	10100110	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000

A (64-bit) address

00000000	00000000	00000000	00000000
00000000	00000000	00000000	00010110
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	10100110	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000

Storage sizes

- 2^{10} bytes $= 1$ kilobyte (1 KB) $\approx 10^3$ bytes (one thousand)
- 2^{20} bytes $= 1$ megabyte (1 MB) $\approx 10^6$ bytes (one million)
- 2^{30} bytes $= 1$ gigabyte (1 GB) $\approx 10^9$ bytes (one billion)
- 2^{40} bytes $= 1$ terabyte (1 TB) $\approx 10^{12}$ bytes (one trillion)
- 2^{50} bytes $= 1$ petabyte (1 PB) $\approx 10^{15}$ bytes
- 2^{60} bytes $= 1$ exabyte (1 EB) $\approx 10^{18}$ bytes

} **Big Data**

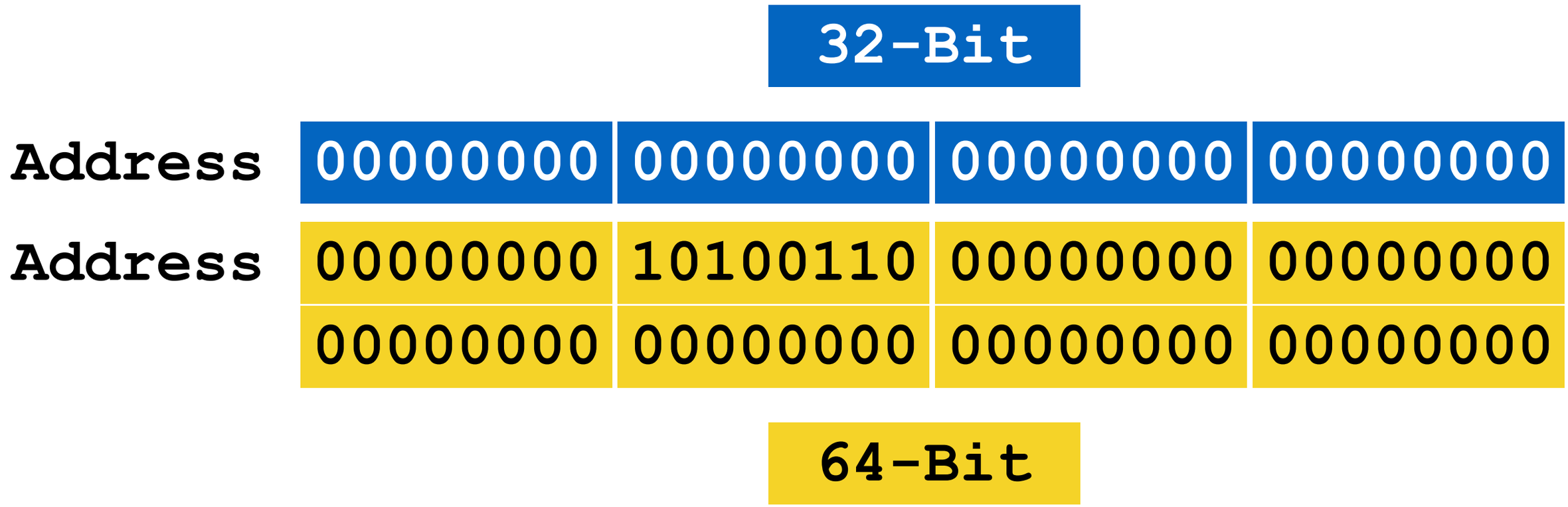
Java Primitive Types

Boolean	00000000	00000000	00000000	00000000
Byte	00000000	00000000	00000000	00000000
Char	00000000	00000000	00000000	00000000
Short	00000000	00000000	00000000	00000000
Int	00000000	00000000	00000000	00000000
Long	00000000	10100110	00000000	00000000
	00000000	00000000	00000000	00000000
Float	00000000	00000000	00000000	00000000
Double	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000

(32-bit) addresses

00000000	00000000	00000000	00010010
00000000	00000000	00000000	00101010
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	10100110	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000

Java Reference Types



byte[] a;

a:			
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000

a=new byte[3];

a:			
00000000	00000000	00000000	00010010
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000

a[0]=166;

a:			
00000000	00000000	00000000	00010010
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	10100110	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000

int[ ] b;

a:	00000000	00000000	00000000	00010010
b:	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000
	00000000	00000000	10100110	00000000
	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000

b=new int[2];


a:
b:
00000000 00000000 00000000 00010010
00000000 00000000 00000000 00101010
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 10100110 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000

b[i]=-i;

a:	00000000	00000000	00000000	00010010
b:	00000000	00000000	00000000	00101010
	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000
	00000000	00000000	10100110	00000000
	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000
	00000000	00000000	11111111	11111111
	11111111	11111111	00000000	00000000

Winter 2016
COMP-250: Introduction to Computer Science

Lecture 4, January 21, 2016