Java Programming AP Edition
U1C2 Elementary Programming

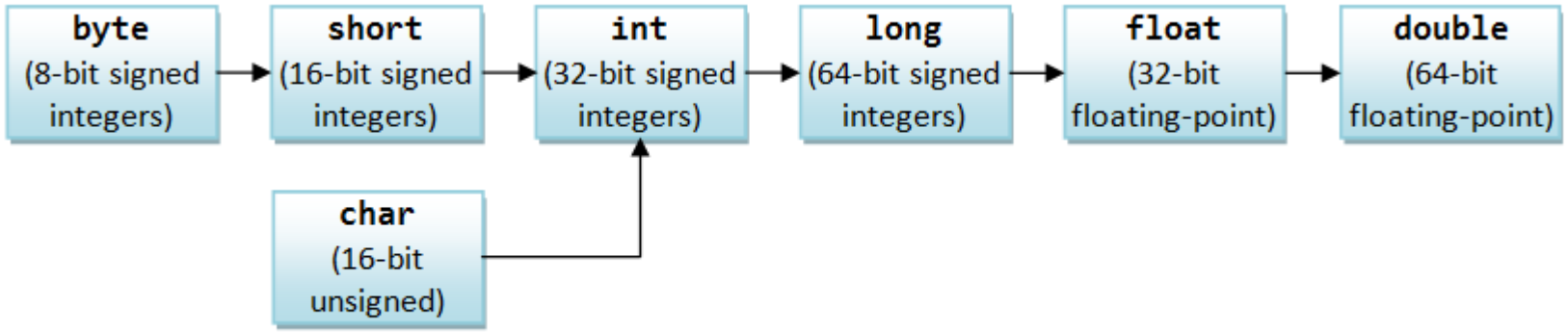
DATA TYPES (INT AND DOUBLE)

ERIC Y. CHOU, PH.D. IEEE SENIOR MEMBER

Numeric Data Types and Operations

Java has six numeric types for integers and floating-point numbers with operators +, -, *, . and %

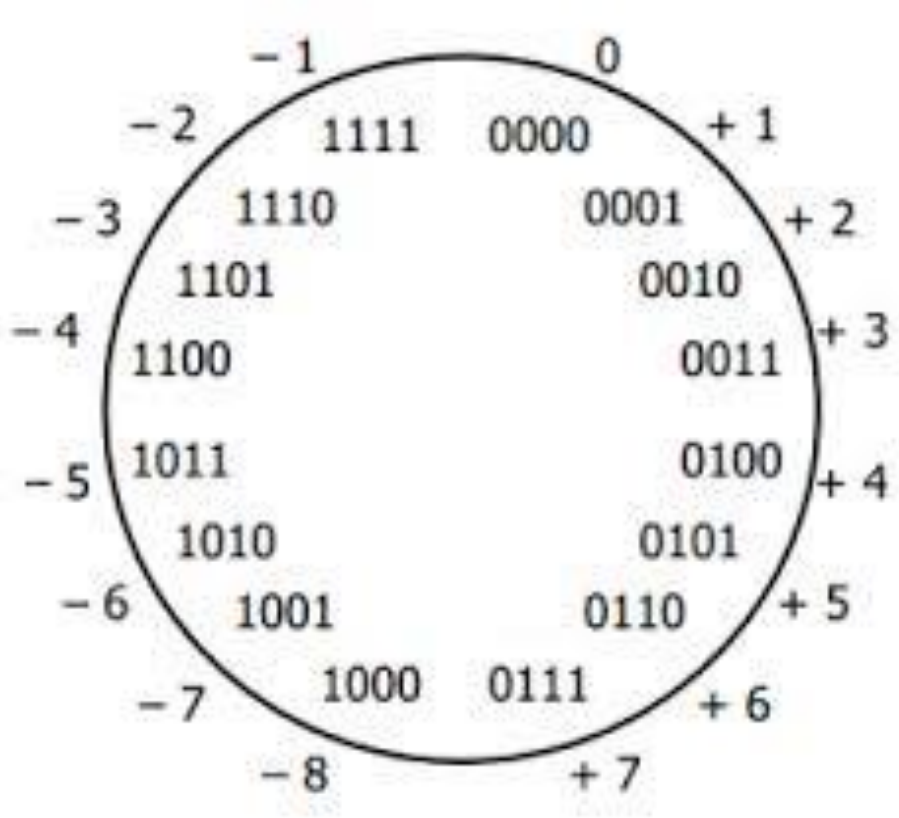
Name	Data	Range	Default Value	Size
byte	signed integer	[-128, 127]	0	8 bits
short	signed integer	[-32768, 32767]	0	16 bits
int	signed integer	[-2147483648, 2147483647]	0	32 bits
long	signed integer	[-9223372036854775808, 9223372036854775807]	0	64 bits
float	floating-point	**MIN:** ±1.4E-45 **MAX:** ±3.4028235E+38	0.0	32 bits
double	floating-point	**MIN:** ±4.9E-324 **MAX:** ±1.79769313486231570E+308	0.0	64 bits
char	Unicode	['\u0000', '\uFFFF']	'\u0000'	16 bits
boolean	logical value	{false, true}	false	≥ 1 bit



Orders of Implicit Type-Casting for Primitives

Integer.MIN_VALUE
Integer.MAX_VALUE

Two’s Complement



Negative number is represented as two’s complement.

For byte number’s (8 bits):

$-X = (2^8 - 1) - X + 1;$

$X + (-X) = X + (2^8 - X) = 2^8 = 0\ ;$

eg.

A = 0100 -> A’s One’s Complement = 1011 ->

A’s Two’s Complement -> 1100

The number 2^8 is a overflow for the byte format, because unsigned byte number range

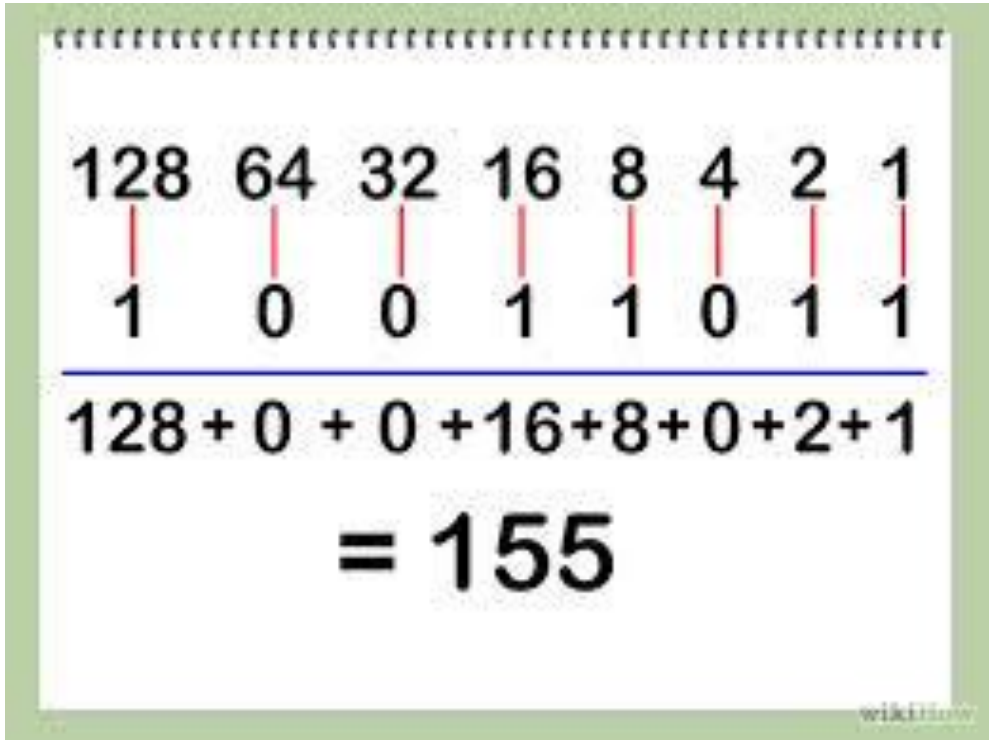
from 0 to $2^8 - 1$ = 11111111.

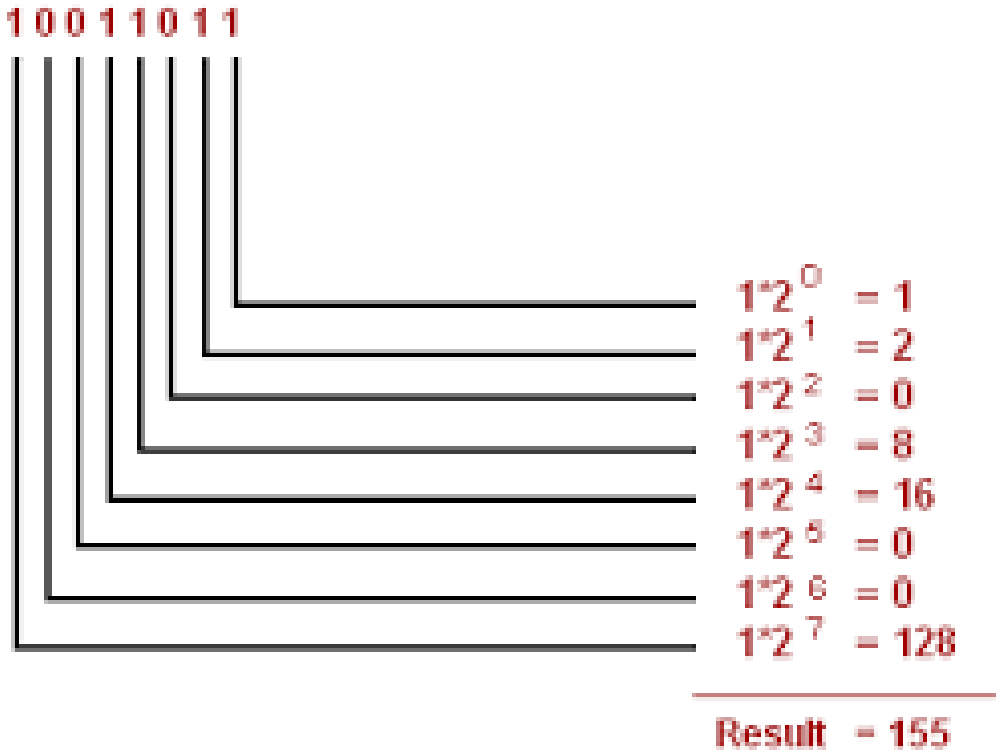
Therefore, this method can work for computer.

Finding 2's Complement

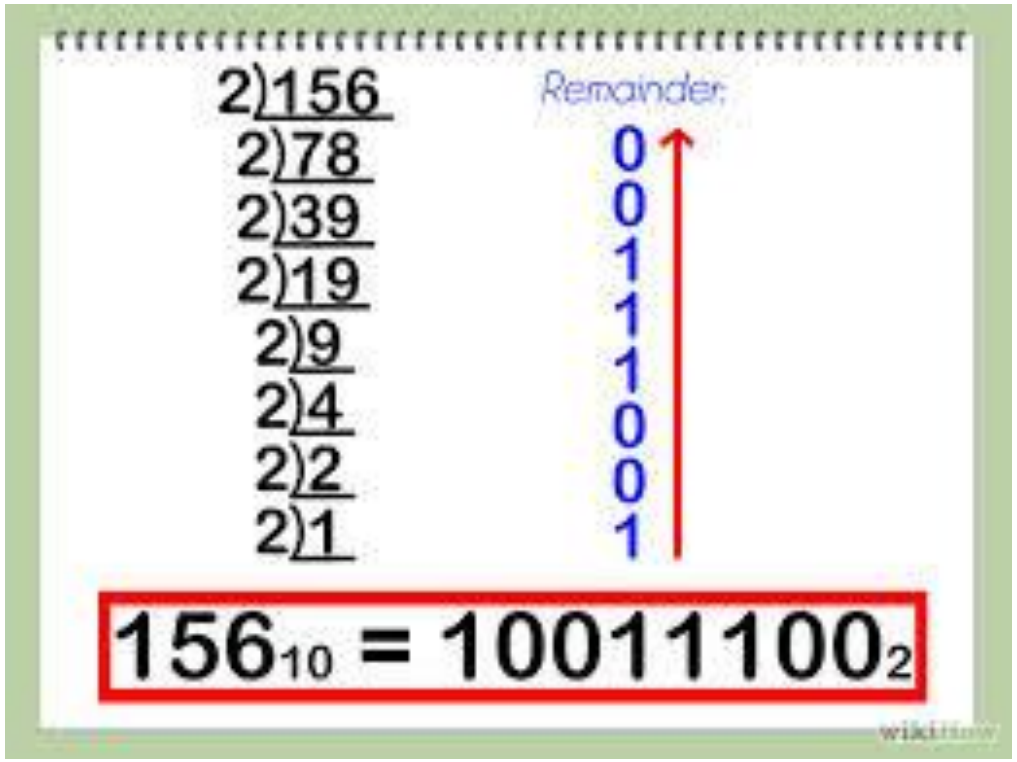
	- 128	64	32	16	8	4	2	1	
X	0	1	0	0	1	1	1	1	**Number : 79 decimal**
$(2^8-1)-X$	1	0	1	1	0	0	0	0	**Flip the bits**
	0	0	0	0	0	0	0	1	**Add 1**
$NegX = (2^8-1)-X+1$	1	0	1	1	0	0	0	1	**Number: -79 in 2's Complement format**

Binary/Decimal Conversion



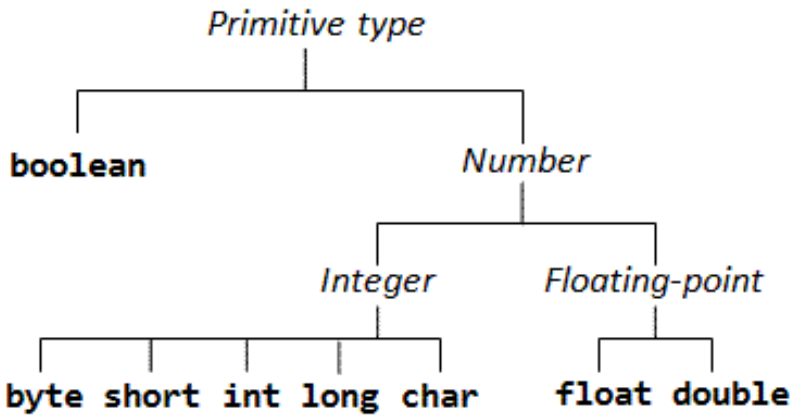


Decimal to Binary



Divider	Dividend	Remainder
2	202	0
2	101	1
2	50	0
2	25	1
2	12	0
2	6	0
2	3	1
		1

Java's special number rules
(different from other languages)



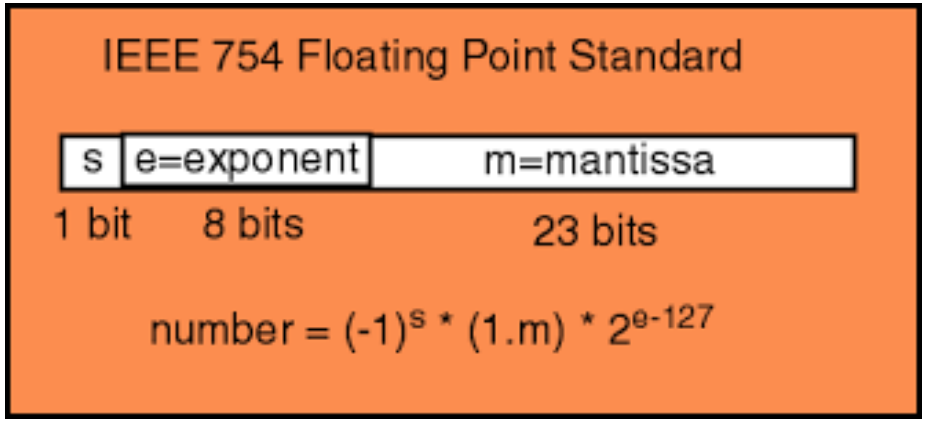
Java doesn't have unsigned number primitives.

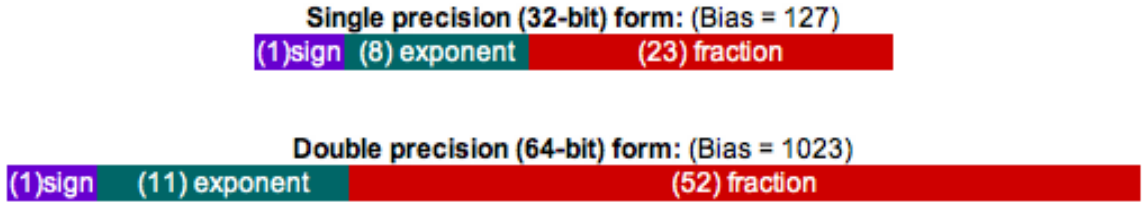
unsigned number is seldom used.

If you need to use unsigned number, use **char** data type instead. Because char does not follow the number operation rules while **char** can still operate the **bit-wise** operations.

Java's char is 16 bit. (supporting Unicode: UTF-16)

IEEE 754 binary floating point representation. (Java's Float Standard)





Named Constants

A named constant is an identifier that represents a permanent value.

Syntax:

final <datatype> CONSTANTNAME = <value> ;

The word final is a Java reserved keyword for declaring a constant.

A constant in Java (or most of other language) is usually in all **UPPERCASE**.

Benefits for using constants:

- (1) you don't have to repeatedly type the same value over over again if it is used multiple times;
- (2) if you have to change the constant value, you need to change it only in a single location in the source code; and
- (3) a descriptive name for a constant makes the program easier to read.

Named Constants


```
final datatype CONSTANTNAME = VALUE;


final double PI = 3.14159;
final int SIZE = 3;
```