

Lesson Notes: 03/02/2026

Numbers

A primitive data type specifies the size and type of variable values, and it has no additional methods

There are 6 numeric types in Java:

Type	Size (bits)	Wrapper Class	Number Range
byte	8	Byte	-128 to 127
short	16	Short	-32,768 to 32,767
int	32	Integer	-2,147,483,648 to 2,147,483,647
long	64	Long	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	32	Float	$\pm 1.5 \times 10^{-45}$ to $\pm 3.4 \times 10^{38}$ ($\approx 6-9$ digits precision)
double	64	Double	$\pm 5.0 \times 10^{-324}$ to $\pm 1.7 \times 10^{308}$ ($\approx 15-17$ digits precision)

The table shows the range of values that can be held (all are signed - can hold positive and negative numbers).

- Byte, short, int and long are integers
- Float and double are used for floating point types

The wrapper classes can be used instead of primitive types but take up more memory and are reference types:

<https://docs.oracle.com/en/java/javase/25/docs/api/java.base/java/lang/Float.html>.

Operators

Exercise

Add a new java class file called Methods to the same package as your Operators class.

Write static methods (in the Methods class) to convert a given number of pounds to pounds and stones, where 1 stone is equal to 14 pounds. So 156 pounds is equal to 11 stones, 2 pounds.

A method can only return 1 value, so you will need to write and test two methods with the signatures:

- `public static int getStones(int totalPounds)` which returns the whole number of stones
- `public static int getPounds(int totalPounds)` which returns the number of pounds left over

Bonus:

```
int j = 5, k = 3, m = 4;
```

```
m += +j++ + ++k;
```

Relational Operators

Operator	Meaning
<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to
==	equal to
!=	not equal to

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/op2.html>

Logical Operators

Logical AND (&&)

Operand a	Operand b	a && b
true	true	true
true	false	false
false	true	false
false	false	false

Logical OR (||)

Operand a	Operand b	a b
true	true	true
true	false	true
false	true	true
false	false	false

Logical NOT (!)

Operand a	! a
true	false
false	true