

Operator Precedence:

What?

- It determines the order in which operators are evaluated.

Expression: $3 + 5 * 2$

OP tells us that multiplication has higher precedence, so we multiply $5 * 2$ first, and then add 3, giving us a result of 13.

OP chart for JAVA

Precedence	Operator	Description	Associativity
1 (Highest)	()	Parenthesis	Left to Right
2	[]	Array subscript	Left to Right
	.	Member access (e.g object.method)	Left to Right
3	++, --	Postfix increment/decrement (e.g a++, b--)	Left to Right
4	++, --	Prefix increment/	Right to left

		decrement (e. g. ++a, --b)	
	+, -	Unary plus and minus (-a, +y)	Right to left
	!, ~	Logical NOT, Bitwise not	Right to left
5	new	Object creation (e. g new MyClass())	Right to left
6	*, /, %	Multiplication , division and modulus	Left to right
7	+, -	Addition and Subtraction	Left to right
8	>>, <<	Bitwise left shift, right shift	Left to Right
9	<, <=, >, >=	Relational Operators	Left to Right
10	==, !=	Equality Operators	Left to right
11	&	Bitwise AND	Left to right
12			
13	&&	Logical AND	Left to right
14	?:	Ternary Conditional	Right to left

15	=, +=, -= *=/=, * =	Assignment Operators	Right to left
----	------------------------------	-------------------------	------------------

IMP:

1. Parenthesis (): have the highest precedence
2. Unary operators have higher precedence than arithmetic
3. Arithmetic Operators (*, /, %, +, -) follows the standard mathematical order of operations
4. Relational and equality operators are evaluated after arithmetic operators
5. Logical AND (&&) has higher precedence than logical OR (||)

Example1:

```
int result = 10+3*2;
```

Step1: $3*2 = 6$

Step2: $10+6$

Result=16

Example2:

```
int result = (10+3)*2;
```

Step1: Parenthesis force the addition because it has highest(1st) precedence. $10+3=13$

```
int result =13*2;
```

Step2: Multiplication: $13*2 = 26$

Final Answer: 26

Example3:

```
Boolean result= 5>3 && 10<20;
```

Step1:

Relational Operators($5>3$ and $10<20$)are evaluated first.

Results are True and Trues

Step2:

```
Boolean result= True && True;
```

Logical AND(true && true)is evaluated.

Final Answer = True

Example4:

`Int result = 10+2*3-6/2;`

Step1: Multiplication($2*3$) and division($6/2$) are evaluated first.

Step2:

`Int result = 10+6-3;`

Addition and subtraction are evaluated from left to right.

First: $10+6 = 16$

Then: $16-3=13$

Final Answer: 13

TASK:

`int result= 8/2*(2+2);`

`boolean result=15>10 && 20<25 || 5 == 5;`