Detailed Summary Table for Java Operator Evaluation

Type of Operation	Operators	Order of Evaluation	Example	Explanation
Arithmetic	+, -, *, /, %	Left-to-right	int result = 10 + 5 * 2;	* is evaluated first (5 * 2), then +, resulting in 20.
String Concatenation	+ (when used with strings)	Left-to-right	"Hello " + "World!"	Concatenates "Hello " with "World!" to produce "Hello World!".
Increment / Decrement	++, -- (prefix)	Right-to-left	int a = ++b;	++b increments b first, then assigns the result to a.
	++, -- (postfix)	Left-to-right	int a = b++;	b++ assigns b to a first, then increments b.
Relational Operators	<, <=, >, >=	Left-to-right	if (a < b)	Evaluates if a is less than b.
Equality Operators	==, !=	Left-to-right	a == b	Checks if a is equal to b.
Logical AND / OR	&&, ||	Left-to-right	a && b	Evaluates a; if a is false, stops and returns false. If a is true, evaluates b.
Bitwise AND / OR / XOR	&, |, ^	Left-to-right	a & b	Performs bitwise AND on each bit of a and b.
Bitwise NOT	~	Right-to-left	~a	Flips all bits in a.
Shift Operators	<<, >>, >>>	Left-to-right	a << 2	Left-shifts a by 2 bits.
Assignment Operators	=, +=, -=, *=, /=, %=	Right-to-left	int x = y = 5;	First assigns 5 to y, then y to x.
Conditional (Ternary)	?:	Right-to-left	a > b ? x : y	Evaluates a > b; if true, returns x; if false, returns y.
Type Casting	(type)	Right-to-left	(int) 4.5	Casts 4.5 to an int, resulting in 4.
Method Calls	object.method()	Left-to-right	str.toUpperCase()	Calls toUpperCase() on str.

Array Access	array[index]	Left-to-right	arr[2]	Accesses the third element of arr.
Member Access (dot operator)	object.field, object.method()	Left-to-right	obj.field	Accesses field of obj.
Lambda Expression	->	Left-to-right	(x) -> x + 1	Creates a lambda function that adds 1 to x.
Method Reference	::	Left-to-right	String::toUpperCase	References String's toUpperCase method.
Grouping with Parentheses	()	Left-to-right	(a + b) * c	Groups a + b to ensure it's evaluated before multiplying by c.
Logical NOT	!	Right-to-left	!isTrue	Negates isTrue; if isTrue is true, result is false.
Compound Expressions	+=, -=, *=, etc.	Right-to-left	a += 5	Adds 5 to a and assigns the result to a.