

Conditional Operators

```
if (x > 0)
```

```
    y = 1
```

```
else
```

```
    y = -1;
```

is equivalent to

```
y = (x > 0) ? 1 : -1;
```

```
(boolean-expression) ? expression1 : expression2
```

Ternary operator

Binary operator

Unary operator

Conditional Operator (1 of 2)

```
if (num % 2 == 0)
    System.out.println(num + "is even");
else
    System.out.println(num + "is odd");

System.out.println(
    (num % 2 == 0)? num + "is even" :
    num + "is odd");
```

Conditional Operator (2 of 2)

`boolean-expression ? exp1 : exp2`

Operator Precedence

- `var++`, `var--`
- `+`, `-` (Unary plus and minus), `++var`, `--var`
- `(type)` Casting
- `!` (Not)
- `*`, `/`, `%` (Multiplication, division, and remainder)
- `+`, `-` (Binary addition and subtraction)
- `<`, `<=`, `>`, `>=` (Relational operators)
- `==`, `!=`; (Equality)
- `^` (Exclusive OR)
- `&&` (Conditional AND) Short-circuit AND
- `||` (Conditional OR) Short-circuit OR
- `=`, `+=`, `-=`, `*=`, `/=`, `%=` (Assignment operator)

Operator Precedence and Associativity

The expression in the parentheses is evaluated first. (Parentheses can be nested, in which case the expression in the inner parentheses is executed first.) When evaluating an expression without parentheses, the operators are applied according to the precedence rule and the associativity rule.

If operators with the same precedence are next to each other, their associativity determines the order of evaluation. All binary operators except assignment operators are left-associative.

Operator Associativity

When two operators with the same precedence are evaluated, the **associativity** of the operators determines the order of evaluation. All binary operators except assignment operators are **left-associative**.

$a - b + c - d$ is equivalent to $((a - b) + c) - d$

Assignment operators are **right-associative**. Therefore, the expression

$a = b + = c = 5$ is equivalent to $a = (b + = (c = 5))$

Example

Applying the operator precedence and associativity rule, the expression $3 + 4 * 4 > 5 * (4 + 3) - 1$ is evaluated as follows:

