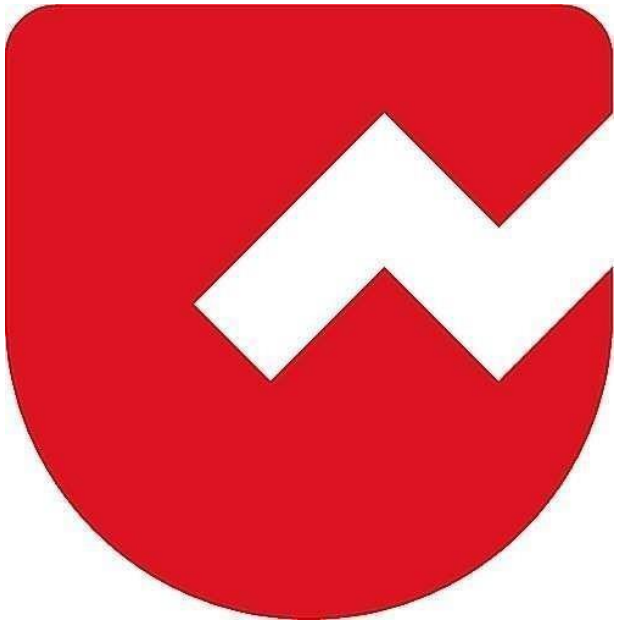
Zoho Schools for Graduate Studies



Notes

Literals

A literal is a constant value that is directly written in the code (fixed value assigned to a variable).

They are case-insensitive

1) Integer Literals

- Whole numbers (without decimal).
- Can be written in:
  - Decimal (base 10): int a = 25;
  - Octal (base 8, prefix 0): int b = 025;
  - Hexadecimal (base 16, prefix 0x/0X): int c = 0x1A;
  - Binary (base 2, prefix 0b/0B): int d = 0b1010;

2) Floating-Point Literals

- Numbers with decimal point or exponent (scientific notation).
- Example: 3.14, 2.5e3 (→ 2500.0)
- By default → double, use f or F for float:
  - float f = 3.14f;

3) Character Literals

- Single character inside single quotes: 'A', '9'
- Can use escape sequences:
  - '\n' (newline), '\t' (tab), '\\' (backslash), '\'' (single quote)

4) String Literals

- Sequence of characters inside double quotes.
- Example: "Hello", "Java\nWorld"

5) Boolean Literals

- Only two values: true or false

6) Null Literal

- Represents "no object".
- Example: String s = null;

Underscore Usage

- Underscore improves readability of numeric literals.
- Cannot be placed at start/end, before decimal/suffix/prefix.
- From Java 9, _ is reserved, not usable as variable name.

Valid Examples

```
int oneMillion = 1_000_000;
long creditCard = 1234_5678_9012_3456L;
double pi = 3.14_159_265;
int binary = 0b1010_1011;
```

Invalid Examples

```
int x = _100;      // cannot start with _
int y = 100_;      // cannot end with _
double d = 100_.0; // not next to decimal
float f = 3.14_f;  // not next to suffix
int hex = 0x_FF;   // not just after 0x
int _ = 10; // Allowed in Java 7/8
int _ = 20; // Compilation error
```

Operators in Java

Operand

An operand is the data (value, variable, or object) on which an operator acts.

Operator

An operator is a symbol that represents an action to be performed.

Operation

An operation is the complete action of applying an operator on operands.

Types of Operators (From high to low precedence)

1. Unary Operator
2. Arithmetic Operator
3. Shift Operator
4. Relational Operator
5. Bitwise Operator
6. Logical Operator
7. Ternary Operator
8. Assignment Operator

Questions

1)Negation

Code

```
int a=5;
System.out.println(~a);
```

Output

6

Explanation

General Rule

For any integer a:

~a = -(a + 1)

Therefore:

~5 = -(5 + 1) = -6

2)Modulus operator

Code

```
System.out.println(-12%5);
```

Output

-2

Explanation

Perform modulus operation and give the sign of the numerator for the result

3)Modulus operator

Code

```
System.out.println(12%-5);
```

Output

2

Explanation

Perform modulus operation and give the sign of the numerator for the result

4)Arithmetic operator

Code

```
short a=5,b=6;
Short c=a+b;
System.out.println(c);
```

Output

Runtime type error

Explanation

Addition is performed only for integer values. So, after performing the addition operation the result is of type int and must be cast back to short

Corrected code

```
short a=5,b=6;
Short c=(short)a+b;
System.out.println(c);
```

Output

11

5)Arithmetic operator

Code

```
int a = 4, b = 3, x = 5;
x /= a * b;
System.out.println(x);
```

Output

0

Explanation

- Evaluate a * b first (multiplication has higher precedence than assignment):
  a * b = 4 * 3 = 12.
- Compute x / (a*b) using integer division:
  x / 12 = 5 / 12. Since x, a, b are int, this is integer division — fractional part discarded. 5 / 12 → 0 (0.416... truncated to 0).