Data Types

Data types are defined as the type of data which a variable can store.

Ex- String s="Shivam";

Data types are divided into two categories:

1.Primitive Data Types
2.Non-Primitive Data Types

int a=10,20

Primitive Data Types:-

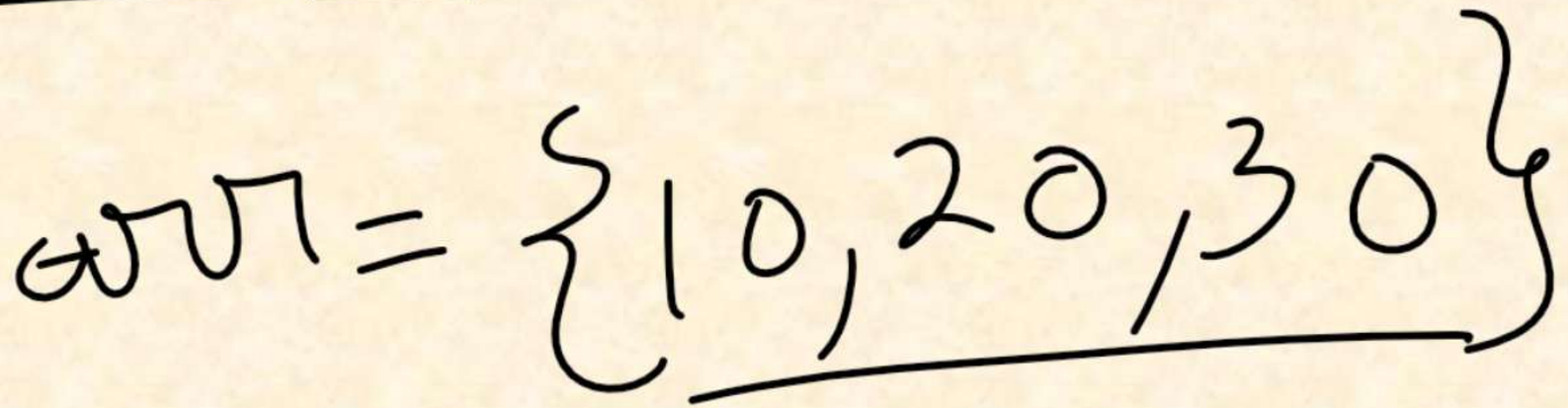
Primitive data types can store only one value.

Ex- int, float, long, double, short, char, byte, boolean.

Non-Primitive Data Types:-

Non-Primitive data types can store multiple value.

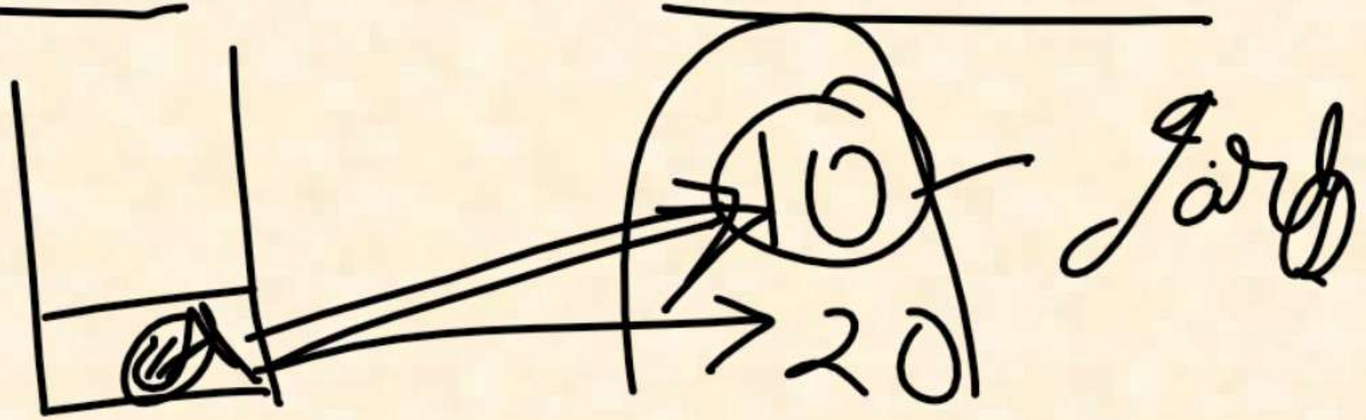
Ex- class, Array, String, etc.



Variable

A variable is the name of a memory location where we store different types of data.

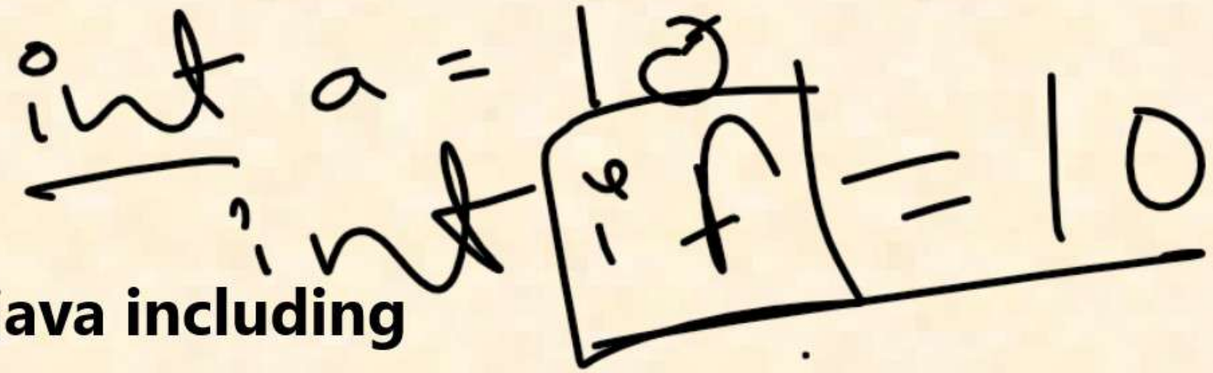
Ex- int a=10; here a is variable.



Keyword

Keywords are the reserved words whose meaning is already defined in compiler.
There are 50 keywords in java.

Ex- class, int, if, for, public, etc.

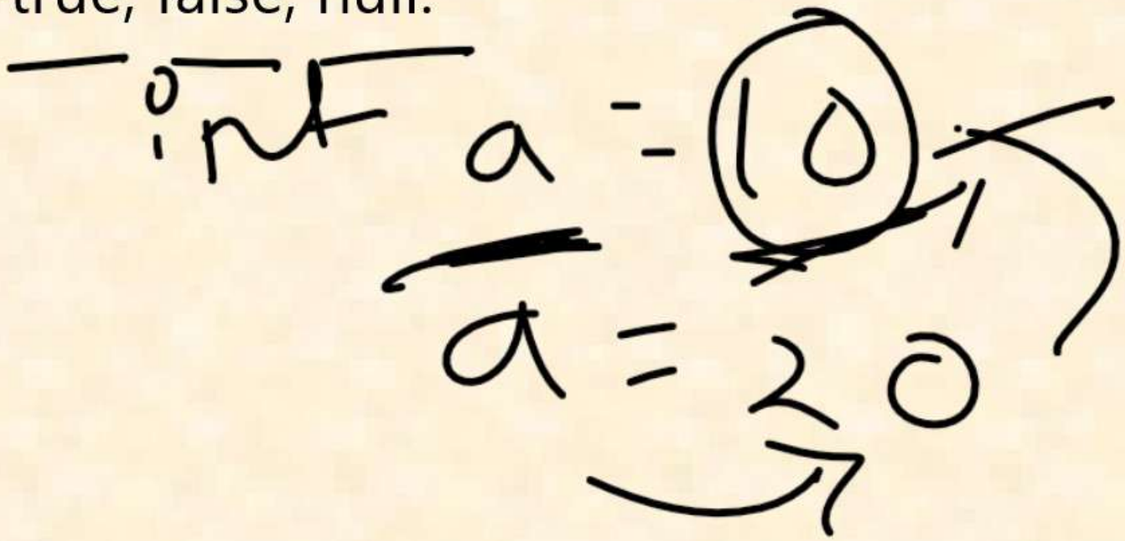


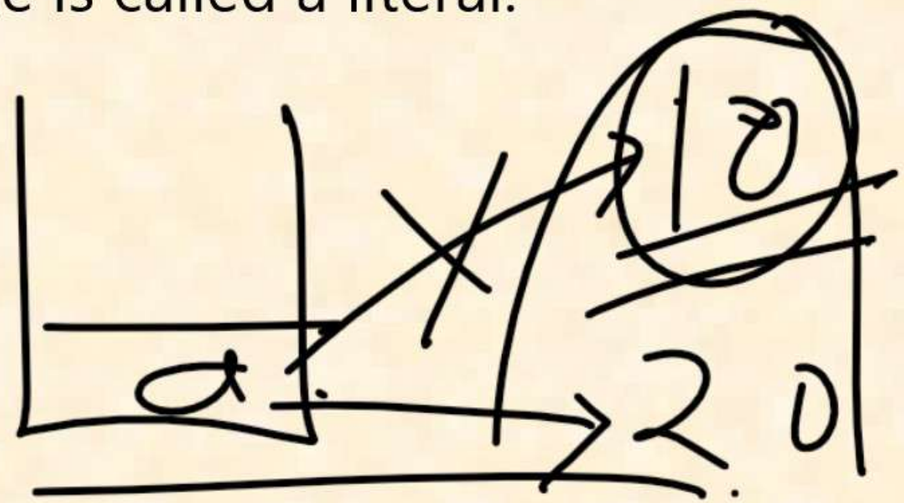
Note: There are 53 reserved words in java including 50 keywords and 3 literals.

Literals

A constant value which can be assigned to a variable is called a literal.

Ex-true, false, null.





OPERATORS IN JAVA

These are predefined symbols used to perform operations on the operands.

Types of operators are:-

- Arithmetic Operators (+, -, *, /, %)
- Relational (Comparison) Operators(==, !=, >, <, >=, <=)
- Logical Operators(&&, ||, !)
- Assignment Operators(=, +=, -=, *=, /=, %=)
- Unary Operators(++, --, +, -, !)
- Bitwise Operators(&, |, ^, ~)
- Shift Operators(<<, >>, >>>)
- Conditional (Ternary) Operator(?:)
- Type Comparison Operator(instanceof)

a+b=

35/20=1

20)35(1

20

15

a%b=15

a=35, b=20

20)35

20

15

a%b=15

Relational (Comparison) Operators(==, !=, >, <, >=, <=)

a=10 b=20 Used to compare two values.

c=10

== -> return True if two values are equal

!= -> return True if two values are not equal

> ->return True if condition is satisfy

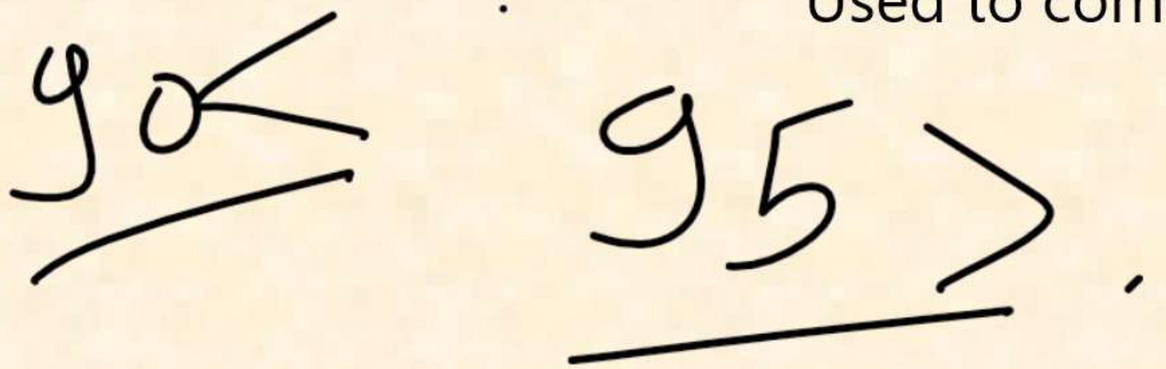
< ->return True if condition is satisfy

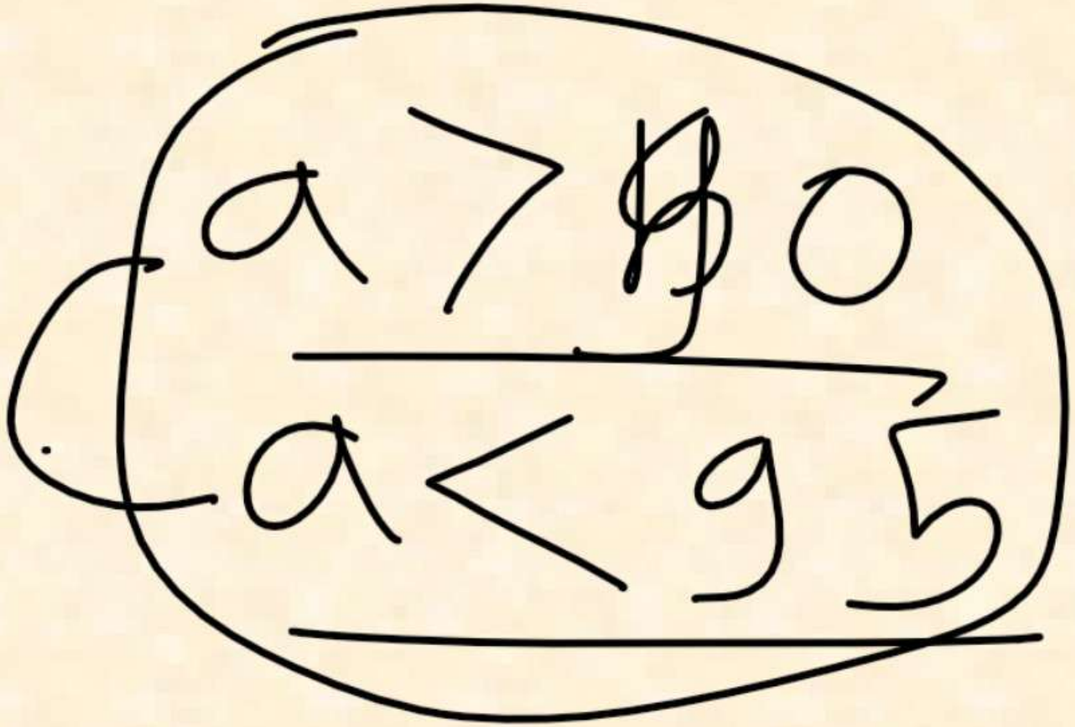
>=->return True if condition is satisfy

<=->return True if condition is satisfy

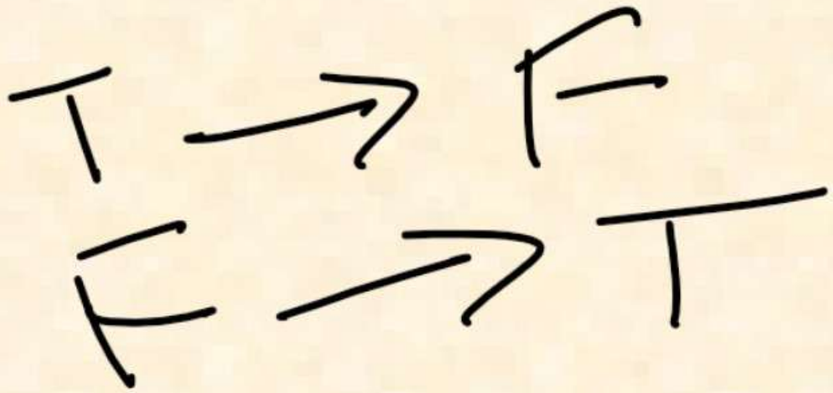
Logical Operators(&&, ||, !)

Used to combine conditions.



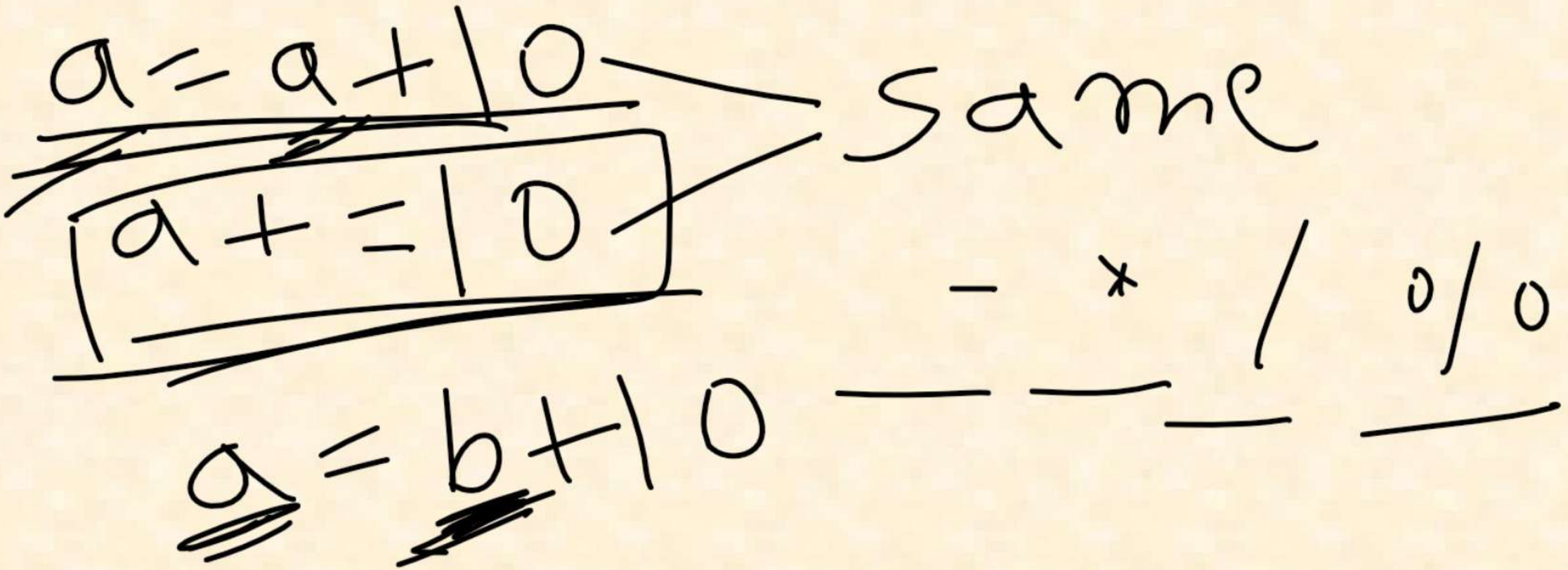


&& -> return True if both condition true
|| -> return True if any one of both condition is true
! -> reverse the condition



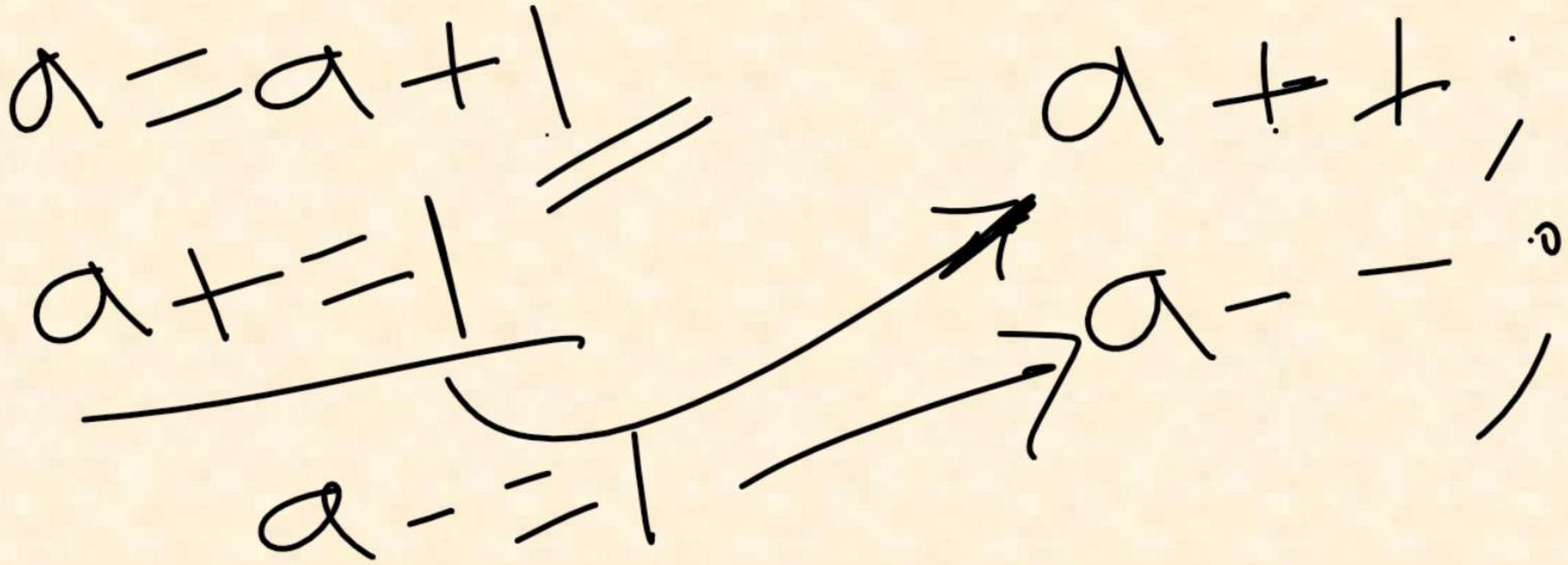
Assignment Operators(=, +=, -=, *=, /=, %=)

Used to assign values.



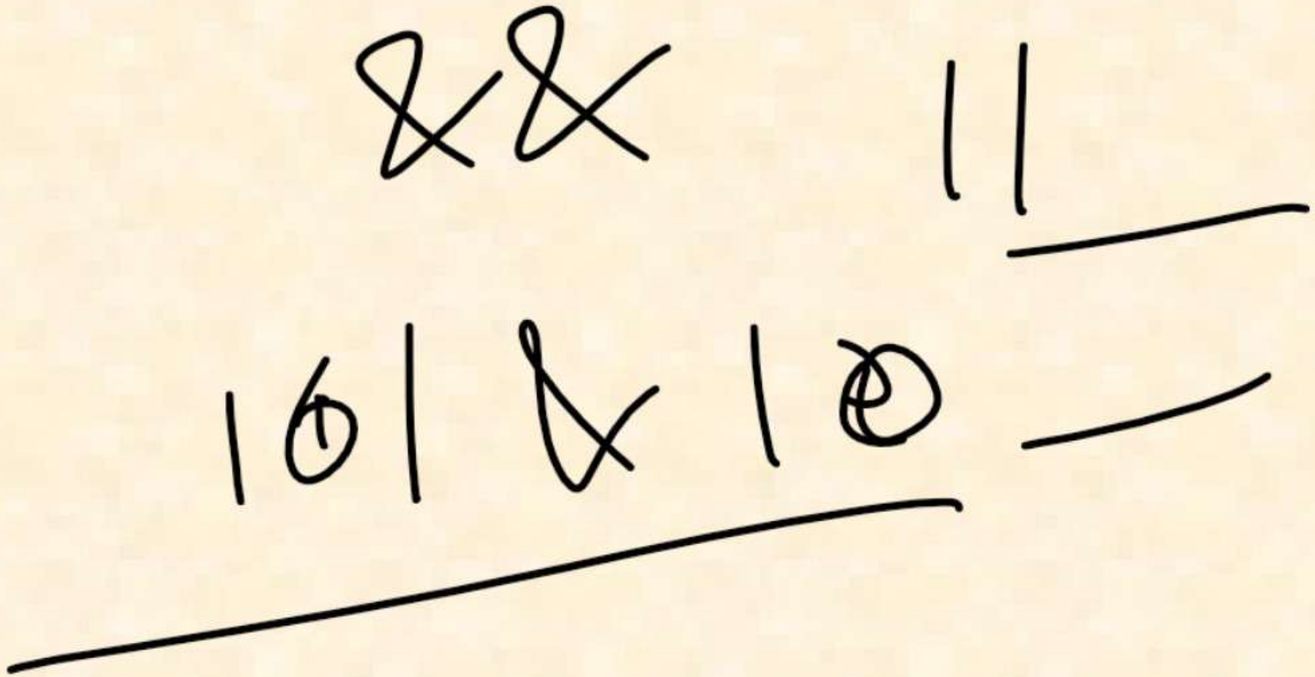
Unary Operators(++, --, [redacted])

Operate on a single operand.



Bitwise Operators(&, |, ^, ~)

Operate on bits.



Shift Operators(<<, >>, >>>)

Shift bits left or right.

$8 >> 1 = 1 \times 2^{3}$

Conditional (Ternary) Operator(?:)

Shorthand for if-else.

Syntax:- (Condition)? Yes : No;

Type Comparison Operator(instanceof)

Used to test object type.