MODULE 3

BASIC OPERATORS

Operators

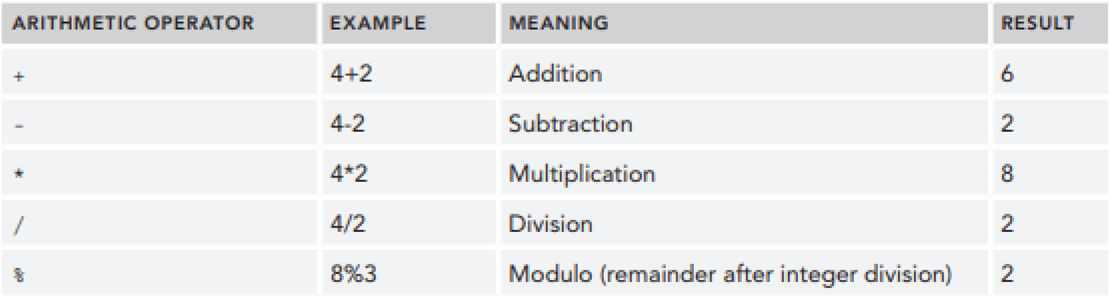
Perform data manipulations on one or more input variables (called operands). For example, in the expression 2+3, the operands are 2 and 3, and the operator is +. In terms of the number of operands, a distinction can be made among unary operators (one operand), binary operators (two operands), and ternary operators (three operands). In terms of the operations performed, a distinction can be made among the following:

- ❖ **Arithmetic operators**
- ❖ **Relational operators**
- ❖ **Logical operators**
- ❖ **Assignment operators**

Arithmetic Operators

Arithmetic operators perform basic mathematical operations on numerical values. The most popular ones are listed in Table 3.1.

Table 3.1: Arithmetic Operators



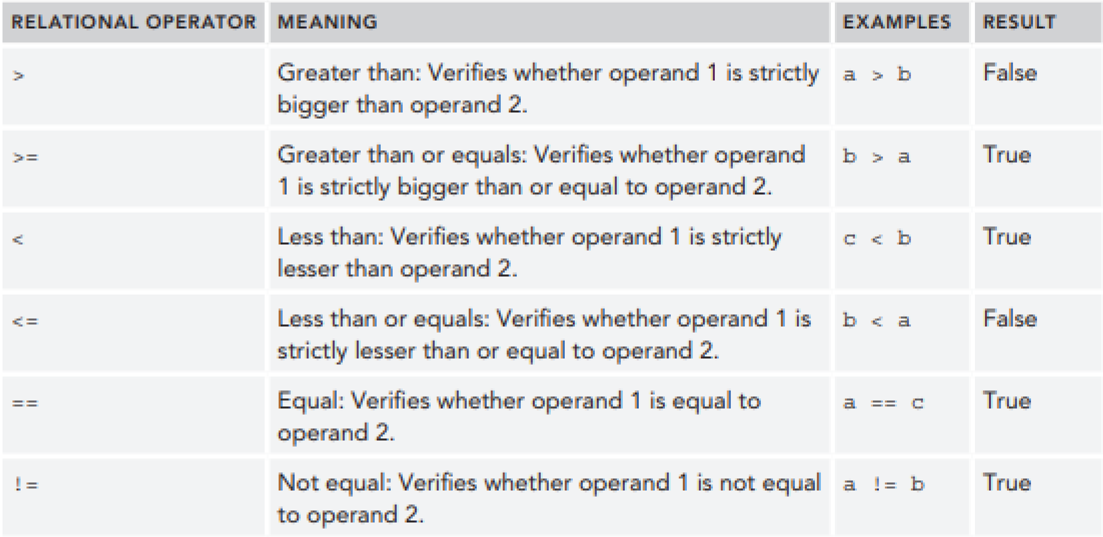
ARITHMETIC OPERATOR	EXAMPLE	MEANING	RESULT
+	4+2	Addition	6
-	4-2	Subtraction	2
*	4*2	Multiplication	8
/	4/2	Division	2
%	8%3	Modulo (remainder after integer division)	2

Relational Operators

Relational operators are usually binary operators. They check the relationship between two operands that are usually numbers or at least can be represented as numbers. They typically return a Boolean value. Consider the following variables:

- **int a=4;**
- **int b=9;**
- **int c=4;**

Table 3.2 illustrates the relational operators that can be used in Java.

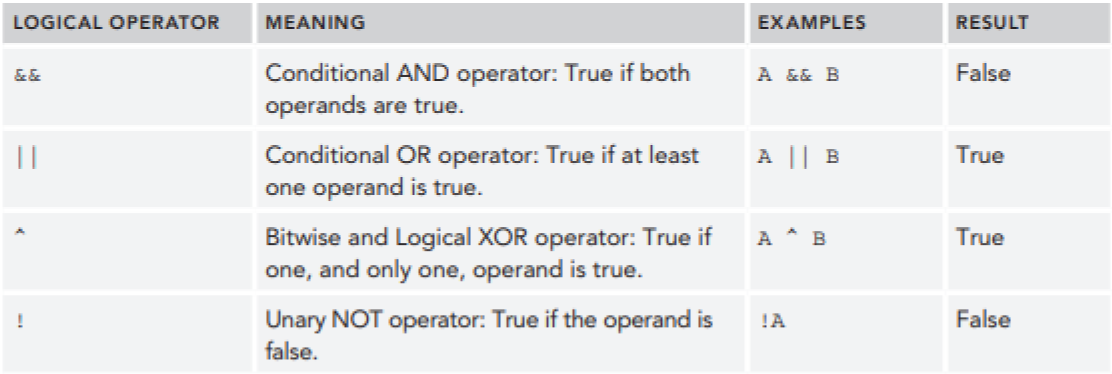


RELATIONAL OPERATOR	MEANING	EXAMPLES	RESULT
>	Greater than: Verifies whether operand 1 is strictly bigger than operand 2.	a > b	False
>=	Greater than or equals: Verifies whether operand 1 is strictly bigger than or equal to operand 2.	b > a	True
<	Less than: Verifies whether operand 1 is strictly lesser than operand 2.	c < b	True
<=	Less than or equals: Verifies whether operand 1 is strictly lesser than or equal to operand 2.	b < a	False
==	Equal: Verifies whether operand 1 is equal to operand 2.	a == c	True
!=	Not equal: Verifies whether operand 1 is not equal to operand 2.	a != b	True

Logical operators

A logical operator returns a Boolean result based on the Boolean result of one or more expressions. For this reason, they may also be called Boolean operators. Logical or Boolean operators are always evaluated from left to right. Consider, for example, the following expressions and their Boolean results. Table 3.3 then illustrates the evaluation of the logical operators that can be used in Java on these expressions.

- **A: 3 > 2 (True)**
- **B: 2 < 1 (False)**

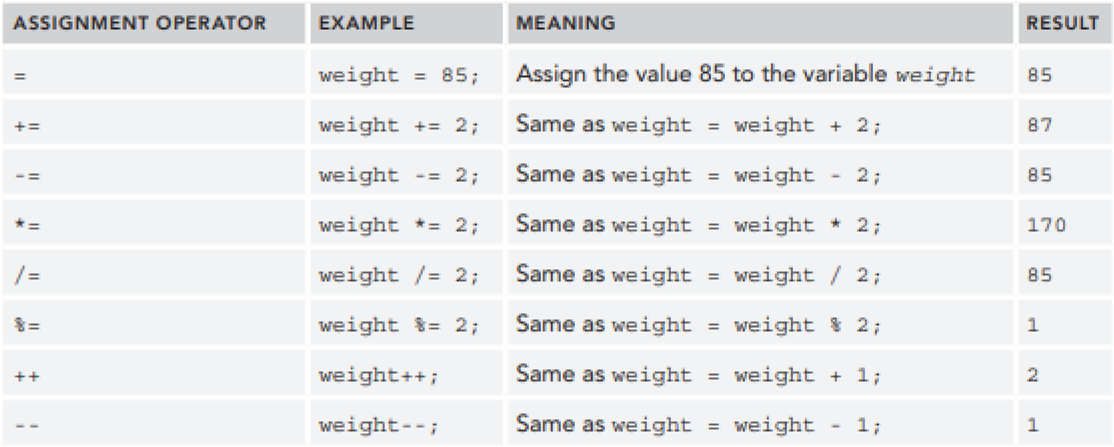


LOGICAL OPERATOR	MEANING	EXAMPLES	RESULT
`&&`	Conditional AND operator: True if both operands are true.	`A && B`	False
`||`	Conditional OR operator: True if at least one operand is true.	`A || B`	True
`^`	Bitwise and Logical XOR operator: True if one, and only one, operand is true.	`A ^ B`	True
`!`	Unary NOT operator: True if the operand is false.	`!A`	False

Assignment Operators

The assignment operator assigns values to a variable. In previous examples, you read about the (=) operator, which assigns a value to a variable. Table 3.4 lists some important assignment operators.

Table 3.2 Assignment Operator



ASSIGNMENT OPERATOR	EXAMPLE	MEANING	RESULT
=	weight = 85;	Assign the value 85 to the variable *weight*	85
+=	weight += 2;	Same as weight = weight + 2;	87
-=	weight -= 2;	Same as weight = weight - 2;	85
*=	weight *= 2;	Same as weight = weight * 2;	170
/=	weight /= 2;	Same as weight = weight / 2;	85
%=	weight %= 2;	Same as weight = weight % 2;	1
++	weight++;	Same as weight = weight + 1;	2
--	weight--;	Same as weight = weight - 1;	1