

```
int a = 10;  
int b = 15;
```

```
System.out.println("Sum is: " + (a + b));  
System.out.println("Sum is: " + a + b);
```

Sum is: 25
Sum is: 1015

2+2:2 = 3
(2+2):2 = 2
yuxaridan -> ashagi
sola -> saga

```
int a = 100;  
int b = 15;  
  
System.out.println("Before:");  
System.out.println(a);  
System.out.println(b);  
  
a++; // => (a = a + 1) -> a = 100 + 1 = 101  
b--; // => (b = b - 1) -> b = 15 - 1 = 14  
  
System.out.println("After:");  
System.out.println(a);  
System.out.println(b);
```

Before:
100
15
After:
101
14

```
int a = 100;  
int b = 15;  
  
System.out.println(a++); // 100 -> print:100 -> a = a + 1 -> a = 101  
System.out.println(b--); // 15 -> print:15 -> b = b - 1 -> b = 14  
  
a = 101; b = 14;  
  
System.out.println(a); // 101  
System.out.println(b); // 14
```

```

int a = 100;
int b = 15;

System.out.println(a);    // 100
System.out.println(b);    // 15

System.out.println(++a);  // 100 => a = a + 1 -> 101
System.out.println(--b);   // 14  => b = b - 1 -> 14

System.out.println(a++);  // 101 => a = a + 1 = 102
System.out.println(b--);  // 14  => b = b - 1 = 13

System.out.println(a);    // 102
System.out.println(b);    // 13

```

100
 15
 101
 14
 101
 14
 102
 13

| Operator | Description | Example |
|-------------------------------|---|-----------------------|
| == (equal to) | Checks if the values of two operands are equal or not, if yes then condition becomes true. | (A == B) is not true. |
| != (not equal to) | Checks if the values of two operands are equal or not, if values are not equal then condition becomes true. | (A != B) is true. |
| > (greater than) | Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true. | (A > B) is not true. |
| < (less than) | Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true. | (A < B) is true. |
| >= (greater than or equal to) | Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true. | (A >= B) is not true. |
| <= (less than or equal to) | Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true. | (A <= B) is true. |

| Operator | Description | Example |
|--------------------|--|---------------------|
| + (Addition) | Adds values on either side of the operator. | A + B will give 30 |
| - (Subtraction) | Subtracts right-hand operand from left-hand operand. | A - B will give -10 |
| * (Multiplication) | Multiplies values on either side of the operator. | A * B will give 200 |
| / (Division) | Divides left-hand operand by right-hand operand. | B / A will give 2 |
| % (Modulus) | Divides left-hand operand by right-hand operand and returns remainder. | B % A will give 0 |
| ++ (Increment) | Increases the value of operand by 1. | B++ gives 21 |
| -- (Decrement) | Decreases the value of operand by 1. | B-- gives 19 |

Show Examples ▾

| Operator | Description | Example |
|------------------|--|-------------------|
| && (logical and) | Called Logical AND operator. If both the operands are non-zero, then the condition becomes true. | (A && B) is false |
| (logical or) | Called Logical OR Operator. If any of the two operands are non-zero, then the condition becomes true. | (A B) is true |
| ! (logical not) | Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false. | !(A && B) is true |

| Operator | Description | Example |
|----------|---|---|
| = | Simple assignment operator. Assigns values from right side operands to left side operand. | C = A + B will assign value of A + B into C |
| += | Add AND assignment operator. It adds right operand to the left operand and assign the result to left operand. | C += A is equivalent to C = C + A |
| -= | Subtract AND assignment operator. It subtracts right operand from the left operand and assign the result to left operand. | C -= A is equivalent to C = C – A |

| | | |
|-----------------|--|---|
| <code>*=</code> | Multiply AND assignment operator. It multiplies right operand with the left operand and assign the result to left operand. | <code>C *= A</code> is equivalent to <code>C = C * A</code> |
| <code>/=</code> | Divide AND assignment operator. It divides left operand with the right operand and assign the result to left operand. | <code>C /= A</code> is equivalent to <code>C = C / A</code> |
| <code>%=</code> | Modulus AND assignment operator. It takes modulus using two operands and assign the result to left operand. | <code>C %= A</code> is equivalent to <code>C = C % A</code> |

Conditional Operator (`? :`)

Conditional operator is also known as the **ternary operator**. This operator consists of three operands and is used to evaluate Boolean expressions. The goal of the operator is to decide, which value should be assigned to the variable. The operator is written as –

```
variable x = (expression) ? value if true : value if false
```

```
double grade = 80;
double scholarship = grade > 90 ? 200 : 100;
System.out.println(scholarship);
```