Operators:

Operators in Java refers to the **Special Symbols.**
These **Specials Symbols** are generally used for performing **Operations** in the program.
The **Operation** may be **Arithmetic or Logic** based on the **Condition**

1.Arthematic Operators
Addition **+**
Subtraction –
Multiplication ***
Division **/**
Modulus **%**

2.Unary Operators
Increment **++**
Decrement **–**

3.Assignment Operators assignment
=
+=
-=
***=**
/=
%=

4.Relational Operators
(Comparison Operators)

equal to **= =**
not equal to **!=**
greater than **>**
less than **<**
greater than equal to >=
less than equal to **<=**

5.Short Circuit Logical Operators

logical AND **&&**
logical OR ||
logical NOT !

6.Boolean Logical Operators

& AND
| OR
^ Exclusive OR

7.Ternary Operators

ternary **? :**

8. Left Shift Operator <<
9. Right Shift Operator >>

Unary Operators

+ Unary plus operator; *indicates* ***positive*** *value*

- Unary minus operator; ***negates*** *an expression*

++ Increment operator; *increments a value by* ***1***

-- Decrement operator; *decrements a value by* ***1***

Logical && AND Operator

All expression needs to be true

Logical || OR Operators

At least one expression needs to be true

! Logical complement operator; *inverts the value of a* ***boolean***

Ternary Operator ? :

Any one of the expression must be True or else False

Separators:

Separators are the **symbols** which are used to **divide** or **arrange** the **code**.
These are **predefined symbols** which generally used to give the **shape of our function or the program.**

Parenthesis() – Generally used to **add parameters** in the functions.
Braces{} – Generally used to define the **classes or functions** and also used to initialize the arrays.
Brackets[] – Generally used for **indexing** of an array.
Comma “,” – It is used to **separate** different elements in the program such as identifiers,
Semicolon “;” – It is used to **end** any statement in the program.
Period . Used to **separate** the **package names** from sub packages and classes.

```java
//Arithmetic Operators + - * % /
public class Eg1 {

public static void main(String[] args) {

int a = 15;
int b = 3;
System.out.println(a+b); // 18
System.out.println(a-b); // 12
System.out.println(a*b);  //45
System.out.println(a%b); //0
System.out.println(a/b); // 5
}

}
```

```
//Assignment Operators += -= *= /= %=
int a = 10;
int b = 20;
System.out.println(a += b); // a = a+b // a= 10+20 // 30

a = 30;
b = 40;
System.out.println(a -= b); // b = a-b // b = 30-40 // -10

a = 5;
b = 2;
System.out.println(a *= b); // a=a*b // a= 5*2 // 10

a = 10;
b = 2;
System.out.println(a /= b); // a=a/b // a=10/2 // 5

a = 10;
b = 20;
System.out.println(a %= b); // 10
}
}
```

```java
//Unary Operators ++ --
public class Eg3 {
public static void main(String[] args) {

int a = 10;
System.out.println(a++); // 10
System.out.println(a++); // 11 //post Increment by 1 (increase next)
System.out.println(a++); // 12 // post Increment by 1

int b = 20;
System.out.println(++b); // 21 // pre increment by 1 (increase first)
System.out.println(++b); // 22 // pre increment by 1

int x = 10;
x++;
System.out.println(x); // 11

int y = 20;
y++;
System.out.println(y); // 21
}
}
```

```java
//Comparison Operators == < > <= >= !=
public class Eg4 {

public static void main(String[] args) {

int a = 10;
int b = 20;
int c = 10;

System.out.println(a == b); // false
System.out.println(a == c); // true
System.out.println(a < b); // true
System.out.println(a > b); // false
System.out.println(a >= c); // true
System.out.println(a <= c); // true
System.out.println(a != c); // false
}
}
```

```java
// Boolean Logical Operands & || !
public class Eg5 {

public static void main(String[] args) {

System.out.println(true & true); // true
System.out.println(false & true); // false
System.out.println(true & false); // false
System.out.println(false & false); // false

System.out.println(true | true); // true
System.out.println(false | true); // true
System.out.println(true | false); // true
System.out.println(false | false); // false

System.out.println(true != true); // Comparing identical expressions // false
}
}
```

```
//ternary operator
public class Eg6 {

public static void main(String[] args) {

int a, b;
a = 10;

//variable = Expression1 ? Expression2 : Expression3
b = (a==1) ? 20 : 30;  //If the left hand expression is true Expression will Execute or Else Right Hand Expression Execute
System.out.println(b); //30

b = (a==10) ? 20 : 30; //If the left hand expression is true Expression will Execute or Else Right Hand Expression Execute
System.out.println(b); //20

}

}
```

```
public class Eg7 {

public static void main(String[] args) {

int a = 10;
System.out.println(a); // 10
System.out.println(a++); // 10 M 11
System.out.println(a);// 11
System.out.println(a++);// 11 M12
System.out.println(a); // 12

}

}
```

```
public class Eg8 {

public static void main(String[] args) {

int a = 10;
System.out.println(a++); // 10 M 11
System.out.println(--a); // 10
System.out.println(--a);// 9
System.out.println(--a);// 8
}

}
```

```
public class Eg9 {

public static void main(String[] args) {

int a = 10;
int b = 10;

int c = a++ + ++b; // 10 + 11
System.out.println(c); // 21

}
}
```

```
public class Eg10 {

public static void main(String[] args) {

int a = 10;
int c = ++a - --a; // 11 - 10
System.out.println(c); // 1

int x = 10;
int y = x++ - x--; //10 - 11
System.out.println(y); //-1
}

}
```

```java
//Assignment Operators
public class Eg11 {
public static void main(String[] args) {

int b = 5;
b -= 2; // b = b-2 //b = 5-2 // 3
System.out.println(b); // 3

int c = 10;
c *= 2; // c = c*2 // c = 10*2 // 20
System.out.println(c); // 20

int d = 15;
d /= 3; // d = d/3 // d = 15/3 // d = 5
System.out.println(d); // 5

int e = 15;
e%=3; //e = e%3 // e= 15%3 //0
System.out.println(e); // 0
}
}
```

```
public class Eg12 {

public static void main(String[] args) {

// Both are same false
System.out.println(true ^ true); // false
// Both are same false
System.out.println(false ^ false); // false
// Both are different true
System.out.println(true ^ false); // true
// Both are different true
System.out.println(false ^ true); // true
}
}
```

A	B	A&B	A|B	A^B
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

```
public class Eg13 {
public static void main(String[] args) {
int a = 10; // 0b1010
int b = 2;   // 0b0010

System.out.println(a & b); // 2 // 0b0010

System.out.println(a | b); //10  //0b1010

System.out.println(a^b); // 8 // 0b1000
}
}
```

A	B	A&B	A|B	A^B
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

Use 8 bit for **Left Shift Operator**, actually it is 4 bit

```
int a = 10; // 0b1010
int b = 2;   // 0b0010
```

```
int a = 10;
int b = 2;
System.out.println(a<<b); // 40
```

add 4 zeros before: 0b00001010

Now left shift is 2 **remove two zeros from left and add at right (append)**

Note: append only zeros and remove can be 0 or 1

Add four zeros Before of it	Append
0b00001010	0b00101000

Right Shift

Remove two numbers from right side and append at left side that can be only zeros not 1’s

```
public class Eg14 {

public static void main(String[] args) {

int a = 10; // 0b1010
int b = 2; // 0b0010

System.out.println(a >> b); // 2

// Change to 8 bit
// 0b00001010

// Remove from Right and Append to left
// 0b00000010

}
}
```