

② ~~AND Gate~~:

Logical operators are used for logical decision making purpose.

Logical operators are used ① AND operator (&&).

② OR operator (||).

③ NOT operator (!).

Note: →

The operand type for logical operator must be boolean type.

The output of logical operator will be boolean type.

AND Operator (&&):

- It is a binary operator.
- AND Operator returns true only if both the operands are true.
- AND operator returns false when any one of the operand returns false.

Truth Table:

operand 1	operand 2	output
True	True	True
True	False	False
False	True	False
False	False	False

OR Operator :

- It is a binary operator.
- OR operator returns true if any one of the operand is true.

Truth Table :

operand1	operand2	output
True	False	True
False	True	True
True	True	True
False	False	False

NOT Operator

- It is a unary operator.
- NOT operator returns true if input is False and vice versa.

Truth Table :

operand	output
True	False
False	True

Mixture → physically mixed.
Compound → chemically mixed.

Eg: →

```
class Logical{
```

```
{  
    public static void main (String args[])
```

```
{  
        System.out.println (true && true); // true  
        System.out.println (true || true); // true  
        System.out.println (true && false); // false  
        System.out.println (!true && true); // false  
        System.out.println (0%2==0 || 1.1*2==1); // false true  
        System.out.println (4>6 && false); // false  
        System.out.println (!false || !true); // true  
        System.out.println ('a'=='A' || false==false); // true  
        System.out.println ('a'/'b'==0 || '2'==1 ?); // true  
    }
```

Compound Assignment Operators: (perform implicit narrowing)

- ① +=
- ② -=
- ③ /=
- ④ *=
- ⑤ % =

• It is a binary operators which is used to update a value in the variable.

• They are also known as update operator.
• They are ① += ② -= ③ /= ④ *= ⑤ % =

Eg:

```
class CompoundAssign{
```

```
{  
    public static void main (String args[])
```

```
{  
        int Sal = 5000;
```

```
        Sal += 2000;
```

```
        System.out.println (Sal);
```

```
        int balance = 15000;
```

```
        balance *= 2;
```

```
        System.out.println (balance);  
    }
```

Pre-Increment \rightarrow prefix
post-Increment \rightarrow suffix.

M	T	W	T	F	S	S
Page No.:						YOUVA
Date:						

Compound Assignment operator performs implicit narrowing

Eg: \rightarrow

```
int num = 300;
```

```
num = num + 101.01; // error: lossy conversion
```

```
num += 101.01; // 301 (implicit narrowing).
```

Syntax:

variable = variable + value \rightarrow
variable += value.

Note: \rightarrow compound Assignment performs implicit narrowing.

Increment and Decrement Operator: \rightarrow

- Increment operator is a ~~unary~~ unary operator.
- $++$ is used to increase the value of a variable by 1.
- $--$ is represented by $++$.
- There are two types of increment operator:

① Pre-Increment: \rightarrow If variable is prefixed with increment operator then it is called as pre-increment operator.

② Post-Increment \rightarrow If variable is ~~pre~~ suffixed with increment operator then it is called as post-increment operator.

Decrement Operator:

- The decrement operator is a unary operator used in programming to ~~increase~~ ^{decrease} the value of a variable by 1.
- It is represented by `--`.
- There are two type of decrement operators: →

① Pre-Decrement: →

If a variable is prefixed with decrement operator then it is called as pre-decrement operator.

② Post-Decrement →

If a variable is suffixed with decrement operator then it is called as post-decrement operator.