

OperatorsArithmetic operators

Addition + → Add

Subtraction - → Sub

Multiplication \* → Multi

division / → quotient

Modulus % → remainder

Exponentiation \*\*

Floor division //

not in java  
(python)Relational operators (Boolean output)

Equal to ==

Not Equal to !=

Greater than &gt;

less than &lt;

Greater than or equal to &gt;=

less than or equal to &lt;=

## Logical operator

AND                      &

true & true

OR                        ||

true || false

NOT                      !

true = false

false = true

## Bitwise operator

⇒ AND                      &

Eg:    5   &   2  
      101 & 010

      000 ⇒ (0)

⇒ OR                        |

Eg:    5   |   2  
      101 | 010

      111 ⇒ (7)

⇒ XOR                      ^

Eg:    5   ^   2  
      101 ^ 010 ⇒ 111

      (7)

AND		*
0	0	0
0	1	0
1	0	0
1	1	0

OR		+
0	0	0
0	1	1
1	0	1
1	1	1

8	4	2	1
---	---	---	---

XOR		
0	0	0
0	1	1
1	0	1
1	1	0



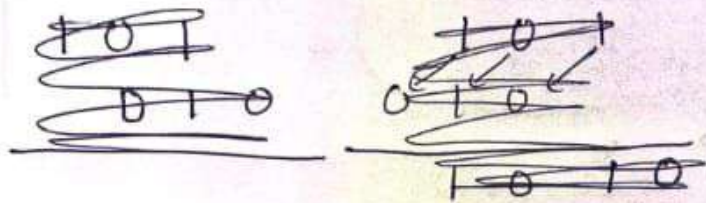
$\Rightarrow$  Not  $\sim$

$\Rightarrow$  Left shift  $\ll$

$\Rightarrow$  Right shift  $\gg$

Eg: left shift.

5  $\ll$  2  
101 ~~00~~

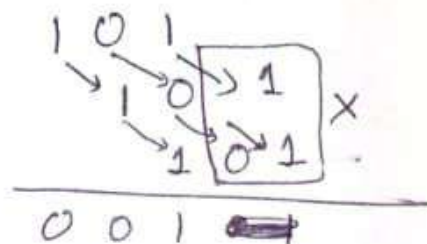


101  
10100  $\Rightarrow$  (20)

Eg: Right shift

5  $\gg$  2

$\Rightarrow$  (1)



## Assignment operator

Assign =

Add & assign +=

Sub & assign -=

Mul & assign \*=

Div & assign /=

Mod & assign %=



```
1 public class Operators {
2
3     public static void arithmeticOperators(){
4         int countofMangos = 10;
5         int countofApples = 3;
6
7         System.out.println("Arithmetic operators");
8
9         //Addition
10        System.out.println("Addition : " + countofMangos + "+" + countofApples + "=>" + (countofMangos + countofApples));
11
12        //Subtraction
13        System.out.println("Subttaction : " + countofMangos + "-" + countofApples + "=>" + (countofMangos - countofApples));
14
15        //Multiplication
16        System.out.println("Multiplication:" + countofMangos + "*" + countofApples + "=>" + (countofMangos * countofApples));
17
18        //Division
19        System.out.println("Division:" + countofMangos + "/" + countofApples + "=>" + (countofMangos / countofApples));
20
21        //Modulus
22        System.out.println("Modulus:" + countofMangos + "%" + countofApples + "=>" + (countofMangos % countofApples));
23
24        //Exponentiation
25
26        //Floor division
27
28        System.out.println();
29    }
30 }
31
```

```
1 public class Operators {  
2  
32     public static void relationalOperators(){  
33         int countofMangos = 10;  
34         int countofApples = 3;  
35  
36         System.out.println("Relational Operators");  
37  
38         //Equal to  
39         System.out.println("Equal to : " + countofMangos + "==" + countofApples + "=>"  
40         +(countofMangos == countofApples));  
41  
42         //Not Equal to  
43         System.out.println("Not Equal to : " + countofMangos + "!=" + countofApples + "=>"  
44         +(countofMangos != countofApples));  
45  
46         //Greater than  
47         System.out.println("Greater than : " + countofMangos + ">" + countofApples + "=>"  
48         +(countofMangos > countofApples));  
49  
50         //Less than  
51         System.out.println("Less Than : " + countofMangos + "<" + countofApples + "=>"  
52         +(countofMangos < countofApples));  
53  
54         //Greater Than or equal to  
55         System.out.println("Greater than equal to : " + countofMangos + ">=" + countofApples + "=>"  
56         +(countofMangos >= countofApples));  
57  
58         //Less than or equal to  
59         System.out.println("Less than equal to : " + countofMangos + "<=" + countofApples + "=>"  
60         +(countofMangos <= countofApples));  
61  
62         System.out.println();  
63     }  
64 }  
65
```

 Operators.java X

Day 13 >  Operators.java >  Operators >  allOperators()

```
1 public class Operators {
65
66     public static void logicalOperator(){
67         boolean isReady = true;
68         boolean isGood = false;
69
70         System.out.println(x:"Logical operators :");
71
72         //AND
73         System.out.println(" Logical AND : " + isReady + " AND " + isGood + " => " +(isReady && isGood));
74
75         //OR
76         System.out.println("Logical OR : " + isReady + " OR " + isGood + " -> " +(isReady || isGood));
77
78         //NOT
79         System.out.println("Logical NOT : " + "!" + isReady + " -> " +(! isReady));
80
81         System.out.println("Logical NOT : " + "!" + isGood + " -> " +(! isGood));
82
83         System.out.println();
84     }
85
86 }
```

```
1 public class Operators {
86
87     public static void bitwiseOperator(){
88         int redTeamScore = 5;
89         int whiteTeamScore = 2;
90
91         System.out.println("Bitwise Operator :");
92
93         //Bitwise AND " & "
94         System.out.println("Bitwise & : " + redTeamScore + " & " + whiteTeamScore + " => " + (redTeamScore & whiteTeamScore));
95
96         //Bitwise OR " | "
97         System.out.println("Bitwise | : " + redTeamScore + " | " + whiteTeamScore + " => " + (redTeamScore | whiteTeamScore));
98
99         //Bitwise XOR " ^ "
100        System.out.println("Bitwise ^ : " + redTeamScore + " ^ " + whiteTeamScore + " => " + (redTeamScore ^ whiteTeamScore));
101
102        //NOT " ~ "
103        System.out.println("Bitwise NOT ~ : " + " ~ " + redTeamScore + (redTeamScore));
104
105        System.out.println("Bitwise NOT ~ : " + " ~ " + whiteTeamScore + (whiteTeamScore));
106
107        //Left shift " << "
108        System.out.println("Bitwise << : " + redTeamScore + " << " + whiteTeamScore + " => " + (redTeamScore << whiteTeamScore));
109
110        //Right shift
111        System.out.println("Bitwise >> : " + redTeamScore + " >> " + whiteTeamScore + " => " + (redTeamScore >> whiteTeamScore));
112
113        System.out.println();
114    }
115 }
```



```
1 public class Operators {
115
116     public static void assignmentOperator(){
117         int totalScore = 5;
118
119         System.out.println(x:"Assignment Operators");
120
121         //Assign " = "
122         int assign = totalScore;
123         System.out.println("Assign :" +assign);
124
125         //Add and assign
126         totalScore += 2;
127         System.out.println("Add and assign :" +totalScore);
128
129         //Subtract and assign
130         totalScore -= 2;
131         System.out.println("Subtract and assign :" +totalScore);
132
133         //Multiple and assign
134         totalScore *= 2;
135         System.out.println("Multiple and assign :" +totalScore);
136
137         //Divide and assign
138         totalScore /= 2;
139         System.out.println("Divide and assign :" +totalScore);
140
141         //Modulus and assign
142         totalScore %= 2;
143         System.out.println("Modulus and assign :" +totalScore);
144
145         System.out.println();
146     }
147 }
```

 Operators.java ×

Day 13 >  Operators.java >  Operators >  allOperators()

```
1 public class Operators {  
147  
148     public static void allOperators()  
149     {  
150         arithmeticOperators();  
151         relationalOperators();  
152         logicalOperator();  
153         bitwiseOperator();  
154         assignmentOperator();  
155     }  
156  
157     public static void main(String[] args){  
158         allOperators();  
159     }  
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Filter Code

[Running] cd "c:\Users\ANANYA\OneDrive\Desktop\Algorithm365\Day 13\" && javac Operators.java && java Operators

Arithmetic operators

Addition :10+3->13

Subttaction : 10-3->7

Multiplication:10\*3->30

Division:10/3->3

Modulus:10%3=>1

Relational Operators

Equal to :10==3->false

Not Equal to :10!=3->>true

Greater than :10>3->>true

Less Than :10<3->false

Greater than equal to :10>=3->>true

Less than equal to :10<=3 => false

Logical operators :

Logical AND :true AND false -> false

Logical OR :true OR false => true

Logical NOT :!true -> false

Logical NOT :!false -> true

Bitwise Operator :

Bitwise & :5 & 2 => 0

Bitwise | :5 | 2 => 7

Bitwise ^ :5 ^ 2 => 7

Bitwise ~ : ~ 5-6

Bitwise NOT ~ : ~ 2-3

Bitwise << :5 << 2 -> 20

Bitwise >> :5 >> 2 => 1

Assignment Operators

Assign :5

Add and assign :7

Subtract and assign :5

Multiple and assign :10

Divide and assign :5

Modulus and assign :1

[Done] exited with code=0 in 2.173 seconds