## **Bitwise Operators**

In Java the bitwise and bit shift operators are used to manipulate the contents of variables at a bit level according to binary format. Java defines several bitwise operators, which can be applied to the integer types, long, int, short, char, and byte.

Bitwise operator works on bits and performs bit-by-bit operation. Assume if a = 60; and b = 13; now in binary format they will be as follows:

a = 0011 1100

b = 0000 1101

Then

a&b = 0000 1100

a|b = 0011 1101

a^b = 0011 0001

~a = 1100 0011

Operator	Description	Example
&	Binary AND Operator copies a bit to the result if it	(A & B) will give 12 which is
	exists in both operands.	0000 1100
I	Binary OR Operator copies a bit if it exists in	(A   B) will give 61 which is
	either	0011 1101
	operand.	
^	Binary XOR Operator copies the bit if it is set in	(A ^ B) will give 49 which is
	one operand but not both.	0011 0001
~	Binary Ones Complement Operator is unary and	(~A) will give -60 which is 1100
	has the effect of 'flipping' bits.	0011

<<	Binary Left Shift Operator. The left operands value	A << 2 will give 240 which is
	is moved left by the number of bits specified by	1111 0000
	the right operand.	
>>	Binary Right Shift Operator. The left operands	A >> 2 will give 15 which is 1111
	value is moved right by the number of bits	
	specified by the right operand.	
>>>	Shift right zero fill operator. The left operands	A >>>2 will give 15 which is
	value is moved right by the number of bits	0000 1111
	specified by the right operand and shifted values	
	are filled up with zeros.	

## For Example

The following simple example program demonstrates the bitwise operators.

public class BitwiseOperators {

public BitwiseOperators( ) {

int a = 11; //1 0 1 1

int b = 12; //1 1 0 0

System.out.println("a & b : "+(a & b));

System.out.println("a | b : "+(a | b));

System.out.println("a ^ b : "+(a ^ b));

System.out.println("~a : "+(~a));

System.out.println("a << b : "+(a << b));</pre>

System.out.println("a >> b : "+(a >> b));

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

}

public static void main(String args[]){

```
new BitwiseOperators();
```

}

}

## **Out Put Screen**

a & b : 8

a | b : 15

a ^ b : 7

~a : -12

a << b : 45056

a >> b : 0

a >>> b : 0