BITWISE OPERATOR(&, |, ^)

& --> AND ==> Returns true if and only both arguments are true

| --> OR ==> Returns true if atleast one argument is true

^ --> XOR ==> Returns true if both arguments are different

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

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

System.***out***.println(**true** ^ **false**); *// true*

We can apply this operator to integral types also.

System.***out***.println(4 & 5); // 4

Explaination:

100

100

-----------------

100

System.***out***.println(4 | 5); //5

Explaination:

100

101

-----------------

101

System.***out***.println(4 ^ 5); //1

Explaination:

100

101

----------------

001

Bitwise complement Operator(~)

We can apply this operator only for integral types but not for boolean type. If we are trying for boolean type then we will compile time error.

Example:

System.***out***.println(~**true**); //CE: bad operand type boolean for unary operator '~'

System.***out***.println(~4); //-5

Explaination:

4 = 0 0000...0100

Negation 4 = 1 1111...1011

if it is negative number we need to find out 2 ‘s complement

= 1 0000...0100

1

-----------------------------

= 1 0000...0101

= -5

Note: Most significant bit acts as signed bit, zero mean positive, one is negative

Positive numbers will be represented directly in the memory where as negative number represented indirectly in memory in 2’s complement form.