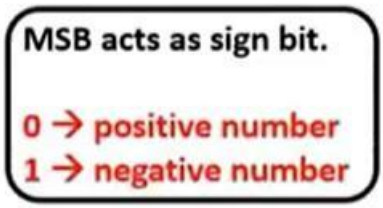
BITWISE NOT (!) /COMPLEMENT (~) OPERATOR

- This operator is denoted by ~
- The ! Operator works similarly for *Boolean* values: it reverses *Boolean* values from *true* to *false* and vice versa.
- It is a unary operator because this operator perform operation on one operand
- This operator will convert the bits from 0 to 1 and 1 to 0,it returns the one's complement representation of input value
- Applicable only for integral types not for Boolean
- It is also called as negation operator Example:

4 – 00000000000000000000000000000100 ,to binary conversion of 4 the first digit also known as sign bit i.e. 0 hence it is a positive number.

~ 4 is 11111111111111111111111111111011 here the sign bit is 1 so it is a negative number the output is shown below.



- it is important to note that bitwise complement of any integer **N** is equal to the **–(N+1)**
- let us take an example let N=5, it's complement **~N = -(N+1)**

= - (4+1) = -5

 -5 is the output that we got after compiling the program.

BIG INTEGER IN JAVA

In Java, all the bytes are stored in 2's complement method. The first bit determines whether the number is positive or negative.

BigInteger class in java used the mag property to store numbers that excess storage capacity of primitive data types.

The Biginteger class in java stores values in an array using the binary representation.

The BigInteger java groups the binary representation in 32-bit portions. This removes the limitation of the number of storage possessed by other data types in java.

BigInteger works with both numbers and strings. You can store a very large number converted to a string using a biginteger in java.