BITWISE NOT OPERATOR

NOT bitwise operator is a unary operator which performs one’s complement of the given binary number
In one’s complement all the bits of the binary number is inverted i.e it inverts 0’s to 1’s and 1’s to 0’s.
The symbol used for NOT operator is tilde(~).

Eg: ~6

The binary representation of 6 is 00000110

By performing one’s complement on the given binary number, the result is :

```
0 0 0 0 0 1 1 0 (6)
___________________
1 1 1 1 1 0 0 1 1
```

The leftmost bit is 1. So this is a negative binary number. In order to obtain the decimal value we need to perform 2’s complement on the given number upon which we get the answer as -7

The result of ~6 is -7

LOGICAL NOT OPERATOR

NOT operator is a unary operator which negates/reverses the boolean value of its operand/boolean expression i.e if the expression is true, then NOT changes this to false and vice versa.

The NOT operator is represented by exclamation mark(!).

Syntax: !(condition)

Eg: a=10 , b=20

!(a>b) returns true

Here in the above example a is not greater than b. So the answer is false. But NOT operator reverses its value to true. So the final answer is true.

CAN A NUMBER LARGER THAN LONG DATATYPE RANGE BE STORED IN JAVA?

Yes a number larger than long datatype can be stored in java using BigInteger class in java. This class is present in java.math package.

Eg; BigInteger bigNumber = new BigInteger("123456789012345678901234567890");

This is the procedure for storing large number in BigInteger class. The string representation of large number is passed to the constructor of BigInteger and is stored in object of BigInteger class called bigNumber.

BigInteger is immutable means the value cannot be changed once the value is initialized.

In order to perform various arithmetic operations on these numbers, we need to utilize the methods present in BigInteger class.

Eg: BigInteger result = bigNumber.multiply(new BigInteger("2"));

multiply is a method present in BigInteger class which used to perform multiplication. The value in the bigNumber is multiplied with 2 and is stored in BigInteger object called result. In this manner other arithmetic operations are performed using the methods of the BigInteger class.

FLOAT DATATYPE

Float datatype is used for storing the real numbers. The size of the float datatype is 32 bits. It is of single precision.
Variables of this type can have values from −3.4E38 (-3.4 * 10^{38}) to +3.4E38 (+3.4 * 10^{38}). Values are represented with approximately 7 decimal digits accuracy. The accuracy is usually low in this datatype.

Eg: float d = 3.147f;

DOUBLE DATATYPE

Double datatype is used for storing the real numbers. The size of the double datatype is 64 bits. It is of double precision.
Variables of this type can have values from −1.7E308 (-1.7 * 10308) to +1.7E308 (+1.7 * 10308). Values are represented with approximately 17 decimal digits accuracy. The accuracy is usually high in this datatype.

Eg: double e = 44.1718932449;