# 18/07/2023

# OPERATORS IN JAVA & BIGINTEGER CLASS

# ASSIGNMENT-01

# NAME:SUVARNA BALLID

EMAIL ID: Suvarnaballid726@gmail.com

# Operators in Java

Java unary operators are the types that need only one operand to perform any operation like increment, decrement, negation, etc. It consists of various arithmetic, logical and other operators that operate on a single operand.

Unary minus(-)

This operator can be used to convert a positive value to a negative one.

Syntax: -(operand)

Illustration:

a = -10

Example:

    Public main(string[] args)

    {

        Int n1 =20;

        System.out.println(“Number= “+ n1)

        n1 = -n1;

        System.out.println(“Result= “+n1)

    }

}

**Output:**

Number = 20

Result = -20

Bitwise Complement(~):

**Bitwise Not or Complement operator simply means the negation of each bit of the input value. It takes only one integer and it's equivalent to the ! operator.**

This operator changes each binary digit of the integer, which means all 0 become 1 and all 1 become 0. The ! operator works similarly for boolean values: it reverses boolean values from true to false and vice versa.

This unary operator returns the one’s complement representation of the input value or operand, i.e, with all bits inverted, which means it makes every 0 to 1, and every 1 to 0.

Syntax:

~(operand)

Illustration:

a = 5 [0101 in Binary]

result = ~5

This performs a bitwise complement of 5

~0101 = 1010 = 10 (in decimal)

Then the compiler will give 2’s complement

of that number.

2’s complement of 10 will be -6.

result = -6

Example:

Class GFG

{

    public main(String[] args)

    {

        n1=2,n2=6;

        System.out.println(“first Number =” + n1);

        System.out.println(“second number=” +n2);

        System.out.println(

            n1+” ‘ s bitwise complement =”+ ~n1);

        System.out.println(

            n2 +” ‘ s bitwise complement =’+ ~n2);

    }

}

Output:

First Number = 6

Second Number = -2

6's bitwise complement = -7

BigInteger Class:

As we know, the biginteger class is used for mathematical operations which involve very big integer calculations larger than the primitive *long* type. It represents immutable arbitrary-precision integers.

Before going further, let's remember that in Java all bytes are represented in the compliment system using big integer notation. It stores the most significant byte of a word at the smallest memory address (the lowest index). Moreover, the first bit of the byte is also a sign bit. Let's inspect example byte values:

* *1000 0000* represents *-128*
* *0111 1111*represents 127
* *1111 1111* represents *-1*

Big Integer class extends Number and implements Comparable interface. It provides analogues to all of Java's primitive integer operators and all methods from java.lang.Math package.

|  |
| --- |
| Import java.math.BigInteger;    public class integer        public static void main(String[] args) {     BigInteger b1 = new BigInteger(“2984732865843857834”);   BigInteger b2 = new BigInteger(“938738759”);            BigInteger product = b1.multiply(b2);          BigInteger division = b1.divide(b2);            System.out.println(“product=”+ product);          System.out.println(“division =”+ division);        }  } |