

UNARY PLUS

Unary + operator is used to indicate the positive value. However, numbers are positive without this also. It follows the following rules of mathematics.

- 1. Minus and Minus makes Plus
- 2. Minus and Plus makes Minus
- 3. Plus and Minus makes Minus
- 4. Plus and Plus makes Plus

```
Syntax: result = + operand
```

```
EXAMPLE:
```

```
public class UnaryDemo
{
  public static void main(String[] args)
  {
   int a=8,b=-2;
   int result;
   System.out.println("Value of a is "+a);
   result=+a;
   System.out.println("After unary plus operation result is "+result);
   System.out.println("Value of b is "+b);
   result=+b;
   System.out.println("After unary plus operation result is "+result);
}
```

OUTPUT:

Value of a is 8

After unary plus operation result is 8

Value of b is -2

After unary plus operation result is -2

UNARY MINUS

Unary - operator is used to convert a positive value into a negative value. It also follows the following rules of mathematics.

- 1. Minus and Minus makes Plus
- 2. Minus and Plus makes Minus
- 3. Plus and Minus makes Minus
- 4. Plus and Plus makes Plus

```
Syntax: result = - operand
```

```
EXAMPLE:
```

```
public class UnaryDemo
{
  public static void main(String[] args)
  {
   int a=8,b=-2;
   int result;
   System.out.println("Value of a is "+a);
   result=-a;
   System.out.println("After unary minus operation result is "+result);
   System.out.println("Value of b is "+b);
   result=-b;
   System.out.println("After unary minus operation result is "+result);
}
```

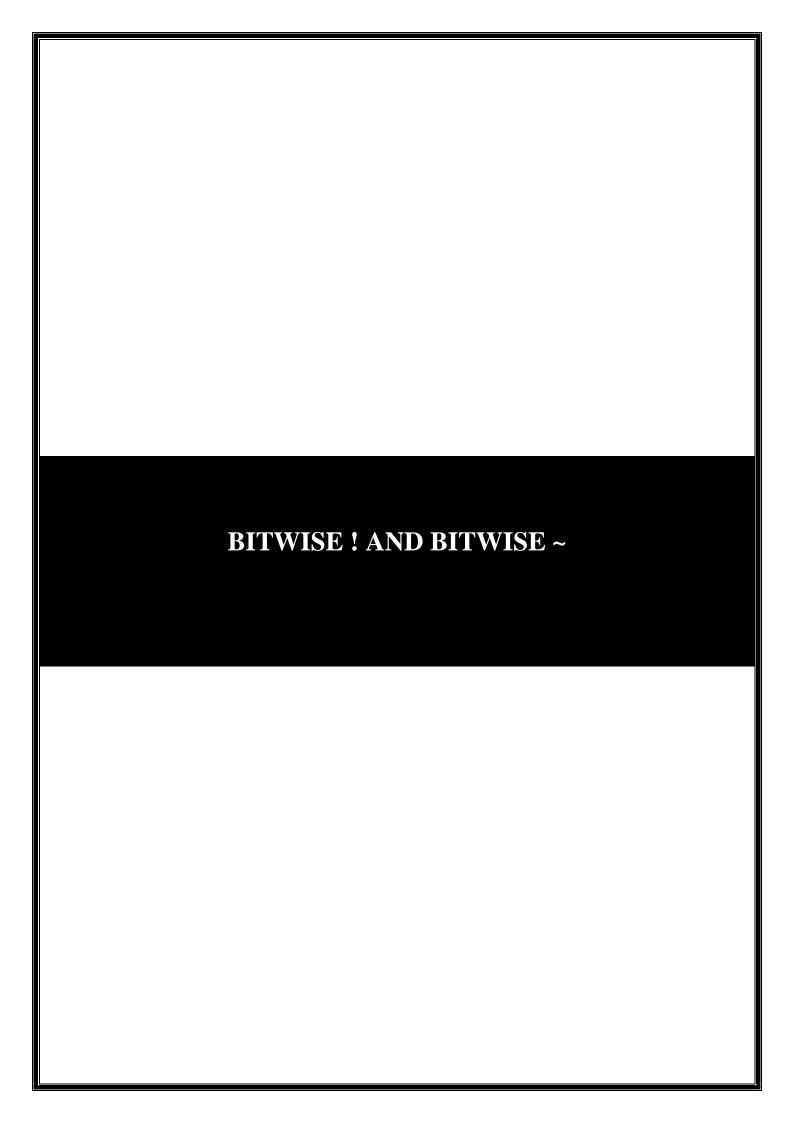
OUTPUT:

Value of a is 8

After unary minus operation result is -8

Value of b is -2

After unary minus operation result is 2



BITWISE! AND BITWISE ~

- Bitwise Complement operator simply means the negation of each bit of the input value. It takes only one integer and it's equivalent to ! 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.

INPUT:

n = 2

Binary form of 2 = 0010

Bitwise complement operation on $2 = \sim 0010$

= 1101

1101 is equivalent to decimal value 13.

Expected output: 13

Correct Output: -3

The compiler returns the 2's complement of the input value.

EXAMPLE:

```
class GFG
{
public static void main (String[] args)
{
int a = 2;
System.out.println("Bitwise complement of " +
a + " : " + ~a);
}
}
```

OUTPUT:

Bitwise complement of 2: -3

EXPLINATION:

Bitwise complement Operation of 2 (~ 0010): 1101

Calculate 2's complement of 3:

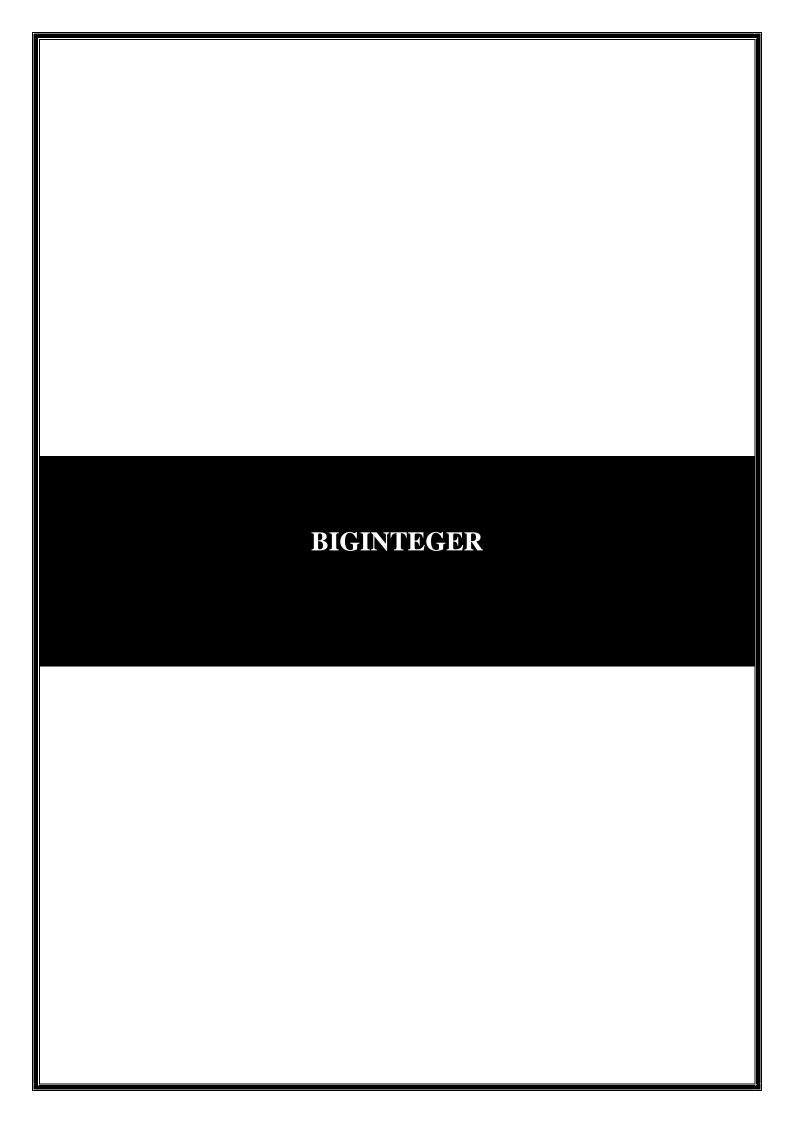
Binary form of 3 = 0011

1's Complement of 3 = 1100

Adding 1 to 1's complement = 1100 + 1

2's complement of 3 = 1101

The bitwise Complement of 2 is same as the binary representation of -3



- BigInteger class is used for the mathematical operation which involves very big integer calculations that are outside the limit of all available primitive data types.
- In this way, BigInteger class is very handy to use because of its large method library and it is also used a lot in competitive programming.
- BigInteger is part of the java.math package. BigInteger allows you to perform arithmetic operations and other manipulations on arbitrarily large integers.