

Assignment-1

Bitwise operator

A **bitwise operator in Java** is a symbol/notation that performs a specified operation on standalone bits, taken one at a time. It is used to manipulate individual bits of a binary number and can be used with a variety of integer types – char, int, long, short, byte.

Bitwise Left Shift

The bits in the left operand are shifted to the left by the number of places specified in the right operand.

Symbol

<<

Example:

```
class Main {  
    public static void main(String[] args) {  
        int number = 2; // 2 bit left shift operation  
        int result = number << 2;  
        System.out.println(result);  
    }  
}
```

Output:

8

RIGHT SHIFT OPERATOR:

There are two types of right shift operators:

1. Signed Right Shift (>>)
2. Unsigned Right Shift (>>>)

SIGNED RIGHT SHIFT (A >>2)

This **bitwise operator in Java** shifts the bit pattern to the right by the specified number of places. LSBs are discarded, and the MSBs are filled with the sign bit. 0 represents a positive sign, while 1 represents a negative sign.

Syntax

```
class Main {  
  
    public static void main(String[] args) {  
  
        int operand1 = 2;  
  
        int operand2 = -2; // 2 bit signed right shift  
  
        System.out.println(operand1 >> 2); // prints 1  
  
        System.out.println(operand2 >> 2); // prints -1  
  
    }  
  
}
```

UNSIGNED RIGHT SHIFT (A >>>2)

The operation here is exactly similar to SIGNED RIGHT SHIFT except that the vacant left-most bits are replaced with 0 instead of the sign bits.

Syntax

```
class Main {  
  
    public static void main(String[] args) {
```

```
int operand = 240; // binary number – 1111 0000

// 2 bit unsigned right shift

System.out.println(operand >>> 2); // prints 60

}

}
```

Sorry for the late submission and Thank you.