Object Oriented Programming with Java (OOPJ)

Session 3: Operators & Basics

Kiran Waghmare

Arithmetic Operators

Operator	Description	Example	Output
+	Addition	10 + 5	15
-	Subtraction	10 - 5	5
*	Multiplication	10 * 5	50
/	Division	10 / 5	2
%	Modulus (Remainder)	10 % 3	1

Relational Operators

Operator	Description	Example	Output
==	Equal to	10 == 5	false
!=	Not equal to	10 != 5	true
>	Greater than	10 > 5	true
<	Less than	10 < 5	false
>=	Greater than or equal to	10 >= 5	true
<=	Less than or equal to	10 <= 5	false

Logical Operator

Operator	Description	Example	Output
&&	Logical AND	(10 > 5) && (5 < 10)	true
||	Logical OR	(10 > 5) || (5 < 10)	true
!	Logical NOT	!(10 > 5)	false

Assignment Operator

Operator	Description	Example	Equivalent
=	Assign	x = 5	x = 5
+=	Add and assign	x += 5	x = x + 5
-=	Subtract and assign	x -= 5	x = x - 5
*=	Multiply and assign	x *= 5	x = x * 5
/=	Divide and assign	x /= 5	x = x / 5
%=	Modulus and assign	x %= 5	x = x % 5

Bitwise Operator

Operator	Description	Example	Output
&	Bitwise AND	5 & 3 (0101 & 0011) 0001	1
|	|	Bitwise OR	`5
^	Bitwise XOR	5 ^ 3 (0101 ^ 0011)	6
~	Bitwise NOT	~5 (~0101)	-6
<<	Left Shift	5<<1 5<< 3	10
>>	Right Shift	5>>1 2 5	2

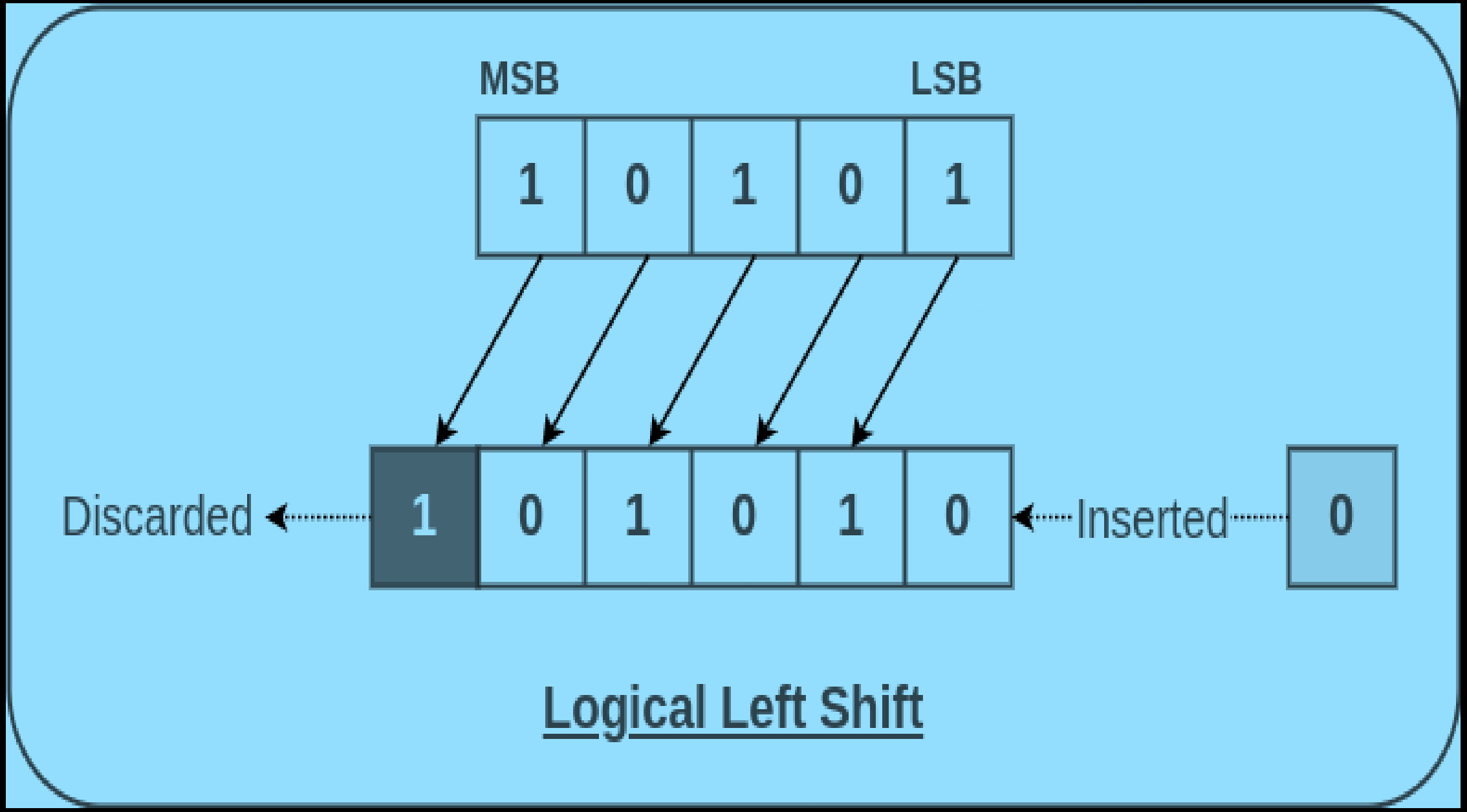
Bitwise Shift Operator

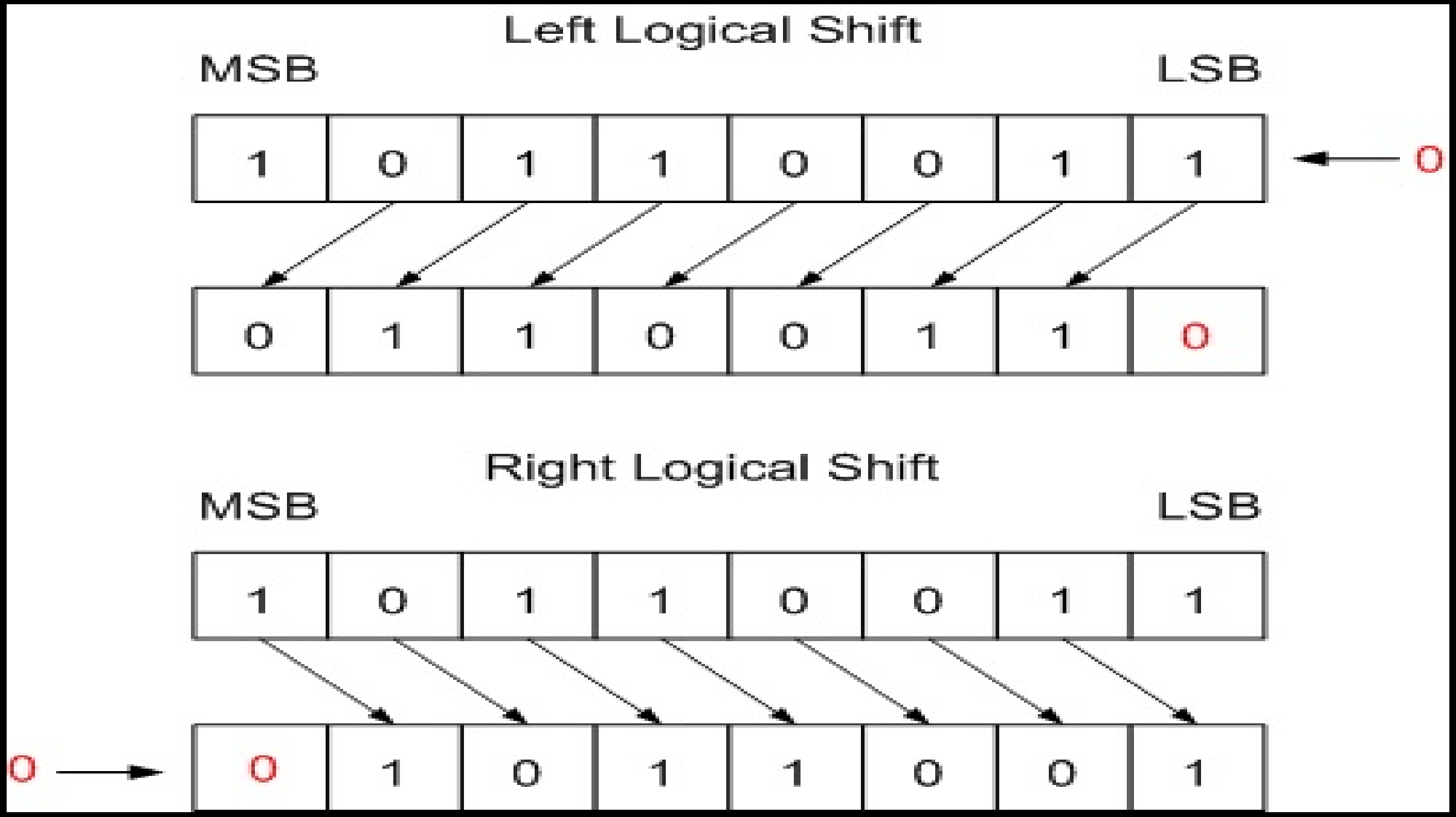
Operator	Operation	Zero Fill?	Used For
x << n = 1, 2	Left Shift	Yes (right-side)	Multiplication by 2^n
x >> n = 1	Right Shift	No (preserves sign bit)	Division by 2^n
>>>	Unsigned Right Shift	Yes (left-side)	Handling unsigned data

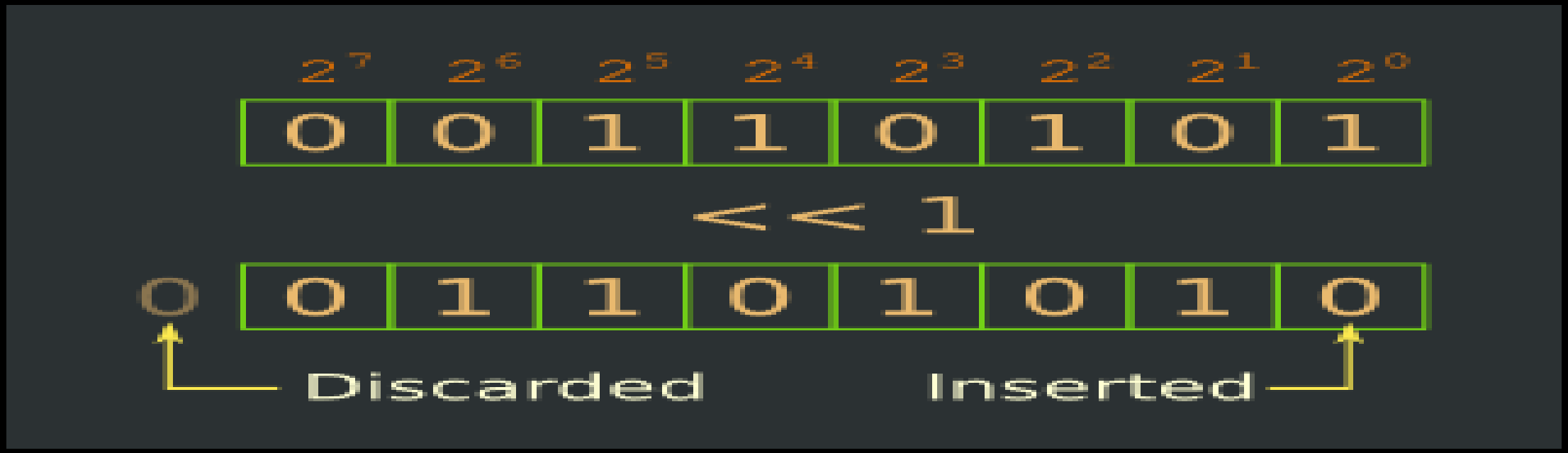
Sign bit Magnitude

1	0	0	0	0	1	0	0	-4

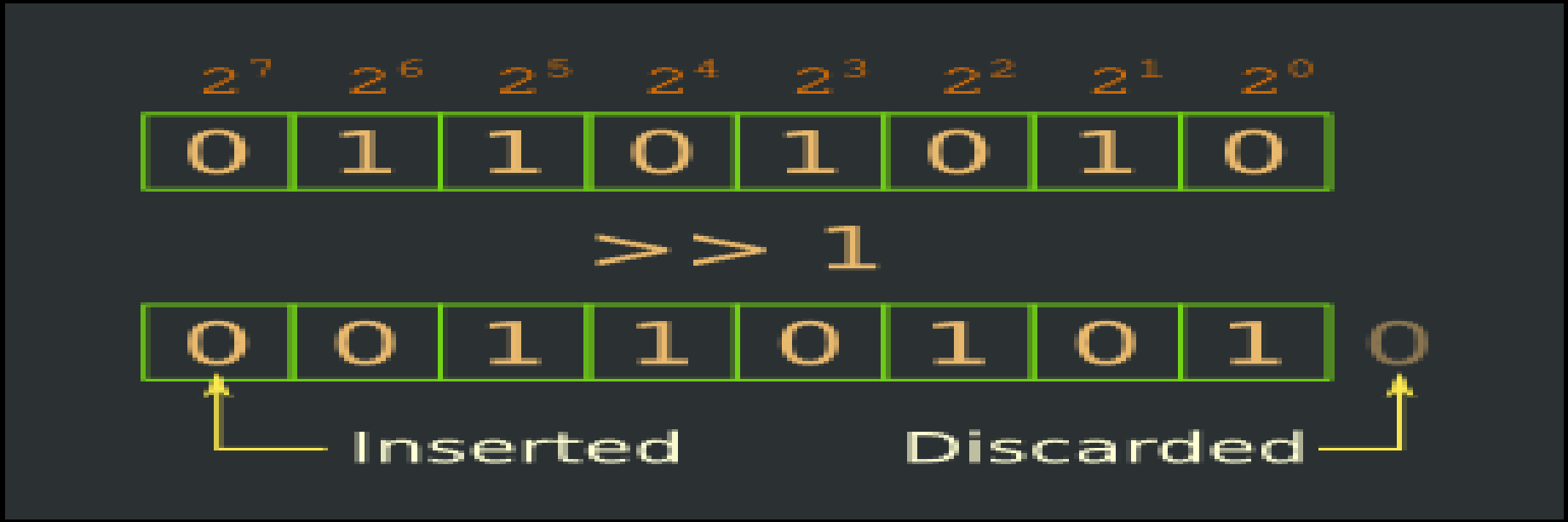
0	0	0	0	0	1	0	0	+4


MSB
LSB
1 0 1 0 1
Discarded
1 0 1 0 1 0
Inserted
0
Logical Left Shift


Left Logical Shift
MSB
LSB
1 0 1 1 0 0 1 1
0
0 1 1 0 0 1 1 0
Right Logical Shift
MSB
LSB
1 0 1 1 0 0 1 1
0
0 1 0 1 1 0 0 1



00110101 << 1 = 01101010 // Left shift by 1 position
10101101 << 1 = 01011010 // Left shift by 1 position
10101101 << 2 = 10110100 // Left shift by 2 positions



00110101 >> 1 = 00011010 // Right shift by 1 position
10101101 >> 1 = 01010110 // Right shift by 1 position
10101101 >> 2 = 00101011 // Right shift by 2 positions

```java
class OperatorDemo {
    public static void main(String[] args) {
        int a=10;
        int b=4;
        //Arithmetic
        System.out.println(a+b);
        //Relational
        System.out.println(a<b);
        //Logical
        boolean x = true, y=false;
        System.out.println(x && y);
        System.out.println(x || y);
        System.out.println(!x);
        //Assignment
        //System.out.println(x+=5); //Error:
        System.out.println(a+=5);
        //Bitwise
        int m=5,n=3;
        System.out.println(m & n);

    }
}
```

0101 → 5
0011 → 3

0001 → 1

```java
class OperatorDemo {
    public static void main(String[] args) {
        int a=10;
        int b=4;
        //Arithmetic
        System.out.println(a+b);
        //Relational
        System.out.println(a<b);
        //Logical
        boolean x = true, y=false;
        System.out.println(x && y);
        System.out.println(x || y);
        System.out.println(!x);
        //Assignment
        //System.out.println(x+=5); //Error:
        System.out.println(a+=5);
        //Bitwise
        int m=5,n=3;
        System.out.println(m & n);

    }
}
```

boolear

bit

```java
class OperatorDemo {
    public static void main(String[] args) {
        int a=10;
        int b=4;
        //Arithmetic
        System.out.println(a+b);
        //Relational
        System.out.println(a<b);
        //Logical
        boolean x = true, y=false;
        System.out.println(x && y);
        System.out.println(x || y);
        System.out.println(!x);
        //Assignment
        //System.out.println(x+=5); //Error:
        System.out.println(a+=5);
        //Bitwise
        int m=5,n=3;
        System.out.println(m & n);
        System.out.println(m | n);
        System.out.println();

    }
}
```

$2^3 = 8 \times 0 = 0$

$2^2 = 4 \times 1 = 4$

$2^1 = 2 \times 0 = 0$

$2^0 = 1$

```
5 = 0101
3 = 0011
---------------
 7 =  0111
```

```
---------
1. Arithmetic Operators
2. Relational Operators
3. Logical Operators
4. Bitwise Operator
5. Assignment Operator
6. Bit shift Operator

    1)<< (Left Shift):
    -shifts bits to the left, filling zeros on the right.
    -a << n, a= number , n=no of bits.

    2)>> (Right Shift):
    -shifts bits to the right, filling zeros/ones on the right based on positive and negative
    -a >> n, a= number , n=no of bits.

    3) >>> (Unsigned Right Shift):
    -Always fills with 0, even for negative numbers.

------------------------
7. Ternary Operator

syntax:

Variable = Exp1 ? Exp2 : Exp3

8. Increment DEcrement
9. Instance Oprator
```

Var = Expression ? True : False

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

        for(int i=0;i<=5;i++)
```

start loop

terminate loop

increment/decrement

```
    }
}
```

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

        for(int i=0;i<=5;i++){
            System.out.println(i);
            if(i==3)
                break;
            System.out.println(i);
        }

    }
}
```

0 | 2 3

0 | 2

C:\WINDOWS\system32

```
C:\Test>javac SwitchDemo1.java

C:\Test>javac ForDemo.java

C:\Test>java ForDemo
0
0
1
1
2
2
3

C:\Test>
```

```
class CmdArgs {
    public static void main(String[] args) {
        int i =10;//compile time
        //user input: input required at run time

    }
}
```

data type

array (String)
-int
-char
-float
-String

args[0]

index

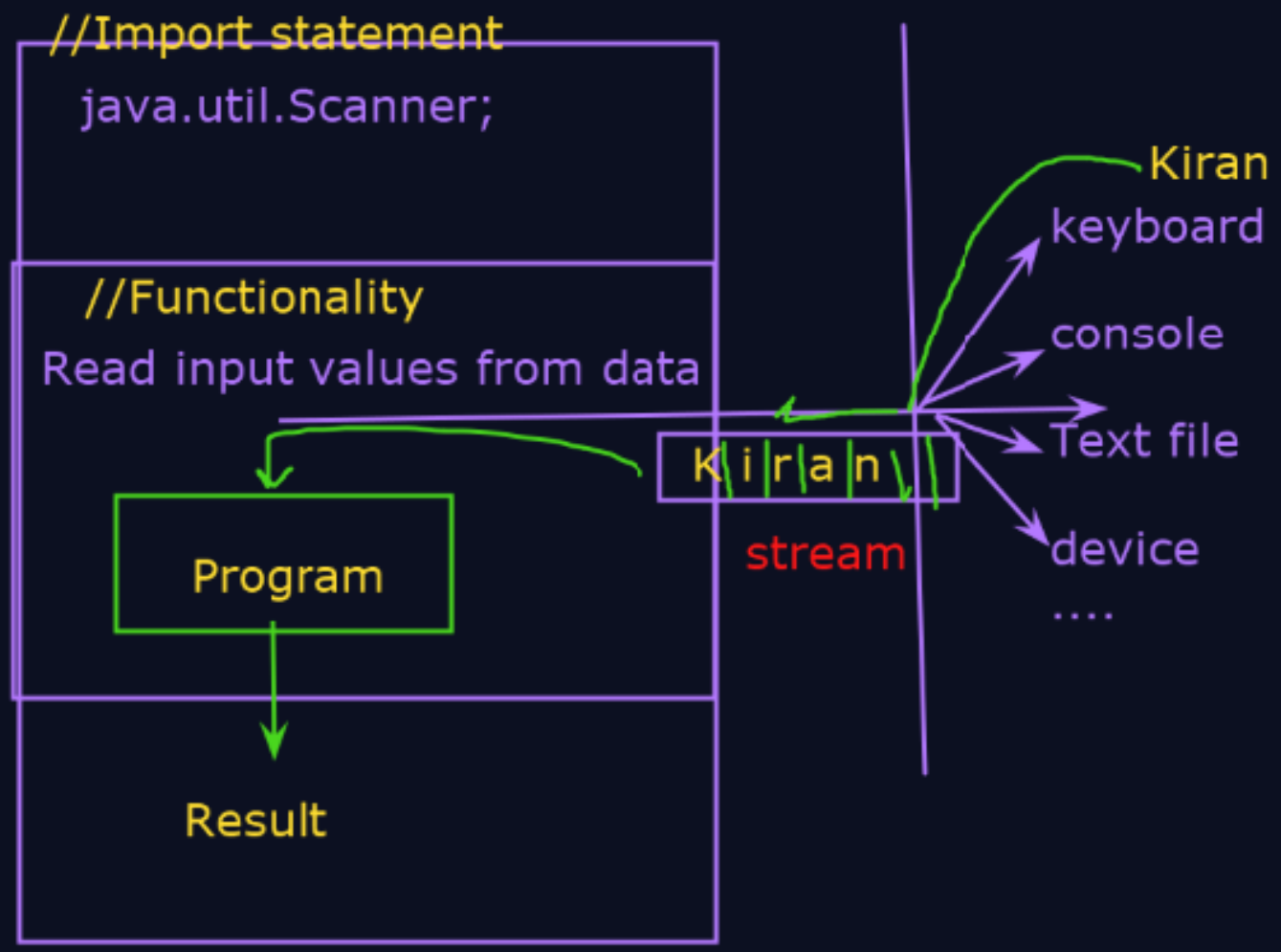
```
args[0] = 10.3443
args[1] = sjdfjdfj
```

```java
class CmdArgs {
    public static void main(String[] args) {
        int i =10;//compile time
        //user input: input required at run time

        String s1 = args[0];
        String s2 = args[1];
        System.out.println(s1);
        System.out.println(s2);

    }
```

C:\WINDOWS\system32

```
2
3
4
4
5
5

C:\Test>javac CmdArgs.java

C:\Test>java CmdArgs Hello CDAC
Hello
CDAC

C:\Test>
```

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

        System.out.println(i+j);

    }
}
```



Reading Different Types of Input

Reading Different Types of Input Method	Reads	Example Input
nextInt()	Integer	10
nextDouble()	Double (decimal)	3.14
nextFloat()	Float (decimal)	5.75
nextLong()	Long Integer	123456789
nextBoolean()	Boolean	true / false
next()	Single word	"Hello"
nextLine()	Full line (including spaces)	"Hello World"

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

        System.out.println(i+j);
Scanner sc = new Scanner(System.in);
```

```
Integer     = nextInt()
Double      = nextDouble()
Float       = nextFloat()
Long        =nextLong()
Boolean     = nextBoolean()
Char/word   = next()
sentencemltiple words = nextLine()

String s = sc.nextLine()
```

```
}
sc.close();
```

//Import statement

java.util.Scanner;

//Functionality

Read input values from data

Program

Result