Binary Number System, Java Operators & Taking User Input

In This Lecture

1. Binary Number System
2. Operators in Java
3. Taking User Input

Binary Number System

Convert Decimal To Binary

10

$\boxed{0}$ $\boxed{1}$

→ 0101001 →

2	49	1
2	24	0
2	12	0
2	6	0
2	3	1
	1	

49 → $(110001)_2$

5 → 101

6 → 110

2	26	0
2	13	1
2	6	0
2	3	1
	1.	

$(26)_{10}$

$(11010)_2$

Binary Number System

Convert Binary To Decimal

5	4	3	2	1	0
1	1	0	0	1	0
$*\ 2^5$	$*\ 2^4$	$*\ 2^3$	$*\ 2^2$	$*\ 2^1$	$*\ 2^0$

$$32 + 16 + 0 + 0 + 2 + 0$$

$$= 50$$

$$(110010)_2 = (50)_{10}$$

$$\overset{4\ 3\ 2\ 1\ 0}{(10101)_2} = (21)_{10}$$

$$16 + 4 + 1$$

```
int age = 21;
```

Binary Number System

Convert Binary To Decimal

```
int age = 128;
byte newAge = (byte) age;
```

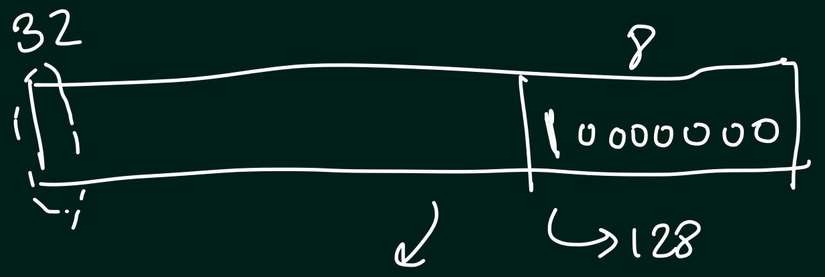
↳ −128

1 Byte → 8 bits

MSB 7	6	5	4	3	2	1	0 LSB
1	0	0	0	0	1	0	1

↳ 1 → −ve

↳ 0 → +ve



01111111

↓

+127

↓

−ve

10000000

↓

−128

```
  10000000
  01111111
     +  1
  ---------
  10000000
```

Byte

Binary Addition

```
   1
  00018
+ 23
------
   41
```

```
   1
  0101  → 5
  1001  → 9
------
  1110  → 14
```

```
  10
  111
  01110  → 14
  01011  → 11
------
  11001  → 25
```

```
  1 1
  1 1 1 1
+ 1 1 1 0
---------
  1 1 1 0 1
```

Binary Subtraction

$9 - 4 = 5$

$9 + (-4) = 5$

$(-4)_{10} \rightarrow (\quad)_2$

$$\begin{array}{r} 1001 \\ -\ 100 \\ \hline \end{array}$$

2's Complement

1. swap the bits
2. add 1

$$\begin{array}{r} 00001001 \\ +\ 11111100 \rightarrow (-4) \\ \hline 00001101 \\ \hline \end{array}$$

00000100

(1) 11111011

(2) $\quad + 1$

11111100

Binary Subtraction

$13 - 6 = 7$

1101

110 → 0001 1 0

↓ ①

1 11001

↓ ②

1001

\+ 1

1 1 11010

$\boxed{\times}$ 1 1

1101

11110101 0

00001 0 111 → 7

Types of Operators in Java

1. Arithmetic Operators
2. Assignment Operators
3. Relational Operators
4. Logical Operators
5. Unary Operators
6. Bitwise Operators

1. Arithmetic Operators

Operator	Operation
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulo Operation (Remainder after division)

$$\begin{array}{r} 1 \\ 7\overline{)12} \\ 7 \\ \hline 5 \end{array}$$

$12 \% 7 = 5$

$12 / 7 = 1$

2. Assignment Operators

Operator	Example	Equivalent to
`=`	`a = b;`	`a = b;`
`+=`	`a += b;`	`a = a + b;`
`-=`	`a -= b;`	`a = a - b;`
`*=`	`a *= b;`	`a = a * b;`
`/=`	`a /= b;`	`a = a / b;`
`%=`	`a %= b;`	`a = a % b;`

3. Relational Operators → always return boolean value → true/ false

Operator	Description	Example
`==`	Is Equal To	`3 == 5` returns **false**
`!=`	Not Equal To	`3 != 5` returns **true**
`>`	Greater Than	`3 > 5` returns **false**
`<`	Less Than	`3 < 5` returns **true**
`>=`	Greater Than or Equal To	`3 >= 5` returns **false**
`<=`	Less Than or Equal To	`3 <= 5` returns **true**

4. Logical Operators

Operator	Example	Meaning
`&&` (Logical AND)	expression1 **&&** expression2	`true` only if both `expression1` and `expression2` are `true`
`||` (Logical OR)	expression1 **||** expression2	`true` if either `expression1` or `expression2` is `true`
`!` (Logical NOT)	**!**expression	`true` if `expression` is `false` and vice versa

a	b	y = a AND B
0	0	0
0	1	0
1	0	0
1	1	1

AND

y = a OR b
0
1
1
1

OR

a	$\tilde{a}$
0	1
1	0

5. Bitwise Operators

Operator	Description
~	Bitwise Complement
<<	Left Shift
>>	Right Shift
>>>	Unsigned Right Shift
&	Bitwise AND
^	Bitwise exclusive OR

5 → 101 >>
<<
&
|

Bitwise operation
↳ Bit manipulation

Other Operators

Increment/ Decrement Operators

++ --

a++; → a=a+1

a--;

↳ a=a-1;

Ternary Operators

? :

if else

Taking User Input using Scanner

In order to use the object of Scanner, we need to import java.util.Scanner package.

Class

```
Scanner sc = new Scanner(System.in);
```

- sc.nextInt()
- sc.nextFloat()
- sc.nextLine()
- sc.next()

Various Input Types using Scanner

We can use nextLong(), nextFloat(), nextDouble(), and next() methods to get long, float, double, and string input respectively from the user.

Note: It is recommended to close the scanner object once the input is taken using the close() method