

Bit Manipulation

▼ Bits : 0 1

- Decimal to Binary :

Binary Number System 0 & 1

Decimal to Binary (4)

↓
4

2	4	0
2	2	0
	1	

↑

$(100)_2 = (4)_{10}$

Diagram illustrating the conversion of the decimal number 4 to binary:

- 4 is divided by 2, resulting in 2 and remainder 0.
- 2 is divided by 2, resulting in 1 and remainder 0.
- 1 is divided by 2, resulting in 0 and remainder 1.

The remainders, read from bottom to top, form the binary number 100.

Verification: $1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 4$

- Binary to Decimal :

Binary to Decimal (100)

$$\begin{array}{cccc}
 1 & 0 & 0 & 1 \\
 \downarrow & & & \downarrow \\
 1 \times 2^3 & + & 0 \times 2^2 & + & 0 \times 2^1 & + & 1 \times 2^0 \\
 \downarrow & & & & & & \\
 8 & + & 0 & + & 0 & + & 1 = (9)_{10} = (1001)_2
 \end{array}$$

numbers, chars, strings - everything converted

- Basic Binary

Binary Number System 0 & 1

Decimal to Binary (4)

0 → 000

1 → 001

2 → 010

3 → 011

4 → 100

5 → 101

6 → 110

7 → 111

8 → 1000

Binary to Decimal (100)



▼ Bit-wise Operators

- And Operator
 - Both values needs to be 1 for the final output to be one

Binary AND &

Rules

0	&	0	0
0	&	1	0
1	&	0	0

1	&	1	1
---	---	---	---

5 & 6

A = 0101

B = 0110

1	0	1	0
&	1	1	0
<hr/>			
1	0	0	0

= (4)₁₀

5 + 6 = 11
5 & 6 = 4



- OR Operator
 - Even if one of the values is 1 the output would be one

Binary OR |

Rules

0		0	0
0		1	1
1		0	1
1		1	1

5 | 6

A = 0101 B = 0110

$$\begin{array}{r}
 101 \\
 110 \\
 \hline
 111 =
 \end{array}$$



- XOR ^ Operator
 - Both the values need to be different then the output would be 1 otherwise if both the values are same the output would be 0

Binary XOR ^

Rules

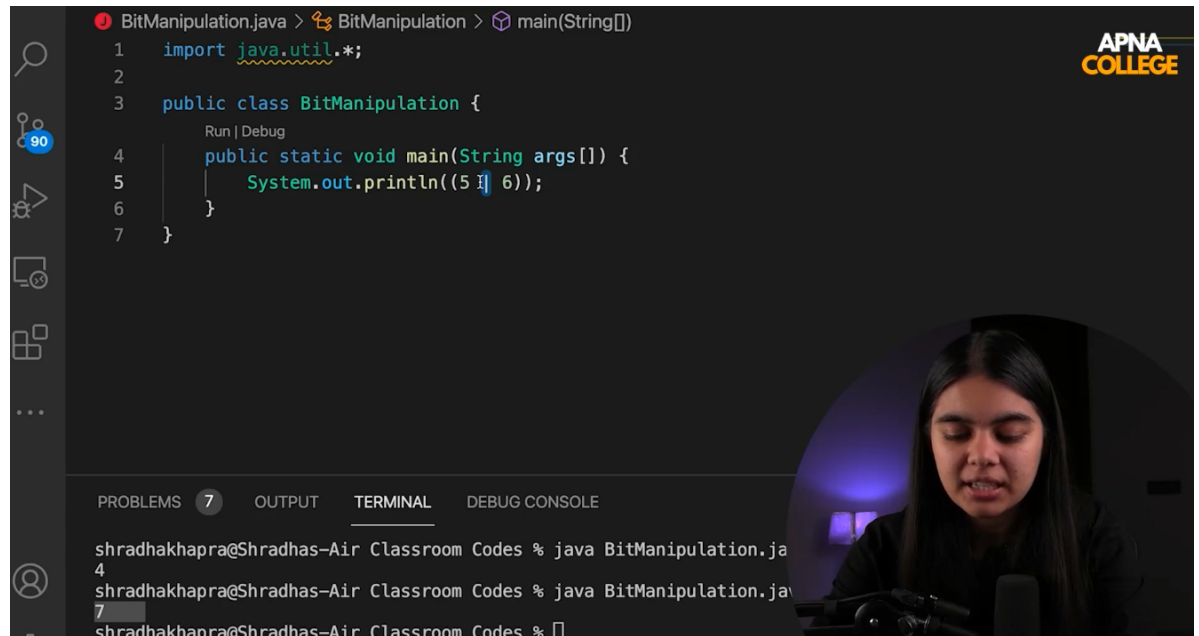
0	^	0	0
0	^	1	1
1	^	0	1
1	^	1	0

5 ^ 6

A = 0101 B = 0110

$$\begin{array}{r}
 101 \\
 ^ 110 \\
 \hline
 011 = (3)_{10}
 \end{array}$$

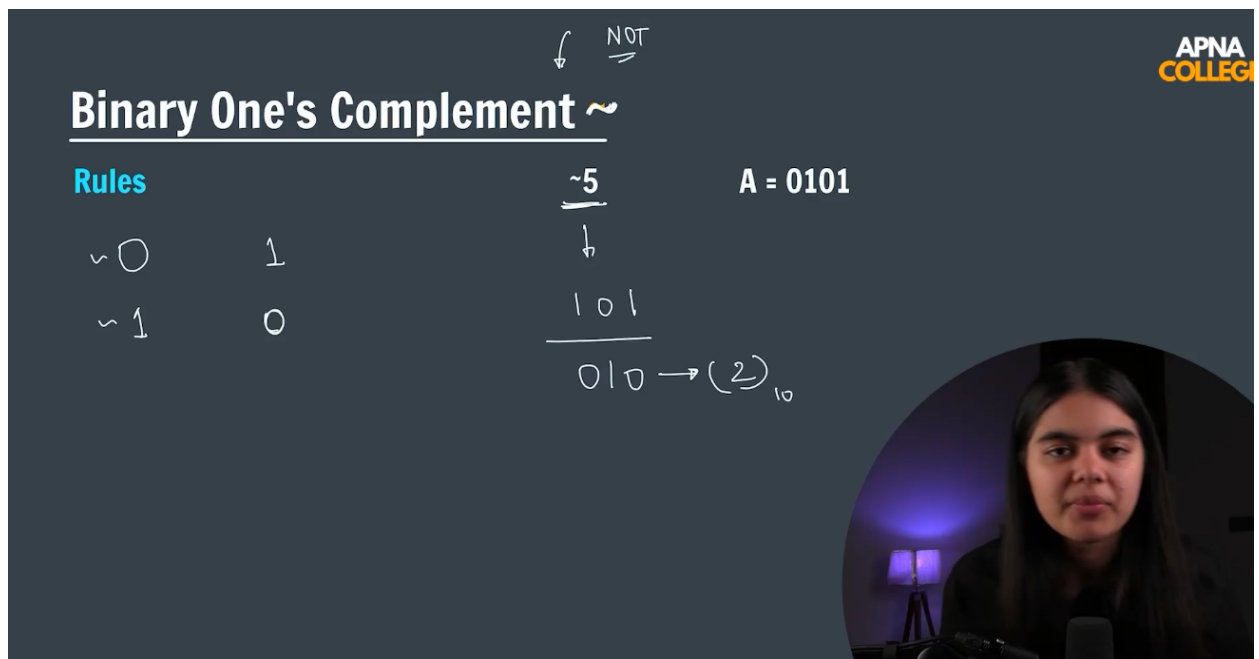




```
BitManipulation.java > BitManipulation > main(String[])
1  import java.util.*;
2
3  public class BitManipulation {
4      public static void main(String args[]) {
5          System.out.println((5 ^ 6));
6      }
7  }
```

shradhakhapra@Shradhas-Air Classroom Codes % java BitManipulation.java

- Binary One's Complement (Not Operator):
 - Changes 0 to 1 in the binary form and then the output is displayed in decimal form
 - It is only used for one value



Binary One's Complement ~

Rules

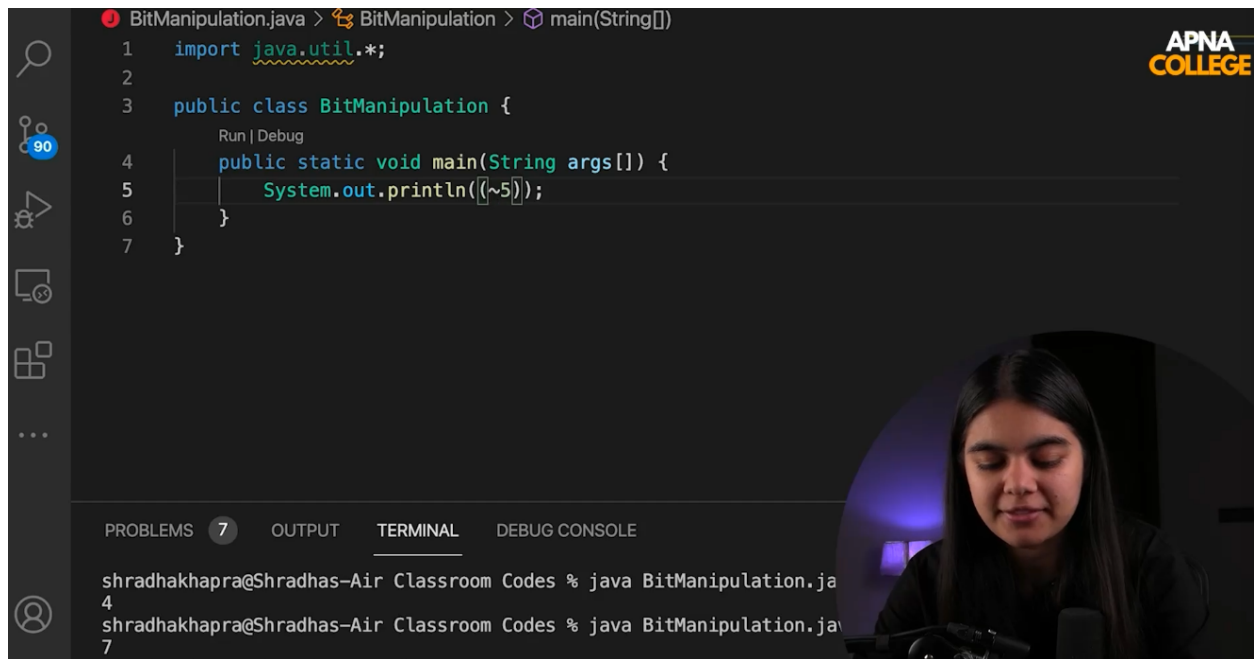
~ 0	1
~ 1	0

~ 5

$A = 0101$

101

$010 \rightarrow (2)_{10}$



```
BitManipulation.java > BitManipulation > main(String[])
1  import java.util.*;
2
3  public class BitManipulation {
4      public static void main(String args[]) {
5          System.out.println(Integer.toBinaryString(5));
6      }
7  }
```

shradhakhapra@Shradhas-Air Classroom Codes % java BitManipulation.java

- Binary Left Shift : The pointy side of the operator <<
 - This values expressed would be removed from the binary number sequence , the numbers are then shifted towards the side from where the numbers were removed
 - Then the empty spaces at the other side of the sequence would be filled with zero.
 - Shortcut : $a \ll b = a * 2^b$

- Binary Right Shift >> :

- $a \gg b = a / 2^b$

▼ Check if a number is ODD or Even

-

Question 1

Check if a number is **Odd or Even**

0 = 000

1 = 001

2 = 010

3 = 011

4 = 100

5 = 101

9 → odd

↓

1001

4 0001

0001

$(1)_{10} \neq 0$

↓
odd

$$\begin{array}{r} 3 \\ 011 = n \\ \& 001 \\ \hline 001 \\ \hline = (1)_{10} \\ \downarrow \\ \text{odd} \end{array}$$

$$\begin{array}{r} 4 \\ 100 = n \\ \& 001 \\ \hline 000 \\ \hline = (0)_{10} \end{array}$$



```
BitManipulation.java > BitManipulation
1  import java.util.*;
2
3  public class BitManipulation {
4      public static void oddOrEven(int n) {
5          int bitMask = 1;
6          if((n & bitMask) == 0) {
7              //even number
8              System.out.println("even number");
9          } else {
10             System.out.println("odd number");
11         }
12     }
13
14     public static void main(String args[]) {
15         oddOrEven(3);
16         oddOrEven(11);
17         oddOrEven(14);
18     }
19 }
```

Run | Debug

shradhakhapra@Shradhas-Air Classroom Codes % java BitManipulation.java

odd number
odd number
even number

