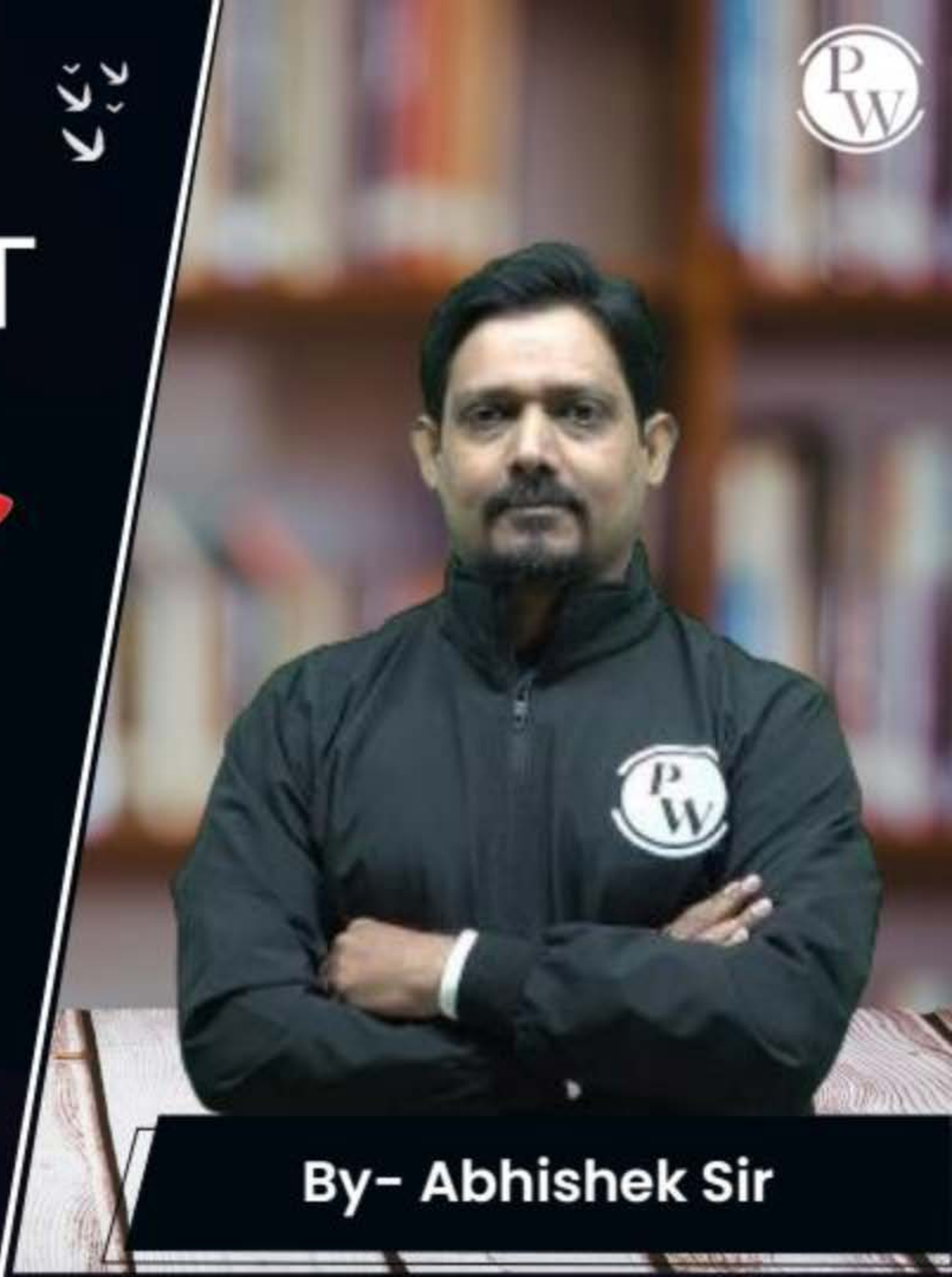
Computer Science & IT

C Programming

Data type & Operator

Lecture No. 04



By- Abhishek Sir

Recap of Previous Lecture

Topic	Relational operator
Topic	Logical operator
Topic	Short circuit code
Topic	Bitwise operator ~, >> << &, ^, |
Topic	

>>= <<= precedence higher != ==

Slide

Topics to be Covered

- Topic — Shift Left , Shift Right
- Topic — comma operator
- Topic — printf returnvalue / scanf return value
- Topic — Ternary operator
- Topic

Slide

Associativity Rule

Precedence	Operator	Description	Associativity
1	`++ --`	Suffix/postfix increment and decrement	Left-to-right
	`()`	Function call	
	`[]`	Array subscripting	
	`.`	Structure and union member access	
	`->`	Structure and union member access through pointer	
	`(type){list}`	Compound literal(C99)	
2	`++ --`	Prefix increment and decrement	Right-to-left
	`+ -`	Unary plus and minus	
	`! ~`	Logical NOT and bitwise NOT	
	`(type)`	Type cast	
	`*`	Indirection (dereference)	
	`&`	Address-of	
	`sizeof`	Size-of	
	`_Alignof`	Alignment requirement(C11)	

Associativity Rule

3	* / %	Multiplication, division, and remainder	Left-to-right
4	+ −	Addition and subtraction	
5	<< >>	Bitwise left shift and right shift	
6	< <= > >=	For relational operators < and ≤ respectively For relational operators > and ≥ respectively	
7	== !=	For relational = and ≠ respectively	
8	&	Bitwise AND	
9	^	Bitwise XOR (exclusive or)	
10	|	Bitwise OR (inclusive or)	
11	&&	Logical AND	
12	||	Logical OR	

Associativity Rule

13	?:	Ternary conditional	Right-to-Left
14	=	Simple assignment	
	+= -=	Assignment by sum and difference	
	*= /= %=	Assignment by product, quotient, and remainder	
	<<= >>=	Assignment by bitwise left shift and right shift	
	&= ^= |=	Assignment by bitwise AND, XOR, and OR	
15	,	Comma	Left-to-right

Bit-wise Operator

```
#include<stdio.h>
   int main(){
            char a = 8;
            int k;
            k =a<<3;
            printf("%d", k);
            return 0;
   }
```

(A) 1 (B) 64

(C) 20 (D) -6

<< Shift Left

$a<<3 = a*2^3$

$= 8 \times 2^3 = 64$

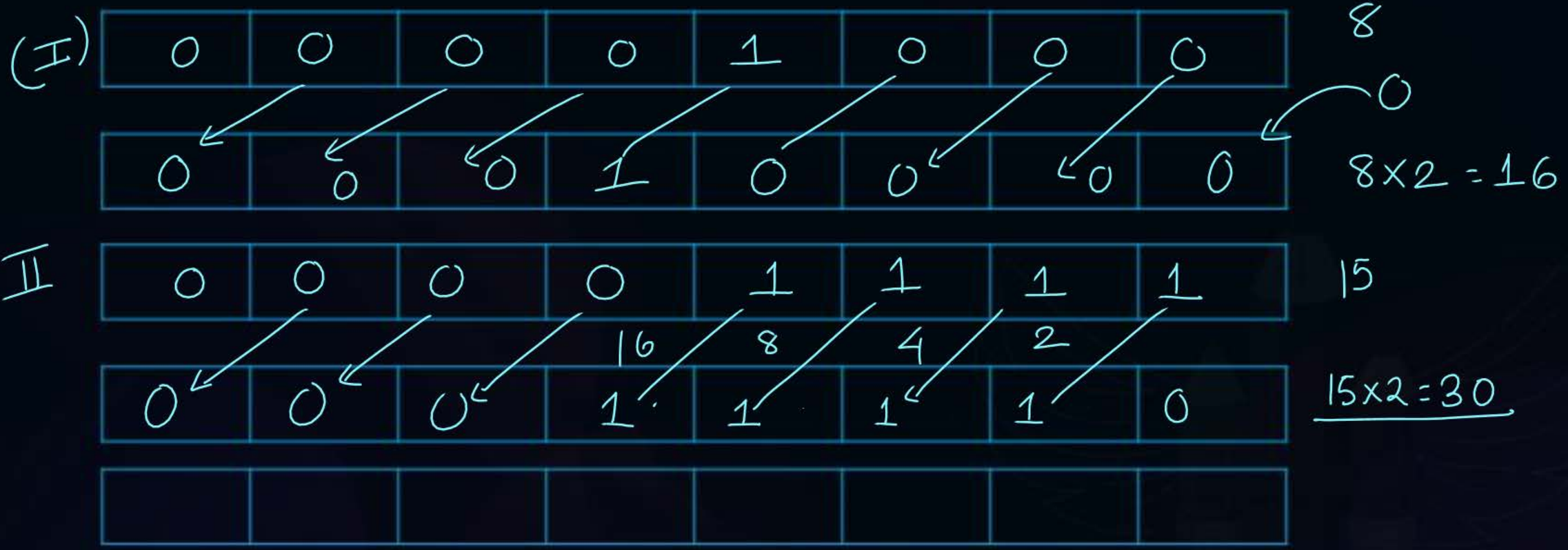
It may lead to overflow

Slide

```
int main() {

    char a = 64;

    char k = a << 1;

    printf("%d", k);

    return 0;

}
```

-128

Bit-wise Operator



Slide

Bit-wise Operator

$-1 \times 2^7 + 0 + 0 \cdots + 0 = -1 \times 128 = -128$

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

"%d" (-128)

II

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

$-1 \times 2^3 + 1 = -7$

$-1 \times 2^4 + 1 \times 2^1$

$= -16 + 2 = -14$

Slide

Bit-wise Operator

```
#include<stdio.h>
   int main(){
            char a = 64;
            int k;
            k =a>>3;
            printf("%d", k);
            return 0;
   }
```

$a >> k = \left\lfloor \frac{a}{2^k} \right\rfloor$ ← Negative

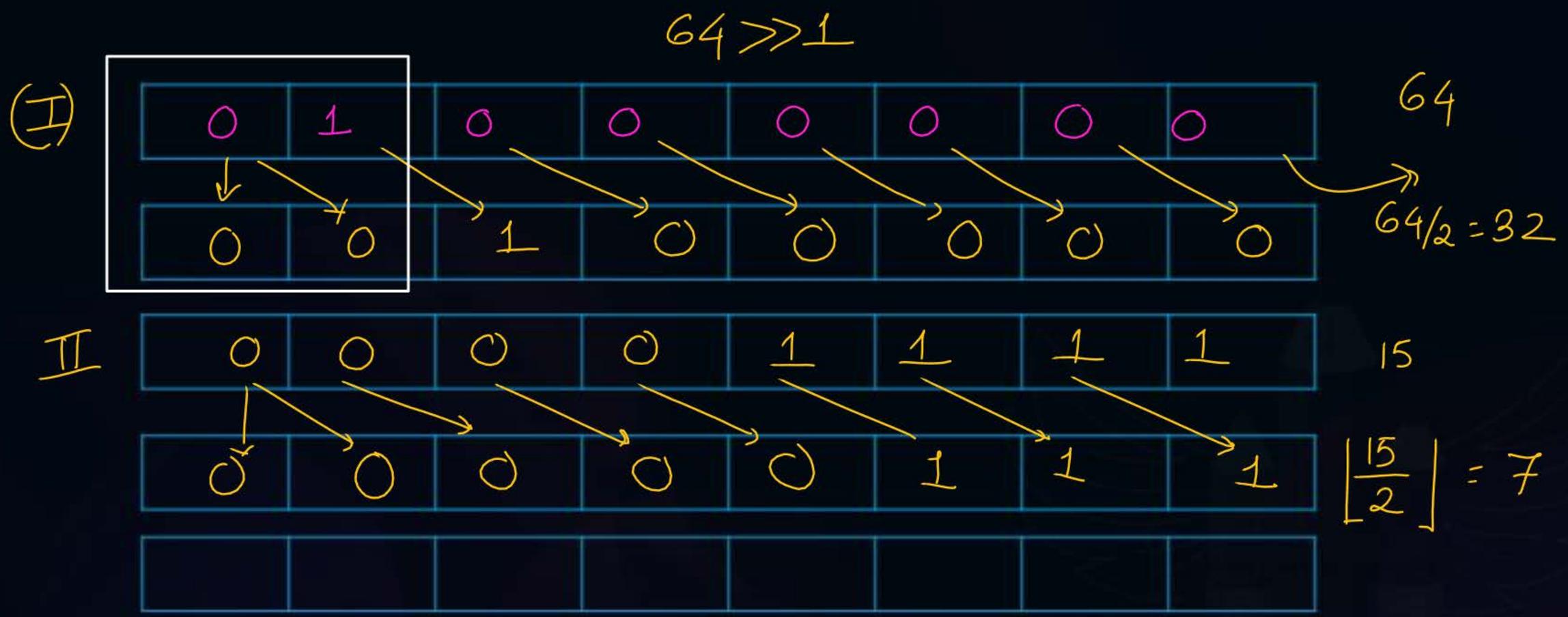
$64 >> 3 = \left\lfloor \frac{64}{8} \right\rfloor = 8$

$\lfloor -7.5 \rfloor = -8$

(A) 1
(B) 21
(C) 8 ✓
(D) -6

Slide

Bit-wise Operator



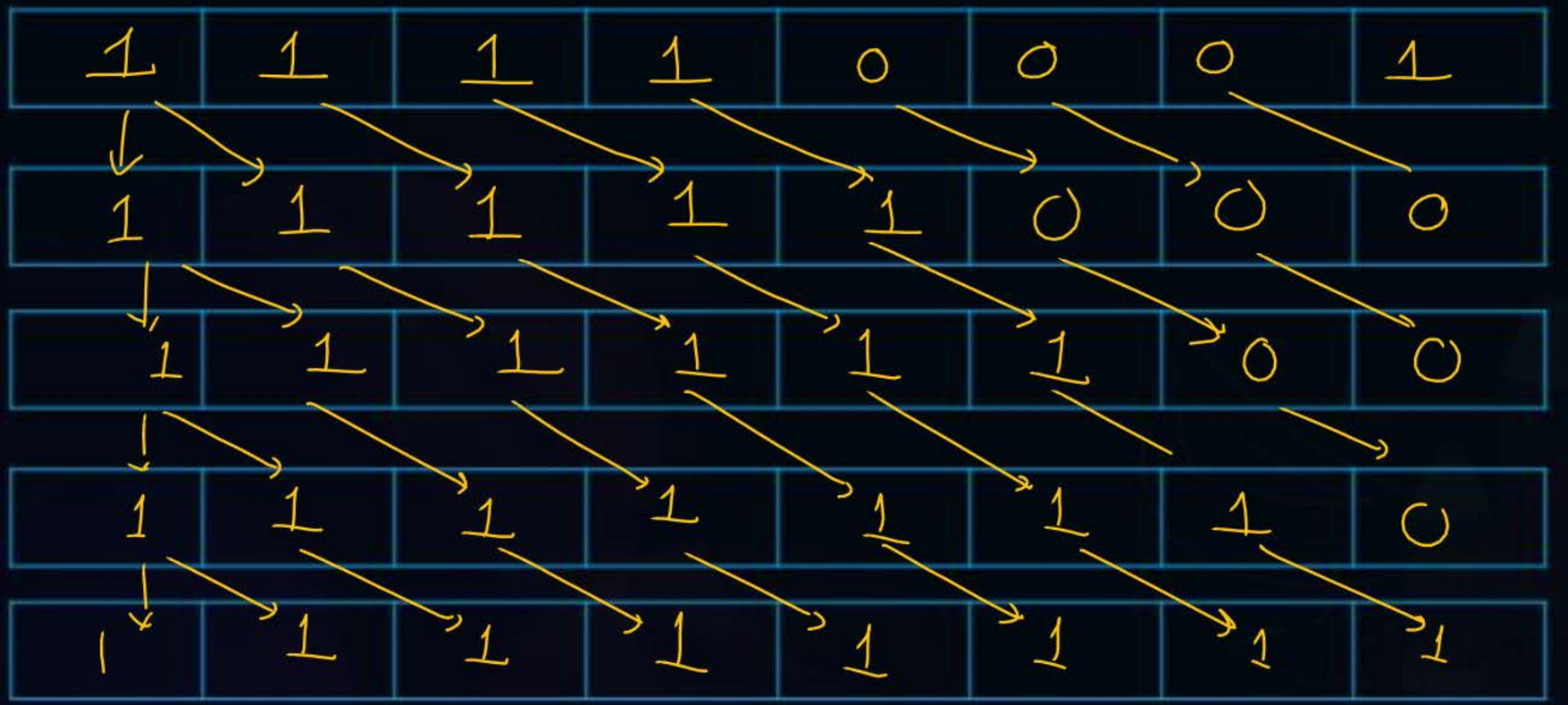
Slide

Bit-wise Operator

$\left\lfloor \frac{-15}{2} \right\rfloor = \lfloor -7.5 \rfloor$

$= -8$

-15



$\left\lfloor \frac{-15}{16} \right\rfloor$

$= -1$

Slide

Bit-wise Operator

$\sim x = -(x+1)$

```
#include <stdio.h>
    int main ()  {
      int x = 5,y=24, z;
          z = ~x + y >> 2;
       printf("%d", z);
    }
```

(A) 0
(B) 4
(C) 12
(D) -6

$\sim x + y >> 2$

$-6 + 24 >> 2$

$18 >> 2 =$

$\lfloor \frac{18}{4} \rfloor = 4$

24/6

$-6 + 24 >> 2$

$-6 + 6 = 0$

Slide

Toipc:Question

$\sim x = -(x+1)$

$\left\lfloor \frac{169}{8} \right\rfloor = 21$

```
#include <stdio.h>
     int main ()  {
        int a =5+5<<1!=169>>3;
     printf("%d", a<<3);
    }
```

a = 1

10 << 1

20 != 21 = (1)

$1 \times 2^3 = 8$

The output of the program ___

Slide

Bit-wise Operator

```
#include <stdio.h>
    int main ()  {
      int x = 5,y=24, z=10;
          x = x & 10;
          y = y | 10;
          z = z >> 2;

       printf("%d", x+y+z);
    }
```

(A) 26
(B) 27
(C) 28
(D) 29

x = x & 10; → 0

y = y | 10; → 26

z = z >> 2; → 10 >> 2 = ⌊10/4⌋ = 2

```
  00101
  1010
 ------
  0000
```

y | 10

```
  11000
  01010
 ------
  11010
 ------
```

Output of the following program is 28

Slide

Bit-wise Operator

```
#include<stdio.h>
int main( ){
   int x=5, y=256, z=1;
   x+=y >>1+z;

   printf("%d", x);
}
```

Output of the following program is

$x += y >> 1+z$

$x += y >> 2$

$x += 64$

$x = x+64 = 5+64 = 69$

$\frac{256}{4} = 64$

Slide

Print() Return value

printf("Name"), Consol: Name

int a;

a = printf("Name"),

a = 4

printf return value is No of character printed.

Slide

Print() Return value

```
#include<stdio.h>
int main(){
   printf("%d",printf("ABCD"));
   return 0; } (
```

(A) ABCD4
(B) 4
(C) ABCD 4
(D) ABCD

Inner printf

ABCD4

outer printf

printf("%d", 4)

Slide

Print() Return value

```
#include <stdio.h>
  int main(){
     printf("\n%d", printf("ABCD"));
     return 0;
   }
```

Newline

Inner printf

ABCD

auto printf

printf("\n%d", 4);

ABCD
4

Print() Return value

```
#include <stdio.h>
  int main(){
     printf("\n%d", printf("ABCD"));
     return 0;
   }
```

```
ABCD
4
```

Print() Return value

```
#include <stdio.h>
  int main(){
     int a=1000;
     printf("%d",printf("\n%d",a));
     return 0;
}
```

Output of the program is _____

(A) 1005
(B) 10005
(C) 1000
(D) 1000
    5

Inner printf

Newline \n part of inner printf

1000 ← Newline

1 Newline + 4 for (a) which is 1000

outer printf printf("%d", 5),

5

Slide

1000

↳ 5

%d\n

Ternary Operator

Conditional operator

TRUE : false

expr1 ? expr2 : expr3;

Nonzero : zero

↑

Relational : True/false

expression : Nonzero/zero

Slide

Ternary Operator

```
#include <stdio.h>
int main()
{
    int a;
    a = 10>7?10:20;
    printf("%d", a);
    return 0;
}
```

(A) 10
(B) 20
(C) 1
(D) 0

10>7 ? 10 : 20

T

a = 10

if (10>7)

a = 10

else

a = 20

Slide

Ternary Operator

```
#include<stdio.h>
int main(){
      int x=3, y=4, z=4;
      printf("%d", (z>=y>=x?100:200));
      return 0 ;
}
```

(a) 100 (b) 200 (c) 0 (d) 1

(4>=4)>=3 = 1>=3

false 0

Slide

Ternary Operator

```
#include <stdio.h>

int main(){
    int a = 10, b=14;

    a == 4 ? printf("%d",a):printf("%d", b);
    return 0;
}
```

10 == 4

(A) 10 (B) 14 (C) 4 (D) Error

Slide

Ternary Operator

```
#include <stdio.h>

int main(){
    int a = 10, b=14;

    (a = 4) ? printf("%d",a):printf("%d", b);

    return 0;
}
```

a=4

a 10 4

b 14

Nonzero?

(A) 10 (B) 14 (C) 4 (D) Error

Slide

Ternary Operator

```
#include <stdio.h>

int main(){
    int a = 10, b=14;

    a = 4 ? printf("%d",a):printf("%d", b);

    return 0;
}
```

(A) 10 (B) 14 (C) 4 (D) Error

Slide

Ternary Operator

10

```
#include <stdio.h>

int main(){
    int a = 10, b=14;

    a = 4 ? printf("%d",a):printf("%d", b);
    printf("%d",a);
    return 0;
}
```

4 Nonzero

a = printf("%d", a);

a : 2

(A) 102 (B) 104 (C) 142 (D) 144

Slide

Topic: Short Circuit Code

```
#include <stdio.h>
void main ()  {
      int x = 0, y = 4, z = 0,w=7;
      int a = printf(" ") && printf("World ") || printf("India");
}
```

A. World
B. India
C. World India
D. India World

Slide

Ternary Operator ?

```
# include <stdio.h>
int main() {
     int i= 10 ,j = 20;
     j = 10?0?6:7:5;
     printf("%d", i+j);
     return 0;

}
output
```

7

10+7
=17

Right

j = 10?0?6:7:5

? Right Associative

j = 10?7:5

j = 7

Slide

Ternary Operator ?

```
# include <stdio.h>
int main() {
     int i= 10 ,j = 0;
     j = ++j?++i? i:j:j ;
     printf("%d  %d ", i ,j);
     return 0;

}
```

output

(A) 11 11 (B) 11 1 (C) 1 1 (D) 1 11

i = 11

j = 1

j = ++j ? ++i ? i : j : j

j = ++j ? 11 ? i : j : j

j = ++j ? i : j

j = 1 1

Slide

Ternary Operator ?

```
# include <stdio.h>
int main() {
    int i, j=15;
    i= 10,++j,++j;
    printf("%d %d ", i, j);
    j = (++i, ++i, 40);
    printf("%d %d ", i, j);
    return 0;

}
```

output 10, 17 . 12 40

int i = 10,20,30, ← Error

Assignment ⇒ 10, 17

14 15

i = 10, ++j , ++j

i = 10 16 17

12, 40

j will be assigne 40

but ++i, ++i will be evaluated

11 12

Slide

Ternary Operator ?

```
# include <stdio.h>
int main() {
     int i= 10 ,j = 20;
     j = i,j?(i,j)? i:j:j ;
     printf("%d  %d ", i ,j);
     return 0;

}
```

Output

(A) 10 10 (B) 11 1 (C) 1 1 (D) 1 11

j?(i,j)?i:j:j

j?j?i:j:j

j?i:j

j=i,i

Slide

2 mins Summary

Topic >> shift right

Topic << shift left

Topic printf return value

Topic Ternary operator

Topic Comma operator

Slide

Thank You