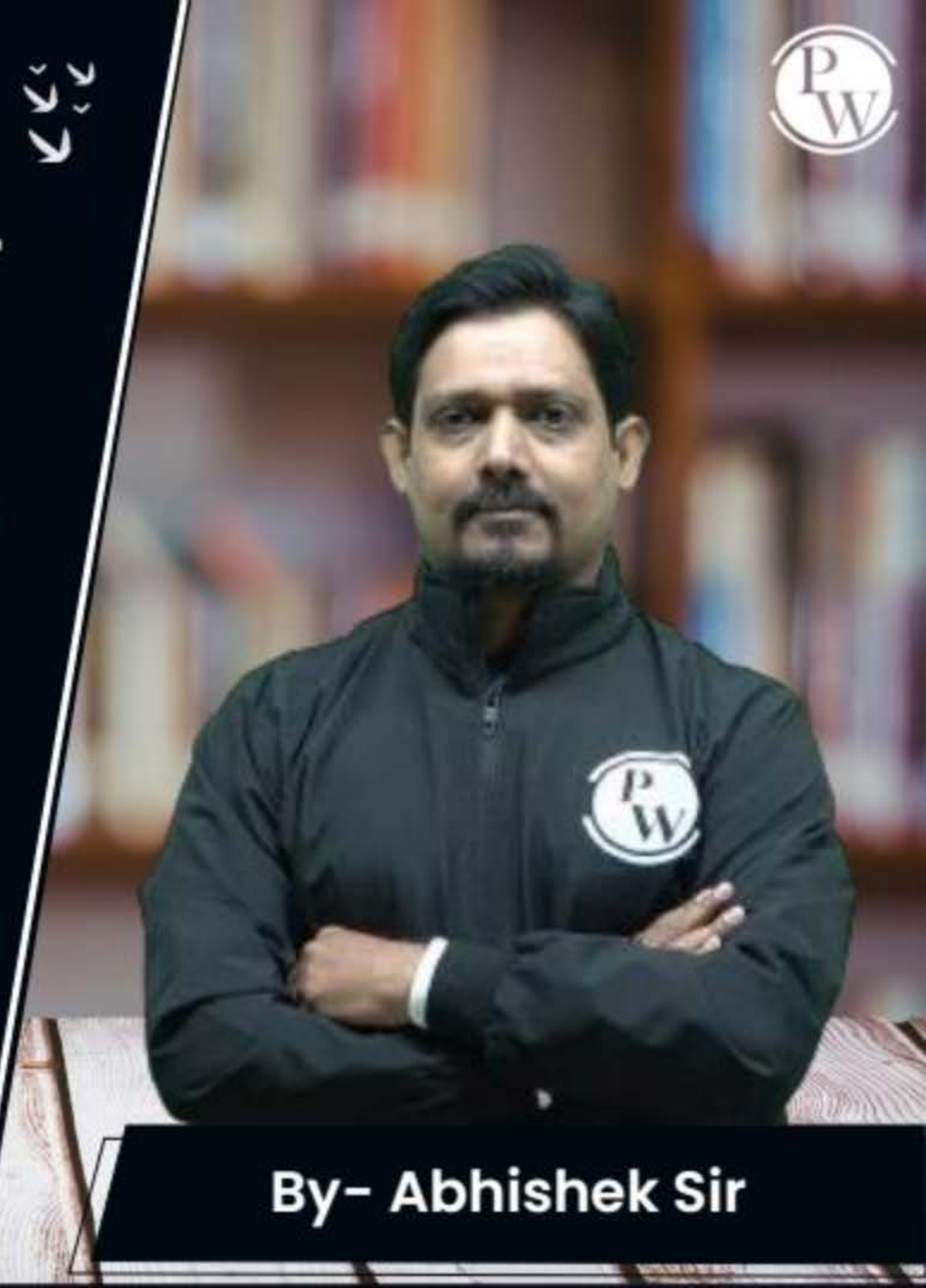
Computer Science & IT

C Programming

Array & Pointers

Lecture No. 02



By- Abhishek Sir

Recap of Previous Lecture

Topic pointer

Topic call by reference

Topic

Topic

Topic

Slide

Topics to be Covered

Topic 1-D array

Topic Data type of 1-D array

Topic

Topic

Topic

Slide

Question

Consider the following C program?

```
#include <stdio.h>
void func(int *x,int *y,int *z){
   *y=*y+4;
   *z=*x+*y+*z;
}
int main()
{
   int x = 10, y=3;
   func(&y, &x, &x);
   printf("%d %d",x,y );
   return 0;
}
```

The value printed by above program is

Call by referece

(A) 10, 3

(B) 31, 3 ✓ [B]

(C) 27, 7

(D) 27, 3

Slide

Question

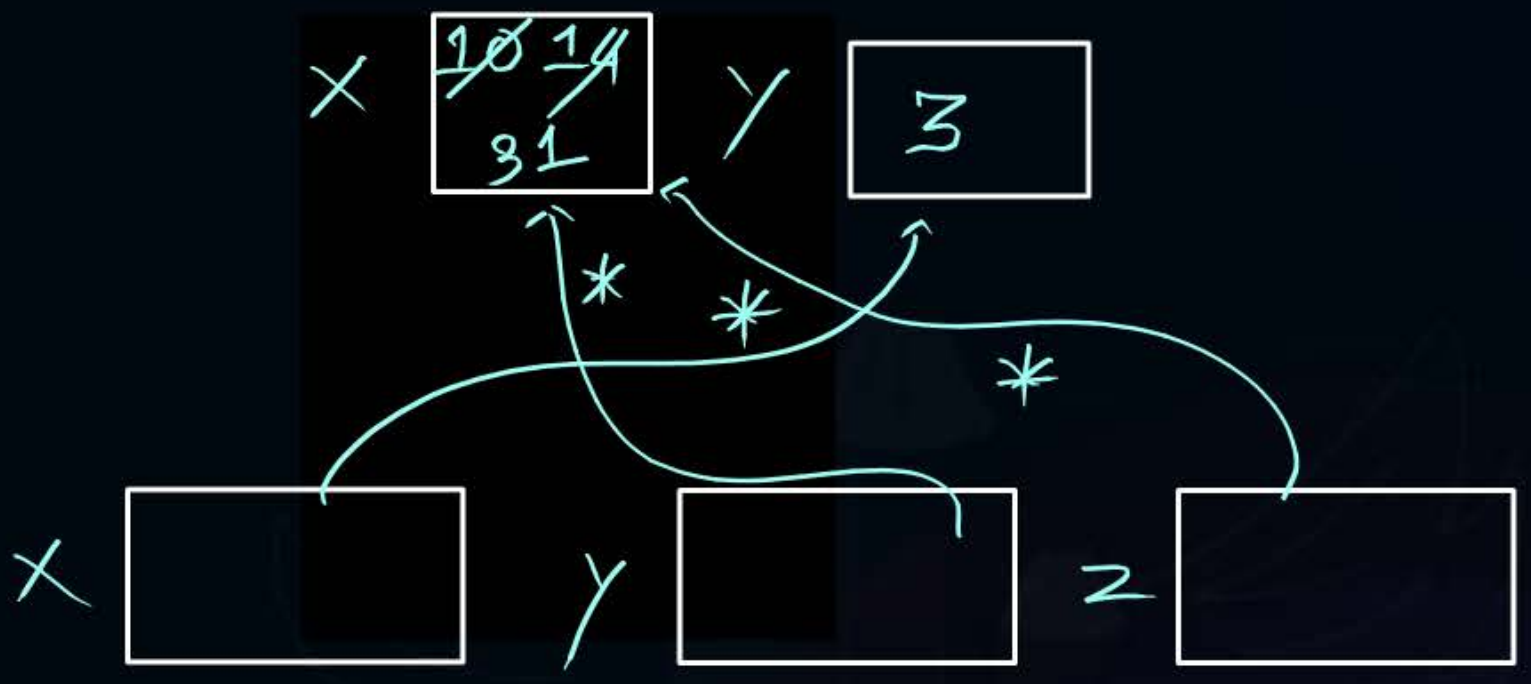
Consider the following C program?

```
#include <stdio.h>
void func(int *x,int *y,int *z){
   *y=*y+4; ✓
   *z=*x+*y+*z; ✓
}
int main()
{
   int x = 10, y=3;
   func(&y, &x, &x);
   printf("%d %d",x,y );
   return 0;
}
```

31 3

The value printed by above program is

Call by reference



$*z = *x + *y + *z$

$3 + 14 + 14 = 31$

Slide

```
void swap1(int a, int b){
  int temp;
  temp= b;
  b= a;
  a= temp;
}
```

Localswap

```
void swap2(int* a,int* b){
        int *temp;
        temp= b;
        b= a;
        a= temp;
}
```

Address swap

```
void swap3(int* a, int* b)
        int temp;
        temp= *b;
        *b= *a;
        *a= temp;
}
```

Actual swap

```c
void swap1(int a, int b){
  int temp;
  temp= b;
  b= a;
  a= temp;
}
#include <stdio.h>
int main(){
  int x = 10, y=20;
  swap1(x,y);
  printf("%d %d", x, y);
  return 0;
}
```

10, 20

```
void swap2(int* a,int* b){
      int *temp; ✓
      temp= b;
      b= a;
      a= temp;
}
#include <stdio.h>
int main(){
  int x = 10, y=20;
  swap2(&x,&y);
  printf("%d %d", x, y);
  return 0;
}
```

No

dereference

a = 100   b : 200 ✓

Addres swap

200   b 100

temp = 200

b = 100

a = 200

```
void swap3(int* a, int* b){
        int temp; ✓
        temp= *b;
        *b= *a;
        *a= temp; ✓
}
#include <stdio.h>
int main(){
   int x = 10, y=20;
   swap3(&x,&y);
   printf("%d %d", x, y);
   return 0;        20 10
}
```

a 100   b = 200

temp = *100

temp = 20

*b = *a

200 $\boxed{10}$

GATE 2016

Consider the following C program.

```
#include<stdio.h>

void mystery(int* ptra, int* ptrb){

            int *temp;

            temp= ptrb;

            ptrb= ptra;

            ptra= temp;

      }
```

Address
swap

```
int main(){
      int a=2016, b=0, c=4, d=42;
      mystery(&a, &b);
      if (a<c)
            mystery(&c, &a);
      mystery(&a, &d);
      printf("%d\n", a);
      }
```

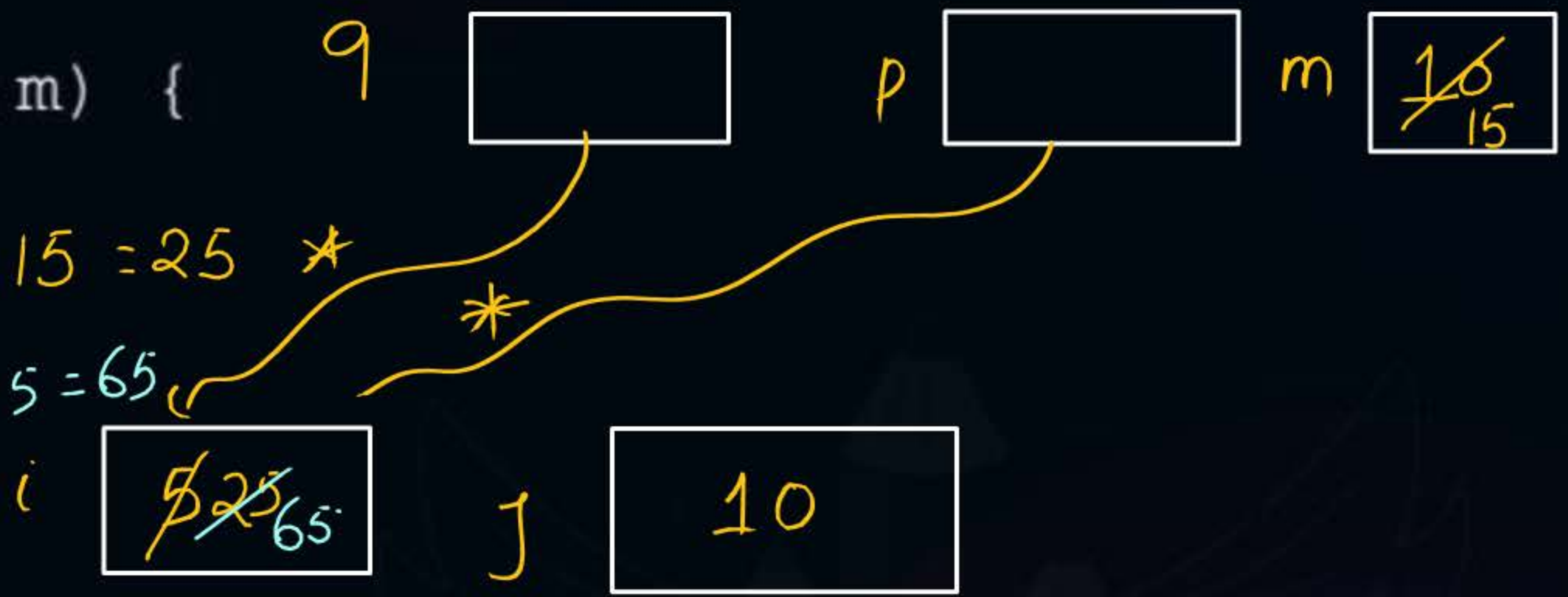
The output of the program is ___2016___ .

Slide

Question

The value printed by the following program is ________.

```
#include <stdio.h>
int bar(int * q,int *p, int m) {
     m = m + 5;
     *p = *p+*q + m; ✓  5+5+15 =25
     *q = *q + m+*p;  25+15+25=65
     return m+*q;
}              15+65=80
void main () {
 80  int i=5, j=10;
 j=  j = bar (&i,&i, j);
     printf ("%d", i+j);
}                 65+80=145
```



Slide

Question

```
#include<stdio.h>
void f(int *x, int *y) {
    int * temp;
    temp = x;
    x = y;
    y= temp;
    *x=*x+2;
    *y=*y<<2;}
int main() {
    int a=10, b=2;
    f(&a, &b);
    f(&a, &b);
    printf("%d [redacted]\n", a+b);
    return 0;
} Output of the program is __________
```

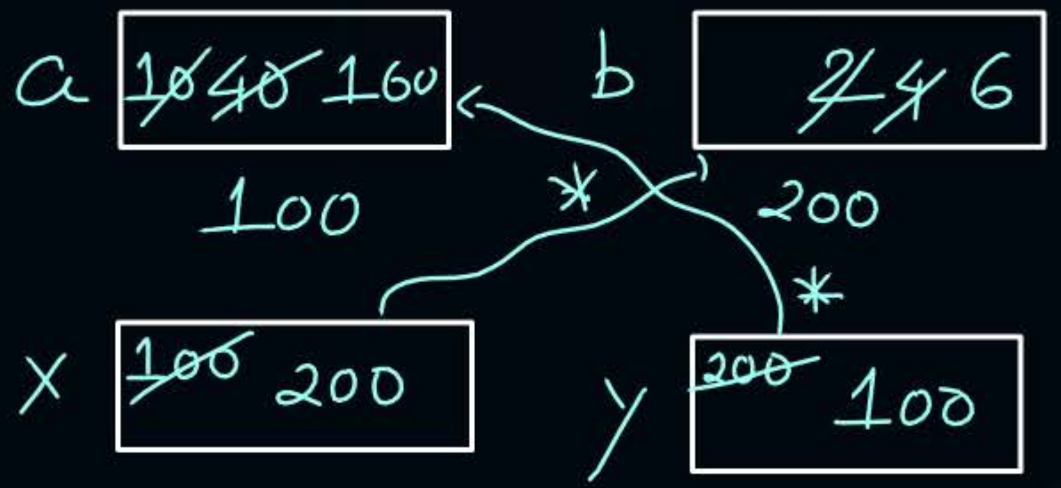
Address swap

Leftshift

$160+6=166$

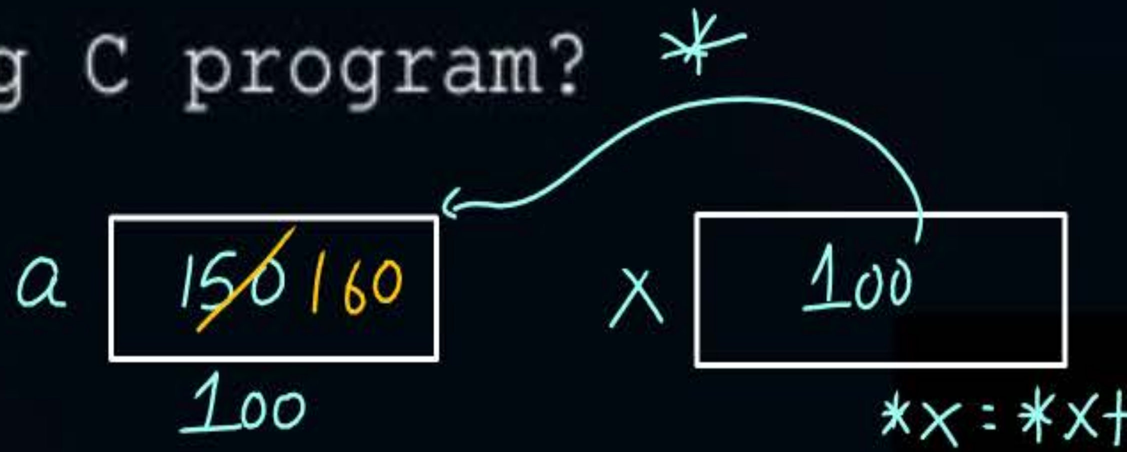
$10\times 2^2=40$

$40\times 2^2=160$



Consider the following C program?

```
#include <stdio.h>
int a = 100; ✓
void fun(int *x ){
      for (int i=1;i<=5;i++){
        *x = *x+10;
        a = a+*x;
      }
}
int main(){
      a = a+50;
      fun(&a);
      printf("%d",a );
      return 0;
}The value printed by above program is
```



	*x = *x+10	a = a+*x
		160+160 = 320
i=1	160	
i=2	330	660
i=3	670	1340
i=4	1350	2700
i=5	2710	$\boxed{5420}$

Pointer Arithmetic

pointer/Address

1 Addition and Subtraction with constant

2 ++, -- pre, post Add/Subtract by const 1

we can Also Subtract two pointer of same data type

Slide

Pointer Arithmetic

- int *ptr, ptr+c = ptr+c* Size of data type

  ptr+ c×4,

- char *ptr; ptr+c = ptr+c×1;

- double *ptr, ptr+c = ptr+c*8

Slide

Array

Array is Collection of Similar data type

Array stores element in Sequential manner

Notvalid int a[ ]; No - Space required, (Size)

✓ int a[4].

int a[ ] = { 1, 2, 3, 4, 5 }; Size = 20B

Slide

Array

int = 4B

int $\underline{a[4]}$ = {1,2,3,4};

1	2	3	4
100	104	108	112

Access array element : [ ] Array Subscript operator

first element : a[0]

Second element : a[1]

Last element : a[4-1] = a[3]

Slide

Array

1-D array

a[i]

a :- Name of array

Address of first element

Address of integer

} data type

Address of integer

*a = a[0] : integer } integer

Slide

Array

$$a[i] = *(a+i) = *(i+a) = i[a]$$

int 4[a]; Not a valid declaration

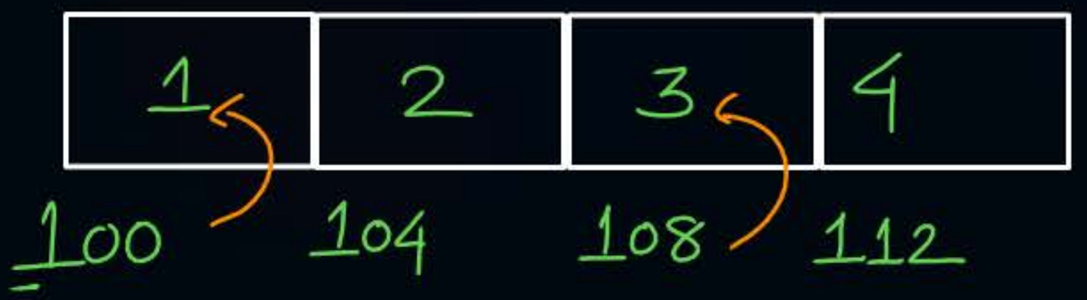
printf("%d", 3[a]);

Slide

Array

int = 4B

$\text{int } \underline{a[4]} = \{1,2,3,4\};$



1	2	3	4
100	104	108	112

$$*(a+0) = *(100+0) = *100 = 1$$

$$a[2] = *(a+2) = *(100+2) = *(100+2\times4)$$

$$= *(108) = \boxed{3}$$

Slide

Question

```
include<stdio.h>
int main()
{
int a[] = {1, 23, 4, 5};
int *b= a+3;

printf("%d " b[-2]);
}
```

100	104	108	112
1	23	4	5

int *b = a+3 = 100+3 = 100+3×4 = 112

b $\boxed{112}$

(A) 1

[✓ (B) 23]

(C) 4

(D) 5

b[-2] = *(b-2)

*(112-2)

*(112-2×4)

*(104)

Slide

Arithmetic Operation with Pointer

#Q

```
main (){
      int a[] ={10,20,30,40,50};
      int i,*b;
      b = &a[4]-4;
      for(i=0;i<=4;i++){
           printf("%d",*b);
           b++;
      }
return 0;
}
Output__________
```

100 104 108 112 116

$\&a[4] = 116$

$116-4 = 116-4\times4$

$b = \boxed{100}$

$\boxed{104}$

$\boxed{1020304050}$

Slide

Question

b - base Address

Out put of the program

```
#include<stdio.h>
     int main(){
     int i , b[] = {2,| 3,| 4,| 5,| 6};
     b++;
     printf ("%d\t" ,*b) ;
     }
```

100

b ↔ $\boxed{100}$

Constant Association

b++

100++

L - value required

(A) 2

(B) 3

(C) Address Increment

[✓ (D) Error]

Slide

Question

3

++ *p right to Left

#Q What is the output of the following program ?

```
#include<stdio.h>
int main(){
int i , b [] = {2, 3, 4, 5, 6}, *p ;
p = b ;
++*p ;
printf ("%d\t" ,*p) ;
p += 2 ;
printf ("%d" , *p);
}
```

100 104 108 112 116

3

p 100

*

*

(A) 1 3

(B) ~~2 3~~ 3, 3

(C) 2 4

(D) 3 4 ✓

100+2 = 100+2×4

= 108

Slide

Question

Consider the following C program

```
                    100  104  108  112  116
#include <stdio.h>
int main(){
   int a[] = {2, 4, 6, 8, 10};
   int i, sum = 0, *b = a + 4;
   for (i = 0; i < 5; i++ )
      sum = sum + (*b - i) - *(b - i);
   printf("%d\n",sum);
   return 0;
}
```

The output of the above C-program is ___10___.

Sum (*b - i) - *(b - i)

i=0 0 + (10-0) - 10

i=1 0 + (10-1) - *(116-1) = *(112) = 8

i=2 1 + (10-2) - *(116-2) = *(108) = 6

i=3 3 + $\overset{7}{(10-3)}$ - 4 = 6

i=4 6 + $\overset{6}{(10-4)}$ - 2 2 = 10

$*b = \boxed{116}$

Slide

2 mins Summary

Topic Swap function

Topic 1-D array

Topic pointer Arithmetic

Topic

Topic

Law of Average

Slide

function

THANK - YOU