

LECTURE - 33

POINTER ARITHMETIC AND
ARRAYS (PART 02)

PROGRAMMING IN 'C'

Pointer Arithmetic

```
int main ( ) {
```

```
    int x = 15;
```

```
    int *ptr = &x;
```

```
    printf ("%d", x+10);
```

```
    printf ("%d", ptr+1)
```

```
    return 0;
```

```
}
```

$$x + 10 = 15 + 10 = 25 \checkmark$$

$$\text{pointer Value} + \text{num} * \text{Size (pointerType)}$$

$$\begin{aligned} &1011 + 1 * 4 \\ &= 1011 + 4 = 1015 \end{aligned}$$

1011

15

x ← int

1020

1011

ptr

integer pointer
↳ 4 Bytes

$$\text{result} = \text{pointer Value} + \text{number} \times \text{sizeof (Type of pointer)}$$

↓
Address

↓
Value at pointer

Type of reference variable

```
int main(){
```

```
    int a = 15;
```

```
    printf("%d \n", a+2);
```

```
    printf("%d \n", &a+2);
```

$$a+2 \Rightarrow 15+2=17$$

$$\begin{aligned} \&a+2 &\Rightarrow 100+2 \times 4 \\ &\Rightarrow 100+8=108 \end{aligned}$$

```
    double b = 75.7575;
```

```
    double *ptr = &b;
```

```
    printf("%d \n", ptr+3);
```

$$ptr+3$$

Size of (double)
68

$$110+3 \times 8$$

$$= 110+24$$

$$= 134$$

```
    return 0;
```

```
}
```

100

110

200

15
75.7575
110

int
a

double
b

ptr

```
value of x = 7569
Addr of x = 6422052
ptr+10 = 6422092
PS>
```

$$x = 7569$$

$$ptr = \&x = 6422052$$

$$ptr + 10 \rightarrow$$

$$ptr + 10 \times \text{Size of (value at pointer)}$$

$$\text{Size of (int)} \\ \rightarrow 4$$

↓

$$6422052 + 10 \times 4$$

$$6422052 + 40$$

$$\Rightarrow 6422092$$

```
nters.c  pointers2.c  pointer3.c  pointer4.c  pointerarithmetic1.c  pointerarithmetic2.c ×
pointerarithmetic2.c > main()
1  #include<stdio.h>
2  #include<conio.h>
3  #include<math.h>
4  int main(){
5      double x = 75.59;
6      printf("value of x = %f \n", x + 10);
7      double *ptr = &x;
8      printf("Addr of x = %d\n", ptr);
9      printf("ptr+10 = %d\n", ptr+10);
10     return 0;
11 }
```

value of x = 85.590000
Addr of x = 6422048
ptr+10 = 6422128
PS>

$x = 75.59$
 $ptr = 6422048$
 $printf(ptr + 10)$

$6422048 + 10 \times 8 \xleftarrow{\text{double}}$
 $\Rightarrow 6422048 \times 80$
 $\Rightarrow 6422128$


```
int main() {
```

```
    int x = 0;
```

```
    int *ptr = &x;
```

```
    printf("%d", ptr++);
```

```
    return 0;
```

```
}
```

$\&x = 100$

$ptr = ptr + 1$

↓

$100 + 1 \times \text{sizeof}(\text{int})$

$100 + 1 \times 4$

104

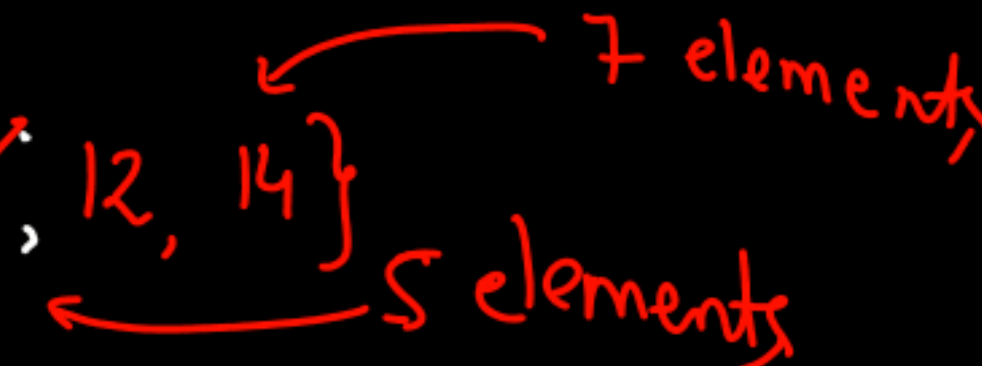
104

int arr []; \longleftrightarrow Error


\Rightarrow The number of elements are must during declaration of an empty array.

defining the array:

- ① datatype Array name [] = { Element 1, Element 2 ... Element n } ;
- ② datatype Array name [no. of elements] = { Element 1, Element 2 ... Element n } ;

Eg int Array 1 [] = { 2, 4, 6, 8, 10, ~~12~~, 14 }


int Array 2 [5] = { 1, 3, 5, 7, 9, 11, 13, 15 } ;

 take first 5 elements
{ 1, 3, 5, 7, 9 }

Structure of an Array

default Value = zero

```
int StudentId[7];
```

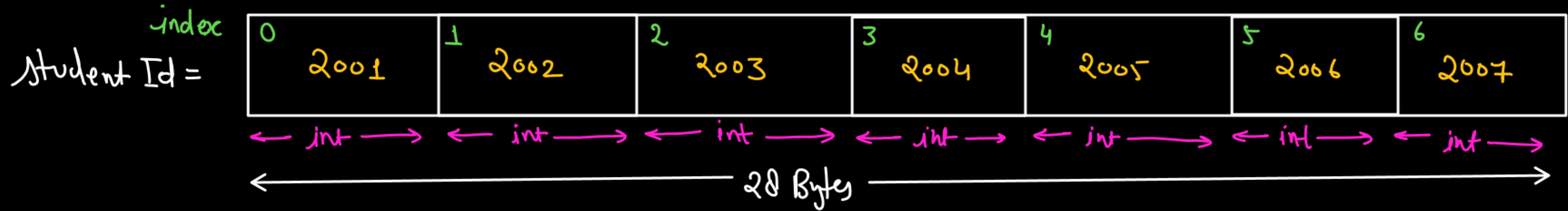
```
StudentId[7] = { 2001, 2002, 2003, 2004, 2005, 2006, 2007 } ;
```

element_s (int) 4 Bytes

Temporary

index \Rightarrow are assigned to the values of an array. (index starts from 0)

\hookrightarrow [first element = 0 last element = n-1]



Size of (StudentId)

\hookrightarrow no. of elements * size of (element)

$7 * 4$

\Rightarrow 28 Bytes

Accessing the Value of an array:

Syntax Array Name [index]

Ex printf ("%d", StudentId [2]);

↘ 2003

StudentId on 2nd index पर
जि value है

printf ("%d", StudentId [6]);

6th index
↓
2007

printf ("%d", StudentId [10]);

index = 0 to 6

Index out of Bound Error