

C Language

Pointers



Saurabh Shukla (MySirG)

Agenda

- ① Introduction to memory address
- ② Referencing and Dereferencing operators
- ③ What is pointer?

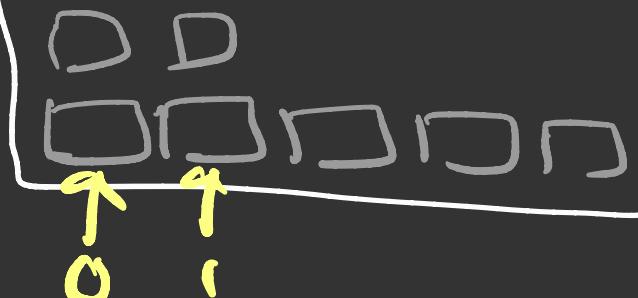
Introduction to Memory Address

int x;



100 ← address
↑
reference
↑
position number
of byte
(0 based counting)

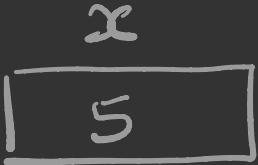
← x →
□ □ □ □
100



- Address number is always a whole number
- we cannot decide an address number of a variable
- we cannot change address of a variable

Referencing and Dereferencing operators

```
int x=5;  
printf("%d", x); 5  
printf("%d", &x); 100  
printf("%d", *(&x)); 5
```



$*\&x \approx x$

$\&$

- 'address of' operator
- referencing operator
- Unary Operator

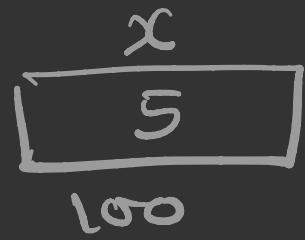
$\& \leftarrow$ variable

$*$

- indirection operator
- dereferencing operator
- Unary operator

$* \leftarrow$ address

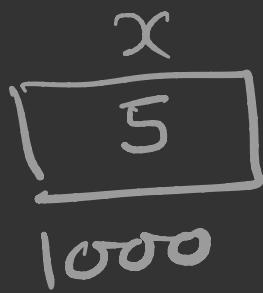
int x = 5;



&x = 7;

```
int x=5;
```

```
&x = 7;
```

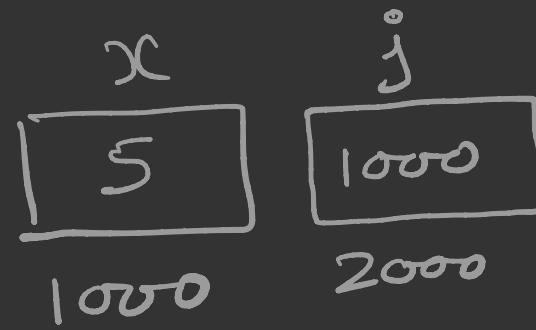


Why error?

$\&x$ is not a variable, it is just a way to represent address of variable x. Address number is a constant value.

We cannot have constant in the left hand side of assignment (=) operator.

```
int x = 5;  
int *j;
```



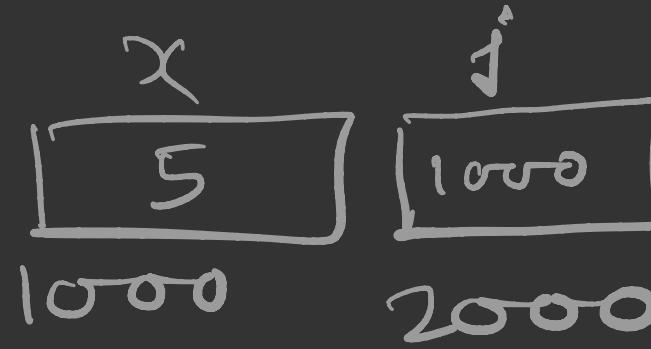
```
j = &x;
```

```
5 1000 1000  
printf("%d %d %d", x, &x, j);
```

```
5 2000 5  
printf("%d %d %d", *x, &j, *j);
```



```
int x = 5;  
int *j;  
j = &x;
```



j is a pointer variable
