C Language

Pointers

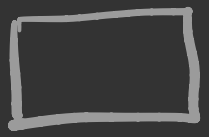
Saurabh Shukla (MySirG)

Agenda

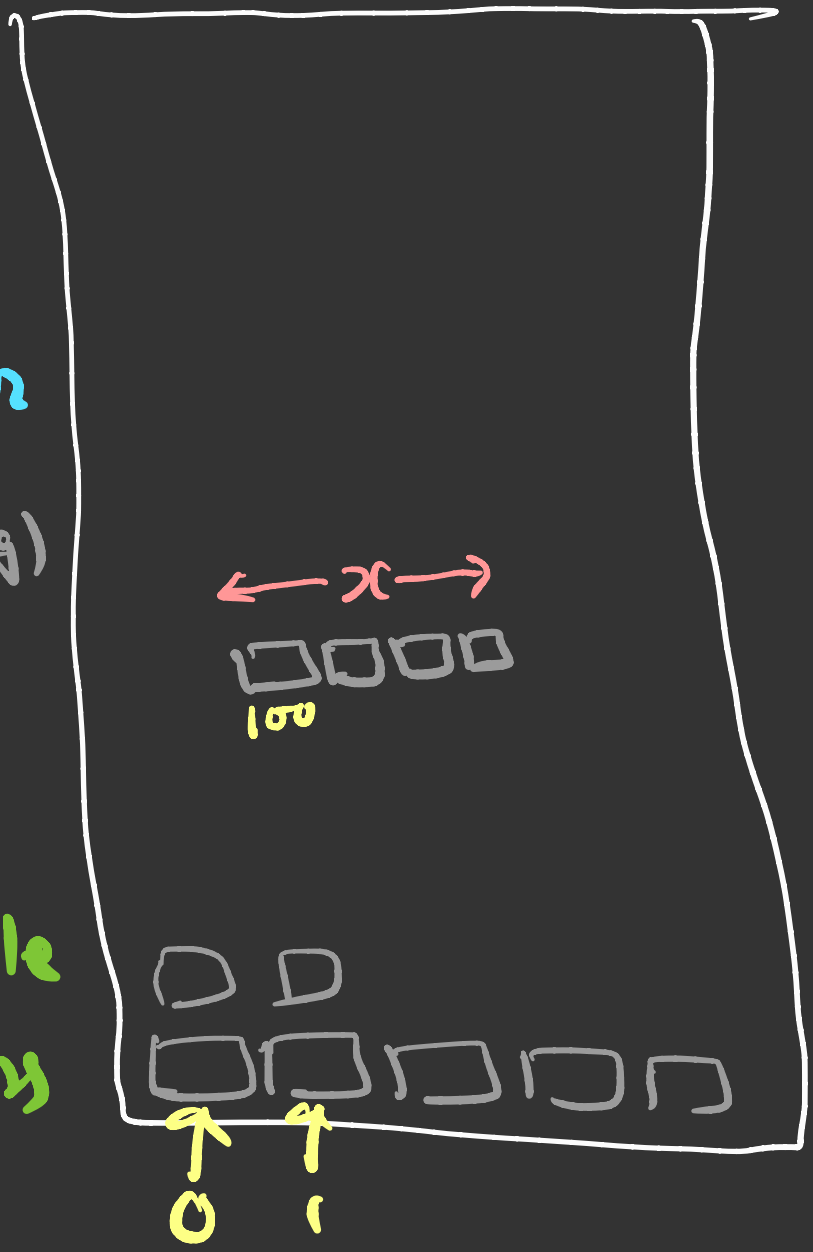
1. Introduction to memory address
2. Referencing and Dereferencing operators
3. What is pointer?

Introduction to Memory Address

```
int x;
```



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



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

Referencing and Dereferencing operators

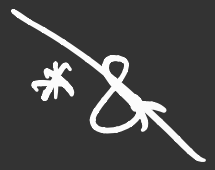
```
int x = 5;
printf("%d", x);
printf("%d", &x);
printf("%d", *&x);
```

5
100
5

x ← variable name is x

5 ← value in x is 5

100 ← address of x is 100
reference of x is 100



$*\&x \leftrightarrows x$

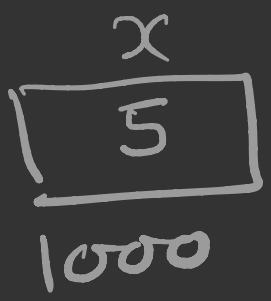
&

- 'Address of' operator
- referencing operator
- Unary operator
- & ___ ← variable

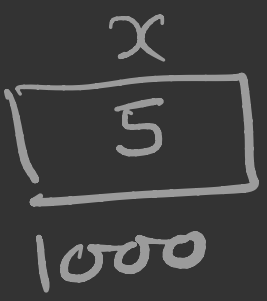
*

- Indirection Operator
- Dereferencing operator
- Unary operator
- * ___ ← address

```
int x = 5;
&x = 7;
1000 = 7
```



```
int x = 5;
&x = 7;
```



Why error?

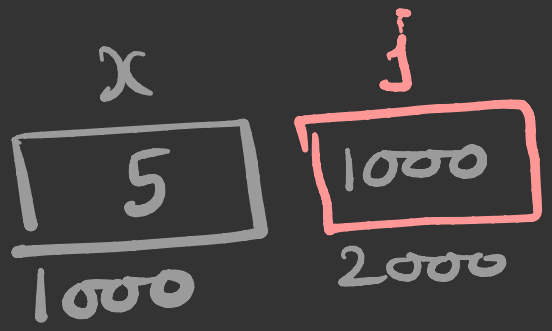
&x is not a variable, it is just a way to represent address of variable x. Address number is a constant value.

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

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

j = &x;

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



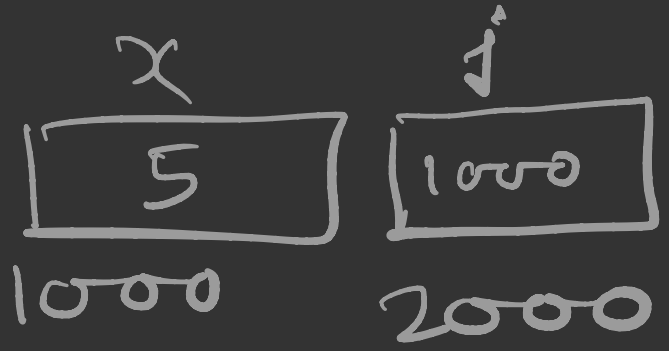
j is a pointer variable

1000 1000 5

5 5 2000

$*j = x$

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



j is a pointer variable