

CS & IT ENGINEERING

Programming in C

Miscellaneous
Chapter 07
Lec- 01

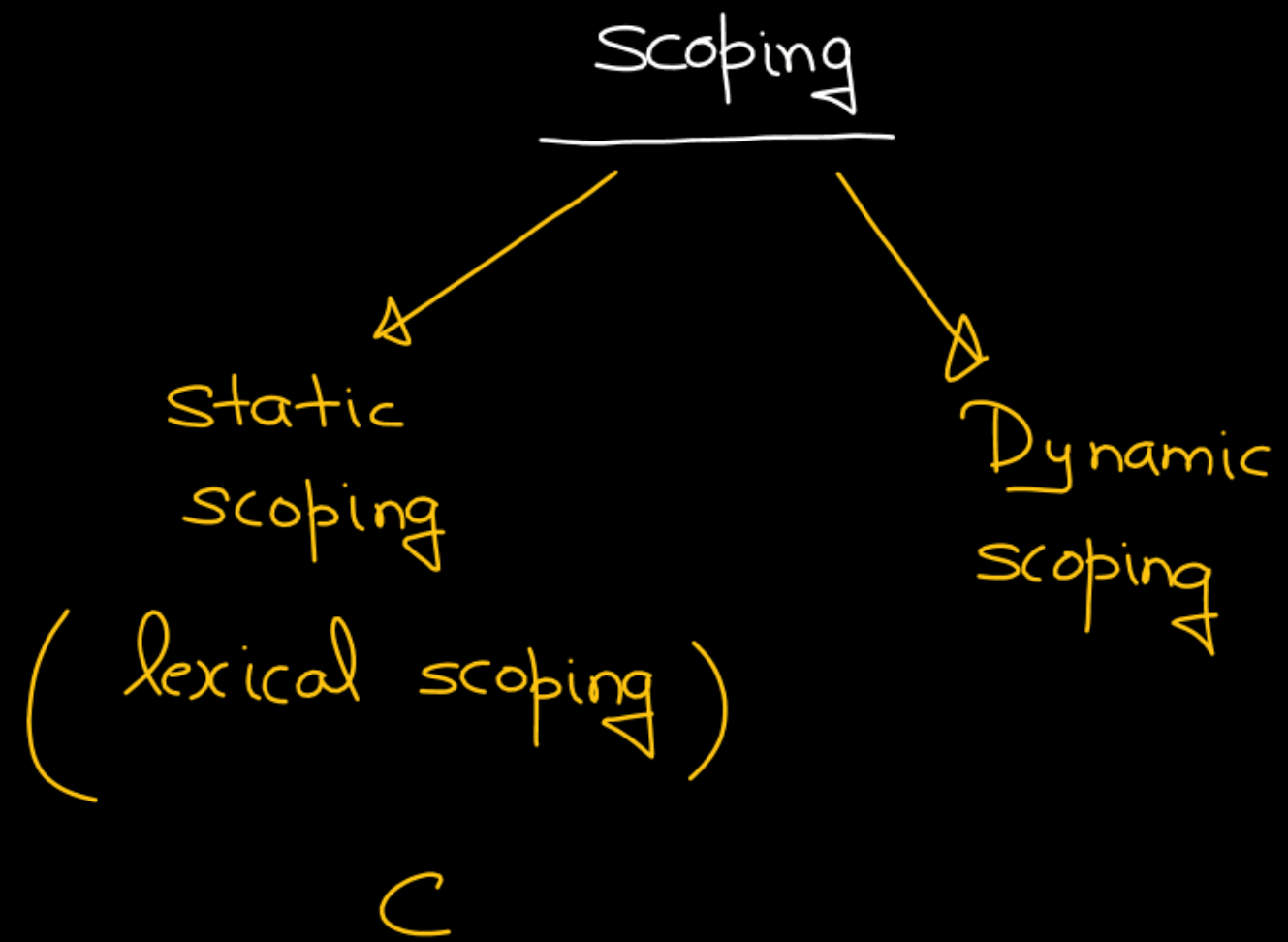


By- Pankaj Sharma sir



TOPICS TO BE
COVERED

Miscellaneous Topics



Consider the program in a hypothetical
lang. that allow global var. & a choice of static
& dynamic scoping.

(4)

LaTeX ✓

Lisp ✓

```
int i;  
Program main() { i = 10;  
                  Call f();  
                  }
```

```
Procedure f() { int i = 20;  
                Call g();  
                }
```

```
Procedure g() { Print i; }
```

let x : value printed under static
scoping
 y : value printed under
dynamic scoping

main()



i [10]

i_f [20]

Dynamic
scoping

20

y = 20

```
int i;
```

```
Program main() { i = 10;  
                  Call f();  
}
```

```
Procedure f() { int i = 20;  
                Call g();  
}
```

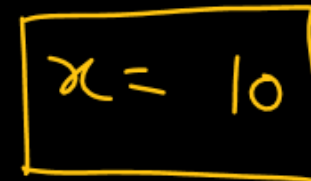
```
Procedure g() { Print i; }
```

let x : value printed under static
scoping
y : value printed under
dynamic scoping

main



Static scoping



```
int i;  
Program main() {  
    i = 10;  
    Call f();  
}
```

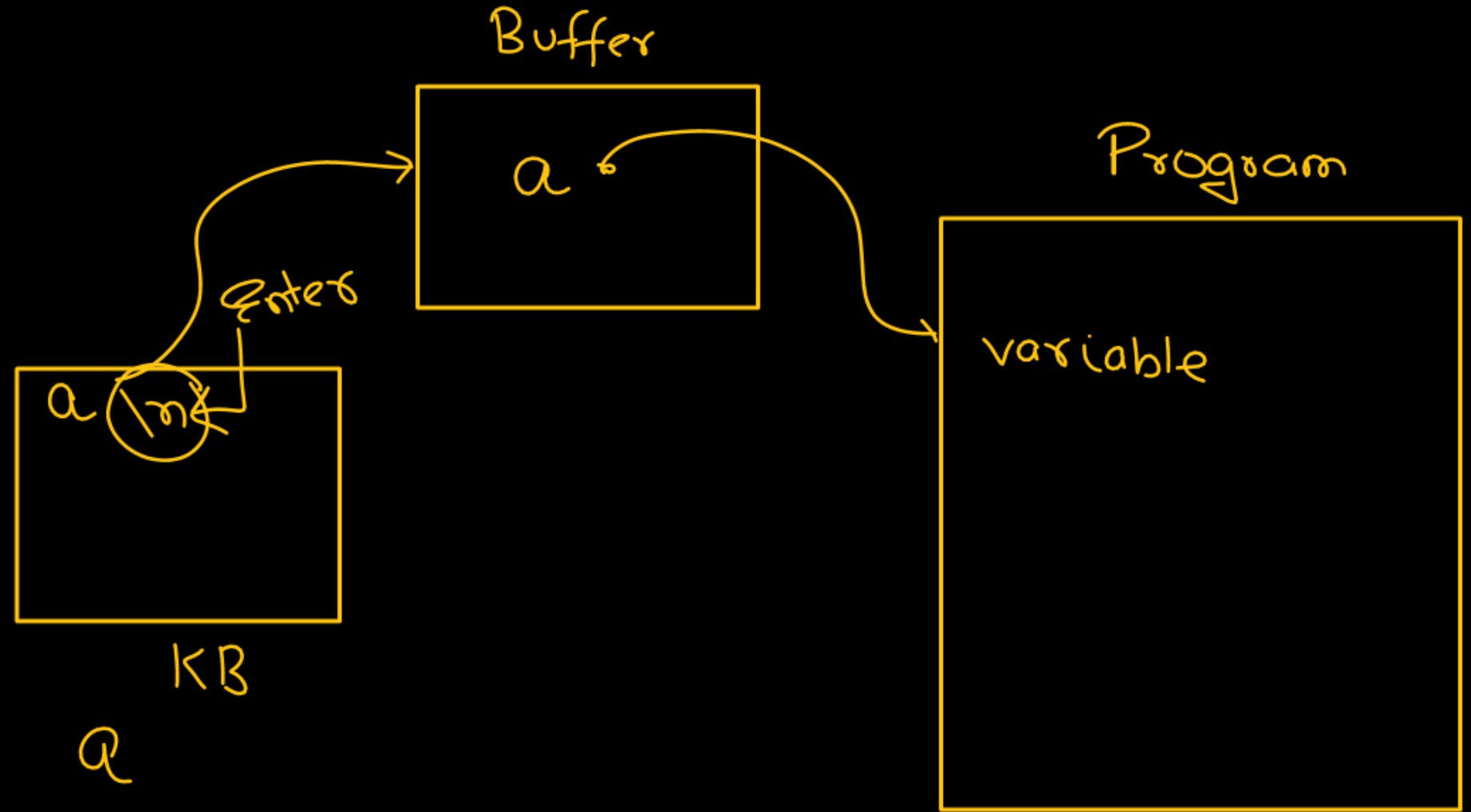
```
Proceduse f() {  
    int i = 20;  
    call g();  
}
```

```
Proceduse g() {  
    Print i;  
}
```

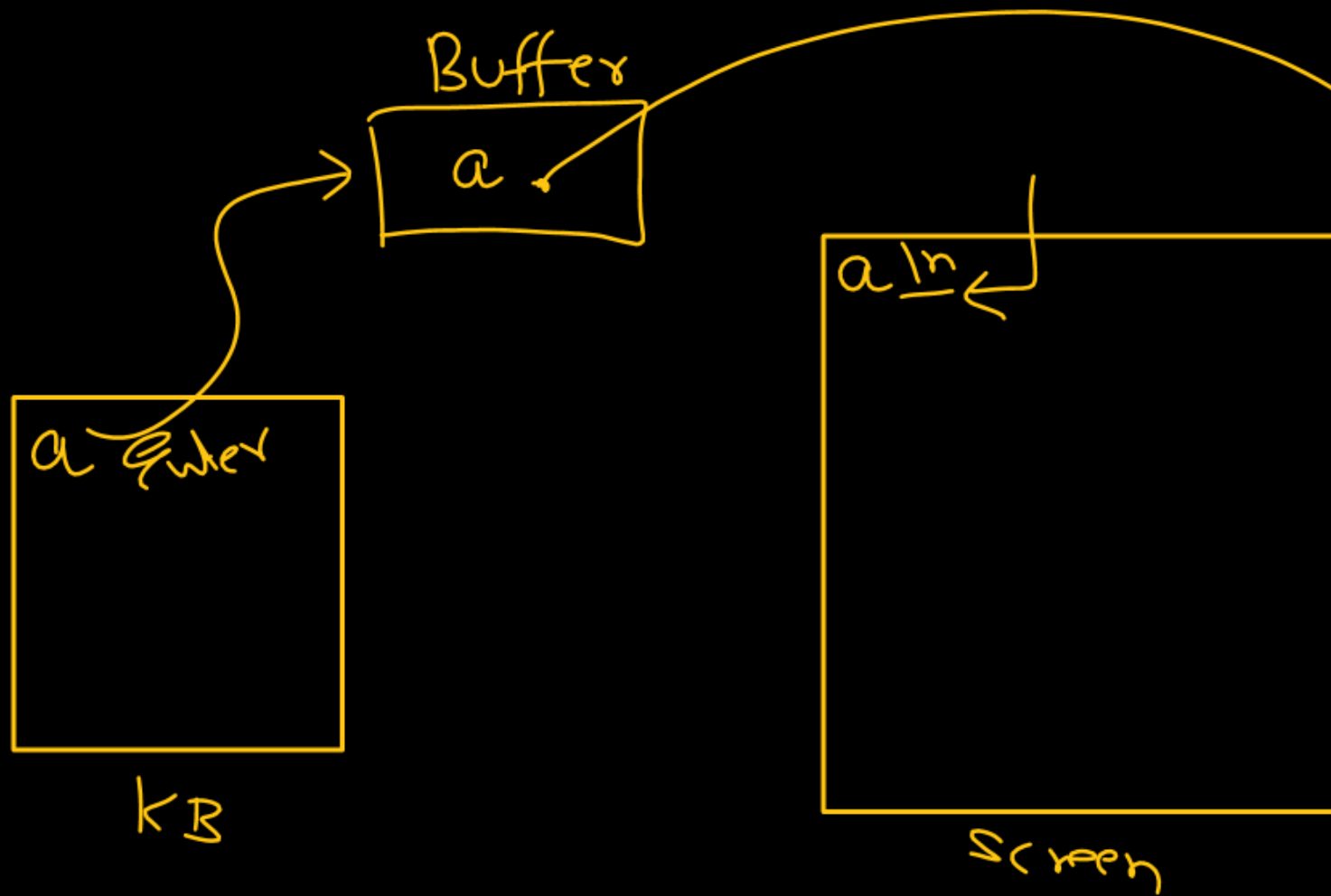
↑
global

- ① getchar
- ② getch
- ③ getche

- ① Buffered or not?
- ② Echoed or not?



① getchar → Buffered
→ echoed



```
void main(){
```

```
    int ch;
```

```
    ch = getchar();
```

```
    printf("%c", ch);
```

```
}
```


②

getch

Unbuffered

unechoed

ch

a

TurboC

a

KB



```
void main(){  
    char ch;
```

```
    ch = getch();  
    printf("%c", ch);
```

```
}
```

③

getche

Unbuffered

Echoed ✓

ch

a

a

aa

```
void main(){
```

```
    char ch;
```

```
    ch = getche();
```

```
    printf("%c", ch);
```

Comma Operator

① Works as separator.

```
int a = 5, b = 4, c = 6;
```

OR

```
int a = 5;  
int b = 4;  
int c = 6;
```

② Works as an operator.

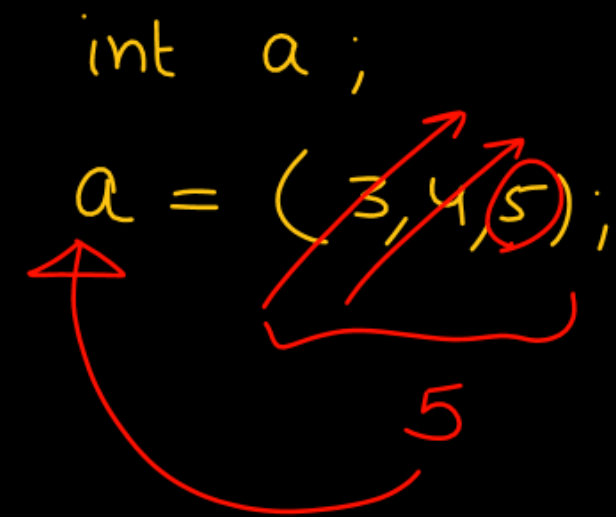
```
int a ;
```

```
a = (3, 4, 5) ;
```

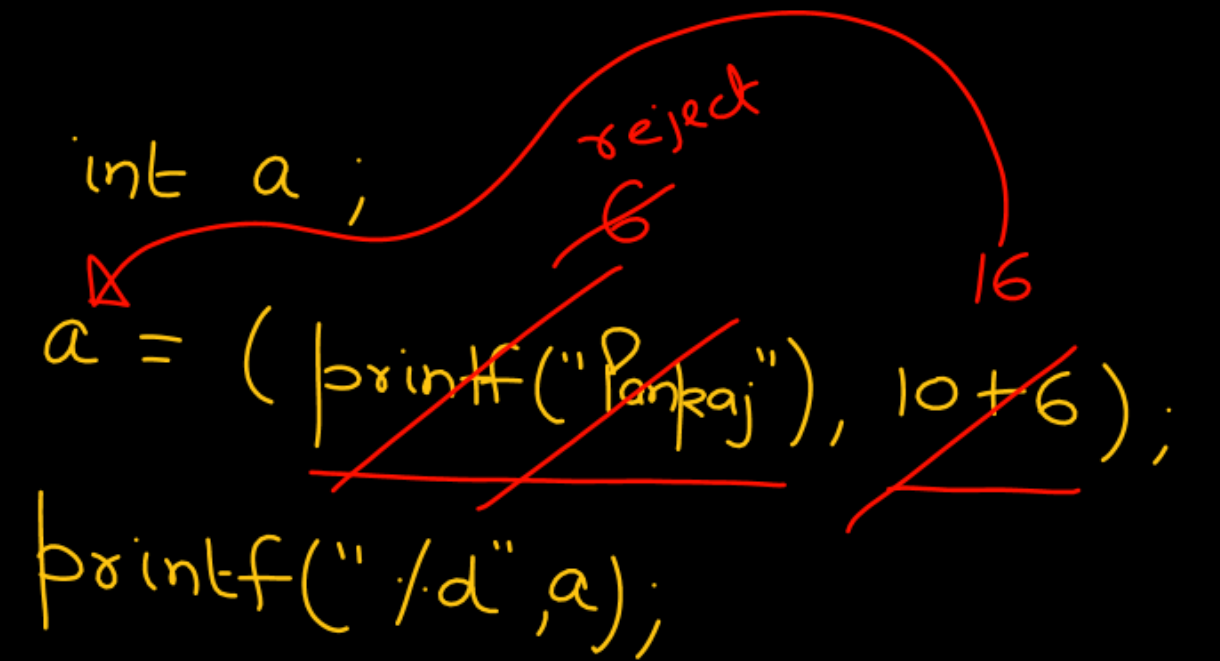
$\text{Var} = (\text{Exp1}, \text{Exp2}, \text{Exp3}, \dots \text{Expn}) ;$

→
Evaluates all these expressions
from left to right and the final
value is rightmost expression value.

```
int a;  
a = (3, 4, 5);
```



```
int a;  
a = (printf("Pankaj"), 10 + 6);  
printf("/d", a);
```



Pankaj16


```

int i, j; i 739
i = 2;
j = (i = i + 1, i = i + 6, i + 4);
printf("/d /d", i, j);
      9 13

```

O/p :

③ comma \Rightarrow last priority

int a ;

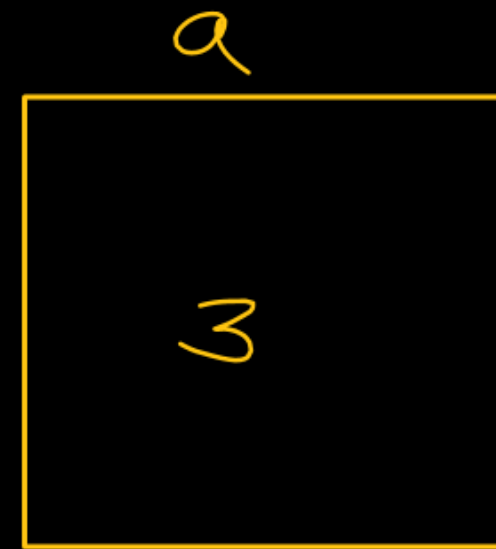
a = 3, 4, 5 ;



more
priority

(a = 3), 4, 5 ;

3, 4, 5 ;



```
void main(){
```

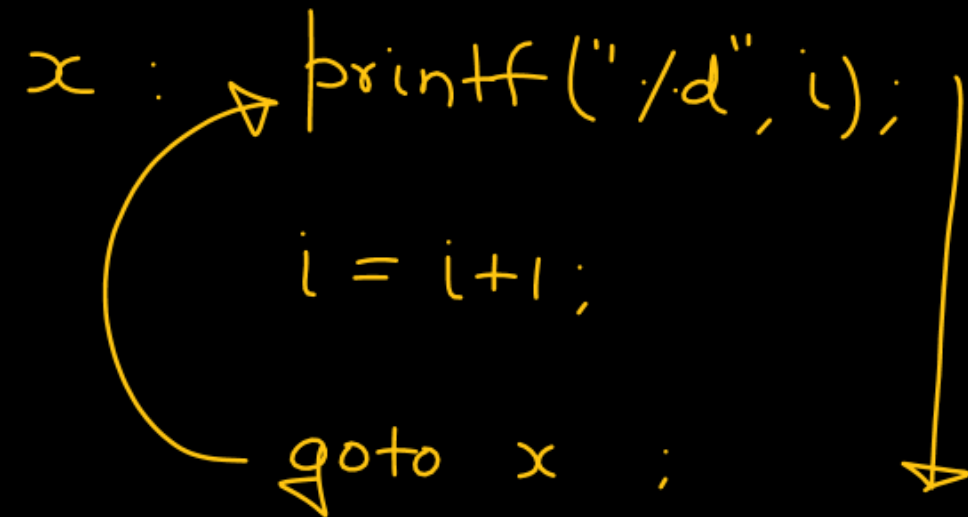
```
    int i = 10 ;
```

```
    x : printf("%d", i);
```

```
        i = i + 1;
```

```
        goto x ;
```

```
}
```



```
#define Max 20
```

```
#include <stdio.h>
```

```
void main(){
```

```
    int i;
```

```
    i = Max + 3;
```

```
    printf("/d", i);
```

```
}
```

Pre proc.



```
_____  
_____  
_____  
_____  
_____  
_____ }
```

```
void main(){
```

```
    int i;
```

```
    i = 20 + 3;
```

```
    printf("/d", i);
```

```
}
```

Q3

```
#define square(x) x*x
```

```
void main(){
```

```
    int i;
```

```
    i = square(5+2);
```

```
    printf("%d", i);
```

```
}
```



```
void main(){
```

```
    int i;
```

```
    i = 5+2 x 5 + 2
```

```
    printf("%d", i);
```

```
}
```

5+10+2

= 17


```
Union A {  
    char c ; 1  
    int i ; 4  
};
```

```
void main(){
```

```
    union A a ;
```

```
    printf("/d", sizeof(a));
```

```
    =
```

```
}
```

max(1,4)

4

```
union A {
    char c;
    int a;
};
```

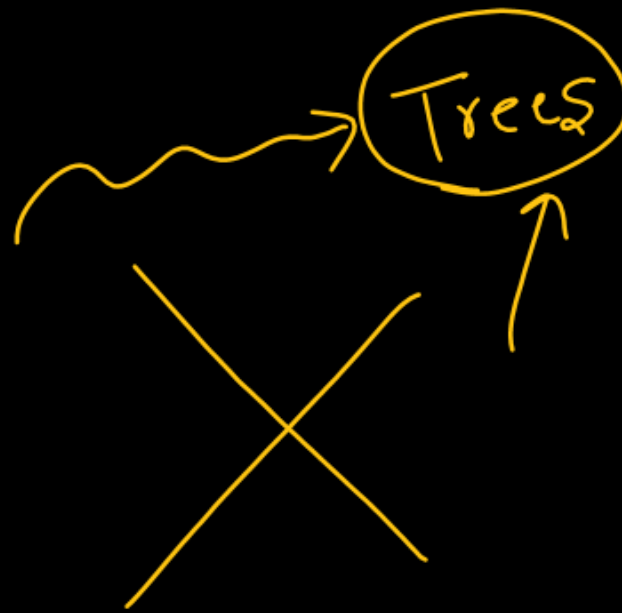
```
void main() {
```

```
    Union A x = {'A'};
```

```
    printf("/c", x.c); ✓
```

```
    printf("/d", x.a); ✓
```

```
}
```



Type Punning
Extra class



← 4 byte →

x.c

Feedback

DS

