

Computer Science & IT

C Programming



Structure & Union

One Shot



By- Abhishek Sir



Recap of Previous Lecture



Topic

String function Stolen

Topic

strcpy

Topic

strcmp

Topic

strcat

Topic

Topics to be Covered



Topic

Structure & union

Topic

Topic

Topic

Topic



Structure



Structure is collection of dissimilar Data type (user defined

```
struct student {
```

```
    char name[20];
```

```
    int RollNo;
```

```
};
```

struct student { data type

```
    char name[20] = "parakram";
```

```
    int RollNo;
```

```
}; Error
```

Structure variable can't be initialized.



Structure



create structure variable

```
struct student S1;
```

↑
S1 is a structure variable

alias → another name
or
Nickname

```
typedef struct {
```

```
    char Name[20];
```

```
    int Roll_No;
```

```
} student;
```

```
Student S1;
```



Structure



if typedef used

student S₁ = { "parakram", 10 }; ← Initialization

How to access Member of structure. for student structure
Name & RollNo are member of structure.

Operator "." Dot operator

Structure member access operator.



Structure



```
printf ("%s", s1.name); // parakram printed  
printf ("%d", s1.RollNo); // 10 printed
```



Structure



Structure pointer

```
student S1 = { 'parakram', 10 };
```

```
student *ptr; ← pointer to structure;
```

```
ptr = &S1;
```

```
printf("%s", ptr->name);
```

```
printf("%d", ptr->Roll_No);
```

using structure pointer to
access member of structure
we use \rightarrow (arrow operator)



Structure



$(*ptr).name$ \leftarrow This will also access Name.



Structure

```
#include <stdio.h>

int main(){
    struct student{

        char name[20] ;
        int roll_no ;// structure variable cant be initialized
    };

    struct student s1 = {"Arun", 10};

    printf("%s\n", s1.name);
    printf("%d", s1.roll_no);

}
```




Structure

```
#include <stdio.h>

int main(){
    typedef struct {

        char name[20] ;
        int roll_no ;// structure variable cant be initialized
    }student;

    student s1 = {"Arun", 10};

    printf("%s\n", s1.name);
    printf("%d", s1.roll_no);

}
```



Structure

```
#include <stdio.h>

int main() {
    typedef struct {

        char name[20] ;
        int roll_no ;// structure variable cant be initialized
    } student;

    student s1 = {"Arun", 10};

    student *ptr=&s1;
    printf("%s\n", ptr->name);
    printf("%d", ptr->roll_no);
}
```




Union



Union → Logical operation

OR

types of union {

```
int i; // 08
char ch; // 08
char ch1[16];
} Name;
```

-flexible

16 Bytes of memory
allocate.

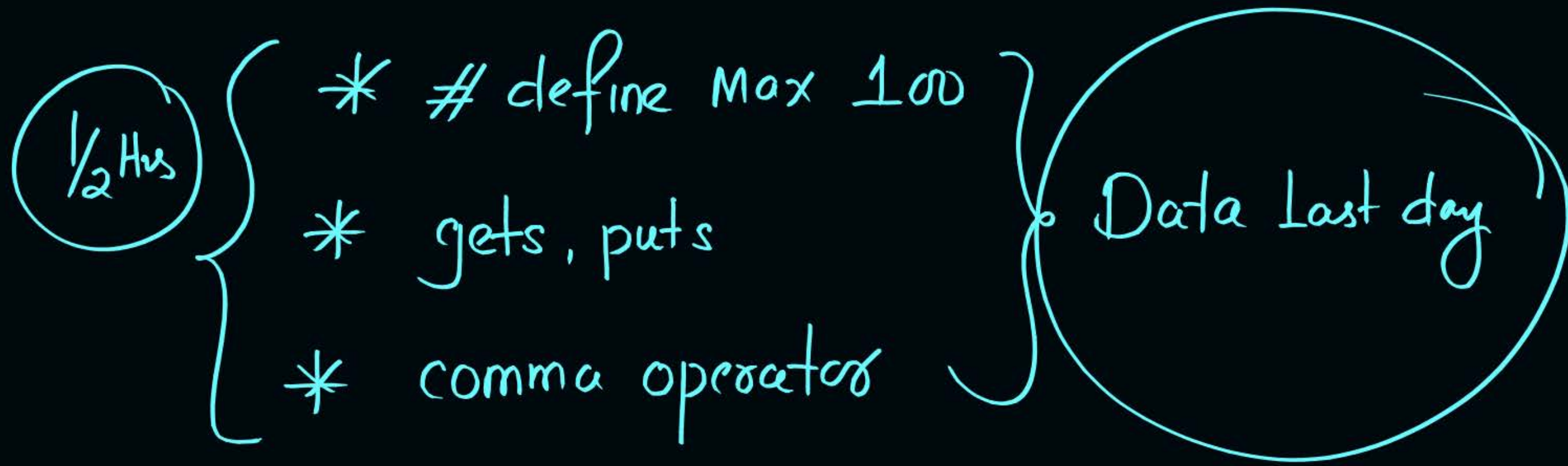
Set : duplicate

$A = \{1, 2, 3\}$ Not allowed

$B = \{3, 4, 5\}$

$A \cup B =$ meaning?!

$A \cup B = \{x \mid x \in A \text{ OR } x \in B\}$





Union

Union is collection of different data but all will be Not used Simultaneously

- * The memory allocation for union is equal to the size of largest Data type.
- * It is declare initialize and access as structure



Question

The following C declarations

```
struct node{  
    int i;  
    float j;  
};  
struct node *s[10];  
define s to be
```

Struct node

Array of structure pointers

Student s_1

Student $s_1[10]$;

- (a) An array, each element of which is a pointer to a structure of type node
- (b) A structure of 2 fields, each field being a pointer to an array of 10 elements
- (c) A structure of 3 fields: an integer, a float, and an array of 10 elements
- (d) An array, each element of which is a structure of type node



Question

Consider the following C program

```
#include<stdio.h>
struct Ournode{
char x,y,z;
};
int main(){
struct Ournode p =
struct Ournode *q = &p;
printf("%c, %c",
return 0;
}
```

'0'

The output of this program is:

- (A) 0, c ✓
- (B) 0, a+2
- (C) '0', 'a+2'
- (D) '0', 'c'

type casting

{ '1', '0', 'c' };

97+2 = 99
struct Ournode p = { '1' , '0' , 'a' +2 };
struct Ournode *q = &p;
printf("%c, %c", *((char*)q+1), *((char*)q+2));
return 0;

0 c

1	0	c
100	101	102



Question

Which one of the choices given below would be printed when the following program is executed?

```
#include<stdio.h>
struct tes{
    int i;
    char *c;
}st[5]={ {5, "becomer"}, {4, "better"}, {6, "jungle"},
{8, "ancestor"}, {7, "brother"}};
```

```
main () {
    struct tes *p=st;
    p+=1;
    ++(p->c); //
    printf("%s, ", p++->c);
    printf("%c", *++p->c);
}
```

array of structure?

tes ← Address of first element of array
Address of second element

→ more precedence
than postfix ++

- (A) jungle, n
- (B) etter, u
- (C) cetter, k
- (D) etter, n



Question



Which one of the choices given below would be printed when the following program is executed?

```
#include<stdio.h>
struct tes{
    int i;
    char *c;
}st[5]={5, "becomer", {4, "better"}, {6, "jungle"},
{8, "ancestor"}, {7, "brother"}};
```

```
main () {
    struct tes *p=st;
    p+=1;
    ++(p->c); //
    printf("%s, ", p++->c);
    printf("%c", *++(p->c));
}
```

array of structure?

ROM
better

→ more precedence
than postfix ++
(A) jungle, n
(B) etter, u
(C) cetter, k
(D) etter, n

4 101

6 201

jungle



Question

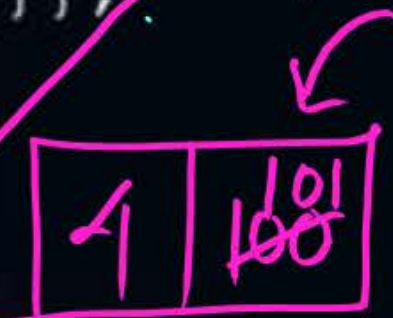
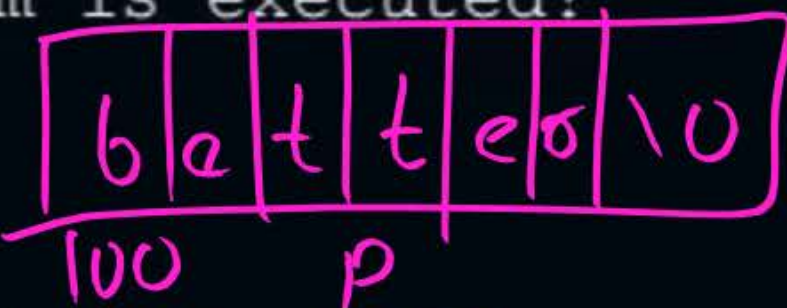
*++(p→c) *201

Which one of the choices given below would be printed when the following program is executed?

```
#include<stdio.h>
struct tes{
    int i;
    char *c;
}st[5]={ {5, "become"}, {4, "better"}, {6, "jungle"}, {8, "ancestor"}, {7, "brother"};

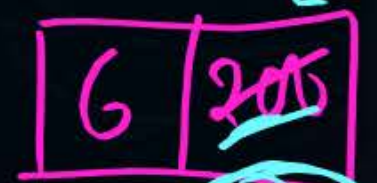
main () {
    struct tes *p=st;
    p+=1;
    ++(p->c); //
    printf("%s", "p++>c);
    printf("%c", *++p->c);
}
```

- (A) jungle, n
- (B) etter, u ✓
- (C) cetter, k
- (D) etter, n



++ Later

*++(p→c)



201





2 mins Summary



Topic

Structure

Topic

Union

Topic

Topic

Topic

Slide

THANK - YOU

