**Structure And Union**

**1.Structure:**  
1. It allocates memory equal to sum of memory allocated to its each individual member.

Program:See

***#include*** *<stdio.h>*

***struct*** *student1*

*{ //STRUCTURE*

***int*** *roll\_no;*

***char*** *name[40];*

***int*** *phone\_number;*

*};*

***union*** *student2*

*{ //UNION*

***int*** *roll\_no;*

***char*** *name[40];*

***int*** *phone\_number;*

*};*

***int*** *main(){*

***struct*** *student1 s1;*

***union*** *student2 u1;*

*printf("size of structure = %d\n",****sizeof****(s1));*

*printf("size of union = %d\n",* ***sizeof****(u1));*

***return*** *0;*

*}*

Output:

size of structure = 48  
size of union = 40

Union allocates piece of memory that is Large enough to hold the Largest variable of type in union.

2.each member have their own memory space.  
  
3. structure can not  be implemented in shred memory.  
  
4. it has less Ambiguity.  
  
5. Self referential structure can be implemented in data structure.  
  
6. All members if structure can be accessed at a time.  
  
You can watch  Structure Video here  
  
**Union:**  
  
1. It allocates piece of memory that is Large enough to hold the Largest variable of type in union.  
  
2. one block is used by all the members of union.  
  
3. Union is the Best environment where memory is shared.  
  
4. as memory is shared, Ambiguity is more in union.  
  
5. self ref. union can not be implemented.

6. only one member is accessed at a time.

Program :See

***#include*** *<stdio.h>*

***#include*** *<string.h>*

***union*** *student*

*{*

***int*** *roll\_no;*

***int*** *phone\_number;*

***char*** *name[30];*

*};*

***int*** *main()*

*{*

***union*** *student p1;*

*p1.roll\_no = 1;*

*p1.phone\_number = 1234567822;*

*strcpy(p1.name,"Brown");*

*printf("roll\_no : %d\n", p1.roll\_no);*

*printf("phone\_number : %d\n", p1.phone\_number);*

*printf("name : %s\n", p1.name);*

***return*** *0;*

*}*

Output:

roll\_no : 2003792450  
phone\_number : 2003792450  
name : Brown

So used following Way to used union better way:

Program:

***#include*** *<stdio.h>*

***#include*** *<string.h>*

***union*** *student*

*{*

***int*** *roll\_no;*

***char*** *name[30];*

***int*** *phone\_number;*

*};*

***int*** *main()*

*{*

***union*** *student p1;*

*p1.roll\_no = 1;*

*printf("roll\_no : %d\n", p1.roll\_no);*

*strcpy(p1.name,"Brown");*

*printf("name : %s\n", p1.name);*

*p1.phone\_number = 1234567822;*

*printf("phone\_number : %d\n", p1.phone\_number);*

***return*** *0;*

*}*

Output:

roll\_no : 1  
name : Brown  
phone\_number : 1234567822

**Which member of the union will be active after REF LINE?**

   #include <stdio.h>

    union temp

    {

        int a;

        float b;

        char c;

    };

    union temp s = {1,2.5,’A’}; //REF LINE

Ans:-a

**Union always point first element in such condition. For more detail see next Program.**

**#include <stdio.h>**

**union p**

**{**

**int x;**

**char y;**

**}k = {97,17};**

**int main()**

**{**

**printf("%d\n", k.x);**

**printf("%c\n", k.x);**

**printf("%d\n", k.y);**

**k.x=23;**

**printf("%d\n", k.x);**

**}**

**Output:**

**97**

**a**

**97**

**23**