

C-114 - Union in C

* Union is also a user defined data type like structures.

* Everything like declaration, initialization of accessing of members using objects are same as in structures for unions also.

* But only difference is memory allocation and keyword union.

* In structure for each structure member with different datatypes will allocate members separately.

Ex:- struct demo{

int a;

char b;

float c;

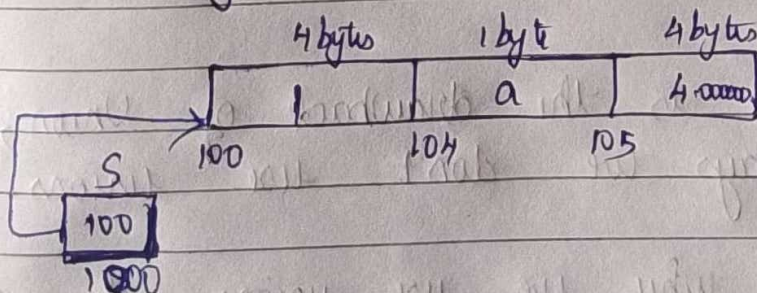
};

void main()

{ struct demo s = {1, 'a', 4.0};

printf("%d %c %f", s.a, s.b, s.c);

}



* Memory is allocated as $4 + 1 + 4 \Rightarrow 9$ bytes.

* In union; memory is allocated based on the maximum number of data types which have maximum memory allocation.

Ex.:

Union demo

```

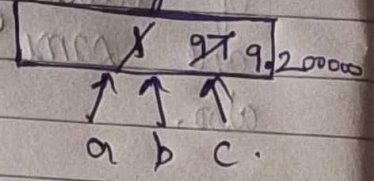
4 bytes → int a;
1 byte  → char b;
4 bytes → float c;
};
void main()
{

```

maximum
4 bytes

Memory allocation of Union

4 bytes



* Only 4 bytes of memory is allocated.

union demo U;

U.a = 1;

U.b = '97';

U.c = 9.2;

```

printf("a = %d", U.a); → garbage value
printf("b = %c", U.b); → garbage value
printf("c = %f", U.c); → 9.20000

```

* Since the data is overwritten and what value we put last, only that value exist others will have garbage value.

* This is the drawback of Union; so nowadays we don't use unions.

* But why we use unions; if we have like 5 to 10 members of structures

and at one time we want to process only one member of structure; it will leads to wastage of memory because of other members, but this can be done in Union without wastage of memory.

* In union we can process only one member at one time without a wastage of memory. but drawback is we can't process all the members of a ~~structure~~ union at one time.

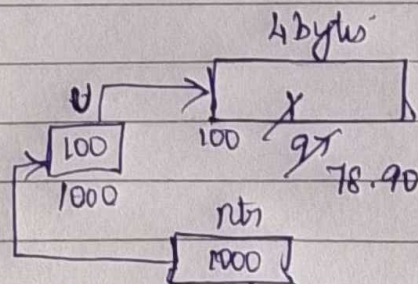
* Array of ~~struct~~ unions and pointer to unions are same as structures.

Ex: union abc

```

{
    int a;
    char b;
    float c;
}
3;
void main()
{

```



```

    union abc u; union abc (*ptr) = &u;

```

```

    u.a = 1;

```

```

    u.b = 97;

```

```

    u.c = 78.90;

```

```

    printf("a value is %d", ptr->a);

```

```

    printf("b value is %c", ptr->b);

```

```

    printf("c value is %f", (*ptr).c);

```

```

}

```

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 15-UNION IN C **/
4  union student
5  {
6      int a;
7      char b;
8      float c;
9  };
10 union student u;
11 int main()
12 {
13     union student (*ptr)=&u;
14     printf("Enter values of a b and c\n");
15     scanf("%d %s %f",&(*ptr).a,&(*ptr).b,&(*ptr).c);
16     printf("Entered values of a b and c are:\n");
17     printf("%d\n",ptr->a);
18     printf("%s\n",(*ptr).b);
19     printf("%0.2f\n",(*ptr).c);
20
21     getch();
22 }
23

```

```

D:\1. C++\NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 8_JENNY'S LECTURE_STRUCTURES\15_UNI...
Enter values of a b and c
1120272384
(null)
99.00

```

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 15-UNION IN C (1) **/
4  union student
5  {
6      int rollno;
7      char name[20];
8      float marks;
9  };
10 union student s;
11 int main()
12 {
13     union student (*ptr)=&s;
14     printf("Enter student details\n");
15     scanf("%d %s %f",&(*ptr).rollno,&(*ptr).name,&(*ptr).marks);
16     printf("Entered values of students are:\n");
17     printf("%d\n",ptr->rollno);
18     printf("%s\n",(*ptr).name);
19     printf("%0.2f\n",(*ptr).marks);
20
21     getch();
22 }
23

```

```
"D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 8_JENNYS LECTURE_STRUCTURES\15_UNI...
Enter student details
1
Ji
98
Entered values of students are:
1120141312
98.00
```