PROGRAMMING IN C

Structure and Union

CONTENTS

STRUCTURE

- A structure is a user-defined data type available in C that allows to combining data items of different kinds. Structures are used to represent a record.
- Structure stores the different types of elements i.e heterogeneous elements.
- E.g. :

```
struct student
{
int rollno;
char name[60];
}s1;
```

DEFINING A STRUCTURE

- **Defining a structure:** To define a structure, you must use the **struct** statement. The struct statement defines a new data type, with more than one member. The format of the struct statement is as follows:

```
struct [structure name]
{
member definition;
member definition; ...
 member definition; };
```

EXAMPLE

```
#include <stdio.h>
 #include <string.h>
 struct student
 {
 int rollno;
char name[60];
}s1;
 void main( )
{
s1.rollno=1;
strcpy(s1.name, “intelligent”);
printf( "Rollno : %d\n", s1.rollno);
printf( "Name : %s\n", s1.name);
}
```

SELF REFERENTIAL STRUCTURES

- Self Referential structures are those structures that have one or more pointers which point to the same type of structure, as their member.
- In other words, structures pointing to the same type of structures are self-referential in nature.
- Example:

```
struct node
{
   int data1;
   char data2;
   struct node* link;
};
void main()
{
   struct node ob;
}
```

ARRAY OF STRUCTURE

- Structure is used to store the information of One particular object but if we need to store such 100 objects then Array of Structure is used.
- **Example :**

```
struct Bookinfo
{ char bname[20];
int pages;
int price;
}b1[100];
```

Explanation :

- Here Bookinfo structure is used to Store the information of one Book.
- In case if we need to store the Information of 100 books then Array of Structure is used.
- b1[0] stores the Information of 1st Book , b1[1] stores the information of 2nd Book and So on We can store the information of 100 books.

UNION

- A union is a special data type available in C that allows storing different data types in the same memory location.
- You can define a union with many members, but only one member can contain a value at any given time. Unions provide an efficient way of using the same memory location for multiple purposes.
- Union takes the memory of largest member only so occupies less memory than structures.

Example

```
union car
{ char name[50];
int price;
};
void main()
{
 union car car1, car2, *car3;
Car1.price=200000;
}
```

Differences

	STRUCTURE	
Keyword	The keyword **struct** is used to define a structure	The keyword
Size	When a variable is associated with a structure, the compiler allocates the memory for each member. The size of structure is **greater than or equal to the sum of sizes of its members.**	when a variat allocates the largest memc **of largest me**
Memory	Each member within a structure is assigned unique storage area of location.	Memory alloc union.
Value Altering	Altering the value of a member will not affect other members of the structure	Altering the v member valu

References

- Programming in C, 2011, by J.B. Dixit
- Basics of C Programming, 2011, by J.B. Dixit
- https://www.tutorialspoint.com/cprogramming/c_unions.htm
- https://www.programiz.com/c-programming/c-unions
- https://intellipaat.com/tutorial/tutorial-c-tutorial/structures-union/