UNIT V

Structure & Union.

Structure.

(Real world Example (Engineer, Doctor - fever) Derived Datatype).

It is a collection of one or more variables of different datatype grouped together under a single name.

Example: To maintain multiple student's records (student name, rollno, marks).

Different datatypes are required. when we use multiple arrays. The source code gets increased in this case.

Declaring & Initialization of Structures.

```
Structure   name   Structure_tag.
{
    datatype    var_name1;
    datatype    var_name2;
};
```

(Structure elements (2 fields))

Example:

```
Structure book
{
    char  book[30];
    int   pg_no;
    float price;
};
Struct book b1;
```

b1 ← Structure Variables/member.

Customer details

Employee details

* Array → build-in datatype.
* Structure → user defined.

```
Struct student
{
    char name[30];
    int  roll_no;
    int  mark;
} s1, s2;
```

s_1 → Variable is created with 34 bytes (30 - char, int - 2+2).

```
s1 = { "Shobana", 96, 92 };
```

* Accessing done through .

Syntax: struct_variable . members

Example: S1 . rollno

↳ period is used to access members of structure

* S1. rollno = 96 ;
S1. name = "Shobana" ;
S1. mark = 96 ; } Assigning value to members

* int a ; Variable → holds single value

int a[10]; Array → holds 10 values of same datatype integer

* When dealing with entities → collection of dissimilar datatype .

Example: students details → name - char
age - int
marks - int array .

Defining a Structure

* First must be defined for its format .
* later on the variables of structure is created .
* It is a programmer defined one
* struct keyword is used to define the structure .

* Example:

```
struct tag_name
{
    data_type member1;
    data_type member2;
    :
};
```

*

```
struct stud
{
    int age;
    char name[30];
};
```

] members of structure.

S1; S1 → int 2bytes + char 1byte

using structure variable, we can access the members of structure.

S1.age S1.name

* Array Vs Structure.

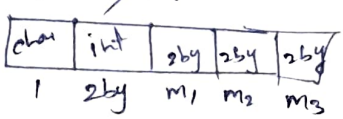
Array	Structure
1. Collection of Similar Datatype	1. Collection of Dissimilar Datatype
2. Derived Datatype	2. Programmer-defined Datatype.

```
int a[10];
struct
{
    int a[10];
    char b;
};
```

Declaring Structure Variables.

```
struct student
{
    char name[20];
    int age;
    int m1, m2, m3;
} s1;
```

S1 => 1+2+2+2+2



S1 => 9 memory Add

Initialization.

```
Void main()
{
    Struct emp
    {
        int empno;
        char empnm[30];
        int salary;
    };
    Struct emp e={23, "Shobana", 4500};
    clrscr();
    Printf("The name = %s \t no = %d \t sal = %d", e.name
                                e.empno, e.salary);
    getch();
}
```

Output: The name = Shobana no = 23 Sal = 4500.

Example : Structure & Arrays - as variable.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    struct emp
    {
        int empno;
        char empnm[30];
        float salary;
    }e[5];
    int i;
    clrscr();
    for (i=0; i<5; i++)
     scanf ("%d %s %f", &e[i].empno, &e[i].empnm, &e[i].salary);
    for (i=0; i<5; i++)
      Printf ("%d %s %f", e[i].empno, e[i].empnm, e[i].salary);
    getch();
}
```

Programs - student details using structure.

```
Void main()
{
    Struct stud
    {
        int rollno;
        char name[30];
        int m1, m2, m3;
        float avg;
    } s[30];
    int i, n, total;
    Printf ("In Enter the no. of students");
    scanf ("%d", &n);
    Printf (" Enter the student details ");
```

```
for(i=0;  i<n ; i++)
{
        printf (" Student %d details ", i );
        scanf (" %d  %s  %d %d  %d",
                &s[i].rollno, &s[i].name, &s[i].m1,
                        &s[i].m2, &s[i].m3);

        total = s[i].m1 +s[i].m2 + s[i].m3;
        s[i].arg = total /3;
}
printf ("\n\t Student  Details\n" );
printf ("\n\t rollno\t name\t mark1 \t mark2 \t mark3 \t Avg\n");
for (i=0;i<n; i++)
{
 printf ("\t%d \t %s \t %d\t %d \t %d \t %f \n",
                s[i].rollno, s[i].name, s[i].m1, s[i].m2,
                        s[i].m3, s[i].arg );
}
getch();
}
```