

## Unit - v

### Structure and Union.

#### Structure:

- It is a user defined data types
- It can be used to group items of possibly diff. types into a single type.
- keyword: Struct.

#### Syntax:

Struct Struct-name → keyword name of the structure

{

data-type member-name1;

data-type member-name2;

data-type member-name n;

} ;

struct employee

{ int id;

char name[10];

float salary;

} ;

## Declaring a structure variable;

To declare a structure variable, we have to use the following syntax:

```
struct struct-name  
{  
    data type member 1 - name;  
    data type member 2 - name;  
    :  
    data type member n - name;  
}  
variable-name;  
    (or)  
struct struct-name variable-name;
```

Def: There are 2 ways to declare struct variables.

- By struct keyword within main() fun.
- By declaring a variable at the time of defining the structure.

eg:

```
struct stud-name
```

```
{ int rollno;
```

```
    char name[15];
```

```
} stud;
```



## Initialization of Structure:

→ It can be done in ways,

① using dot (.) operator.

②. all the member of Structure can be initialized one statement.

① → using Dot (.) operator: -

→ To initialize the Structure member using dot (.) operator, we have to use following Syntax:

Syntax: Structure Variable name . member = Value;

eg:

Std. rno = 654 ;

Std. name = "Sai";

Std - branch = "CST";

② → To initialize all the Structure members in one Statement, we have to use the following Syntax:

Syntax: Struct Structure name Variable name = { value 1, value 2, ... value n };

eg: Struct Student Std = { 1, "Sairam" };

Ex:

```
struct student
{
    int rno;
    char name[20];
    char branch[10];
} std;
```

Dot (.)

```
std.rno = 654;
std.name = "Sai";
std.branch = "CST";
```

### Accessing the Structure Members:

→ We have use 2 operators

- ① Dot (.) operator
- ② Arrow approach.

#### Dot (.) operator:

used to access the members of the structure.

#### Syntax:

Structure Variable name . member name;

eg:

```
std.rno;
std.name;
std.address;
```

```
struct student
{
    int roll no;
    char name[20];
    char address[20];
} std;
```

#### Arrow ~~approach~~ operator (→):

It is used to access the members of the structure of the structure variable is a pointer variable.



eg:

```
struct Student  
{  
    int roll no;  
    char name[20];  
    char address[20];  
};
```

std;

std → roll no;

std → name;

std → address.

sample program:

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void main()
```

```
{  
    struct Student
```

```
{  
    int num;
```

```
    char name[10];
```

```
    float avg;
```

```
};
```

```
clrscr();
```

```
struct Student s = {1205, "Sai", 70.8};
```

```
printf("Student Roll no: %d\n", s.num);
```

```
printf("Student name is: %s\n", s.name);
```

```
printf("Student avg Avg of student: %.2f", s.avg);
```

```
getch();
```

```
}
```

O/p:

Student Roll No: 1205

Student Name: Sai

Average of the student: 70.8