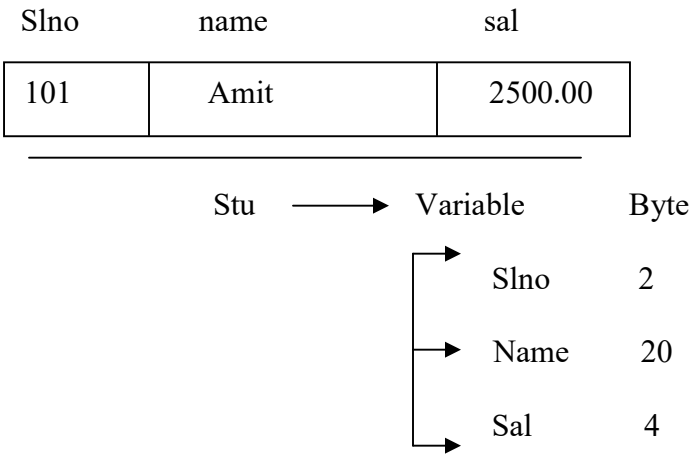
Structure

- Structure is a Collection of variable having a different data type
- Such a Record

Example Suppose you want to store data of student information
101 Amit 2500.00
Int slno
Char name[20]
Float sal



- In the Structure is Declared keyword **struct**
- Under surrounded by braces which describe the member of the structure.
- Here stu is a variable having slno, name and sal

Define s Structure

```
Struct structure_name
{
        member 1;
        member 2;
        ---
        ---
        member n;
};
```

Example

```
Struct student
{
        int slno;
        char name[20];
        float sal;
};
```

These statement define a new data type called student. Each variable of this data type will collection of integer variable called slno, charactype variable called name and float type variable called sal.

Declare a Structure type variable

struct student stu

Here stu is a variable having slno, name and sal

There are two type of variable declare

Struct student
{
int slno;
char name[20];
float sal;
};
struct student stu1, stu2,stu3;
	Struct student
	{
	int slno;
	char name[20];
	float sal;
	}stu1, stu2, stu3;

Acessing a Structure

a) struct student stu={101, “Computer”,2500.00};

b)
```
stu.slno=101;
strcpy(stu.name, “Kumar”);
stu.sal=2500.001
```

Printing a Structure variable

printf("\n %d %s %f",stu.slno,stu.name, stu.sal);

Read a structure variable

```
int n;
char str[20];
float rs;

printf("\n Enter the Serial No ");
scanf("%d",&n);
printf("\n Enter the Name ");
scanf("%s",str);
printf("\n Enter the Amount salary ");
scanf("%f",&rs);

stu.slno=n;
strcpy(stu.name, str);
stu.sal=rs;
```

Display sizeof structure

```
Int n;
n=szeof(stu);
printf("\n Size of  Structure %d",n);                    Output        28
```

Structure operator

'.' Direct member access
'->' Indirect member access These operator is used to pointers.

```
Struct student
{
        int slno;
        char name[20];
        float sal;
};
```

Simpe Method Access	Pointer Method Access

struct student stu;
stu.slno=n;
strcpy(stu.name, str);
stu.sal=rs;
printf("\n roll no %d", stu.roll);
printf("\n name %s ",stu.name);
printf("\n salary %f", stu.sal);
	struct student stu;
	struct student *ptr;
	ptr=&stu;
	ptr->slno=n;
	strcpy(ptr->name,n);
	ptr->sal=rs;
	printf("\n roll no %d", ,ptr->roll);
	printf("\n name %s ",ptr->name);
	printf("\n salary %f", ptr->sal);

1. **Write a C program to create a structure simple method**

```
Void main()
{
        struct student
        {
                int roll;
                char name[20];
                float sal;
        };
        int r;
        char n[20];
        float s;
        struct student stu;
        clrscr();
        printf("\n Enter Roll Number ");
        scanf("%d",&r);
        printf("\n Enter Name       ");
        scanf("%s",n);
        printf("\n Enter Salary     ");
        scanf("%f",&s);
        stu.roll=r;
        strcpy(stu.name,n);
        stu.sal=s;
```

```
        printf("\n Roll Number %d",stu.roll);
        printf("\n Name       %s",stu.name);
        printf("\n Salary     %f",stu.sal);
        getch();
}
```

2. **Write a C program to create a structure pointer method**

```
void main()
{
        struct student
        {
                int roll;
                char name[20];
                float sal;
        };
        int r;
        char n[20];
        float s;
        struct student stu;
        struct student *ptr;
        clrscr();
        ptr=&stu;
        printf("\n Enter Roll Number ");
        scanf("%d",&r);
        printf("\n Enter Name       ");
        scanf("%s",n);
        printf("\n Enter Salary     ");
        scanf("%f",&s);
        ptr->roll=r;
        strcpy(ptr->name,n);
        ptr->sal=s;
        printf("\n Roll Number %d",ptr->roll);
        printf("\n Name       %s",ptr->name);
        printf("\n Salary     %f",ptr->sal);
        getch();
}
```