

Structure in C programming.

1. Structure is a userdefined data type
2. It is used to group different types of data types

Syntax:- `struct Student` → It can be any name
 {
 int a;
 char b;
 float c;
 long int d;
 char ;
 :
 };

↑
 Structure definition (layout of structure)

3. Defining a structure doesn't allocate memory
4. Memory is allocated when we create a structure variable

Example:- `struct student s1;` → this is structure variable.

5. Size of structure (How many bytes it will take in memory)
 Is sum of it's all members memory.

Example:-

① `struct student`
 {
 char a;
 };

Size of student is 1 Byte

int → datatype
 student → datatype

② `struct student`
 {
 char a;
 char b;
 char c;
 };

Size is $1+1+1 = 3$ Bytes

③ `struct family`
 {
 char a; - 1 Byte
 char name[10]; - 10 Bytes
 };

Size of structure is 11 Bytes

5. Size of structure can also be calculated using "sizeof"

Example:- `sizeof (struct family)` → 11 Bytes

`struct family myfamily;`

`sizeof (myfamily);` → 11 Bytes

`sizeof (int)` → 4 Bytes

`int a;`
`sizeof (a)` → 4 Bytes

`int int;` X error

6. To create variable of any structure "struct" is must.

`struct family myfamily;` ✓

`family myfamily;` X error

7. Structure members are accessed using dot (.) operator.

`struct student`

{
 char sex;
 int id;
 char name[50];
 char address[100];
}

Size = 115

};

`struct student st;`

`st.sex = 'M';`
`st.id = 10005;`

`st.name = "Divyansh";`
`strcpy (st.name, "Divyansh");`

`strcpy (st.address, "#123 ST1");`

`printf ("In student is : %c", st.sex);`

`printf ("In student id is : %d", st.id);`

`printf ("In student's name : %s", st.name);`

`printf ("In student's address : %s", st.address);`

sex	M
id	10005
name	Divyansh
add.	#123 ST1
	XXXX

8. Padding may be added by the compiler to ^{satisfy} the alignment requirement

`struct xyz`

{
 char a;
 char b;
}

};

⇒ Size: 2 Bytes

`struct xyz`

{
 char a;
 char b;
 int c;
}

};

⇒ Size: 6 Byte
: 8 Bytes.

0	a
1	b
2	c

0	a
1	b
4	c

0	a
1	b
2	—
3	—
4	size c

} 2 Bytes.