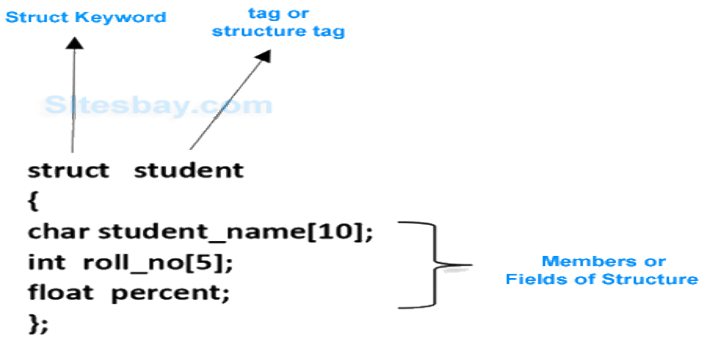
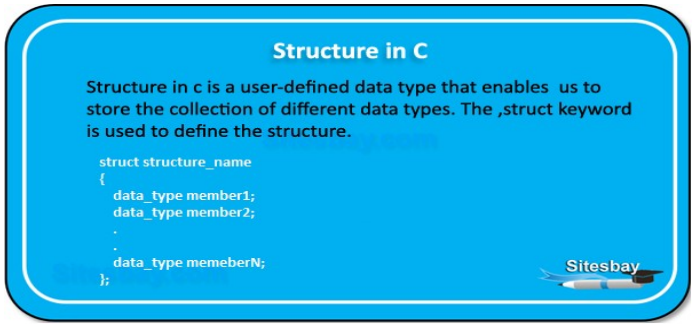
AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

CHAPTER - 8
STRUCTURE





Subject : PPS Code : 3110003	Prepared By: Asst. Prof. Rupali Patel (CSE Department, ACET)	AMIRAJ COLLEGE OF ENGINEERING & TECHNOLOGY

Structure

Arrays allow to define type of variables that can hold several data items of the same kind.

Similarly **structure** is another user defined data type available in C that allows to combine data items of different kinds.

A structure is a user defined data type in C.

A structure isn't a variable type. Instead, think of it as a frame that holds multiple variable types. In many ways, a structure is similar to a record in a database.

How to create a structure?

'**struct**' keyword is used to create a structure.

Following is an example.

```
struct address
{
   char name[50];
   char street[100];
   char city[50];
   char state[20];
   int pin;
};
```

How to declare structure variables?

A structure variable can either be declared with structure declaration or as a separate declaration like basic types.

```
struct Point
{
  int x, y;
} p1;
```

Another way:

```
struct Point
{
  int x, y;
};

void main()
{
  struct Point p1;
}
```

AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

How to initialize structure members?

Structure members **cannot be** initialized with declaration. For example the following C program fails in compilation.

```
struct Point
{
  int x = 0;  // COMPILER ERROR:  cannot initialize members here
  int y = 0;  // COMPILER ERROR:  cannot initialize members here
};
```

The reason for above error is simple, when a datatype is declared, no memory is allocated for it. Memory is allocated only when variables are created.

How to initialize structure members? (cont..)

Structure members **can be** initialized using curly braces '{}'. For example, following is a valid initialization.

```
struct Point
{
   int x, y;
}p1;

int main()
{
    p1 = {0, 1};
}
```

How to access structure elements?

Structure members are accessed using dot (.) operator.

```
#include<stdio.h>
struct Point
{
   int x, y;
};

int main()
{
   struct Point p1 = {0, 1};
   p1.x = 20;
   printf ("x = %d, y = %d", p1.x, p1.y);
 }
```

AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

Nested Structure

C provides us the feature of nesting one structure within another structure by using which, complex data types are created.

For example, we may need to store the address of an entity employee in a structure. The attribute address may also have the subparts as street number, city, state, and pin code.

Hence, to store the address of the employee, we need to store the address of the employee into a separate structure and nest the structure address into the structure employee.

Nested Structure Example

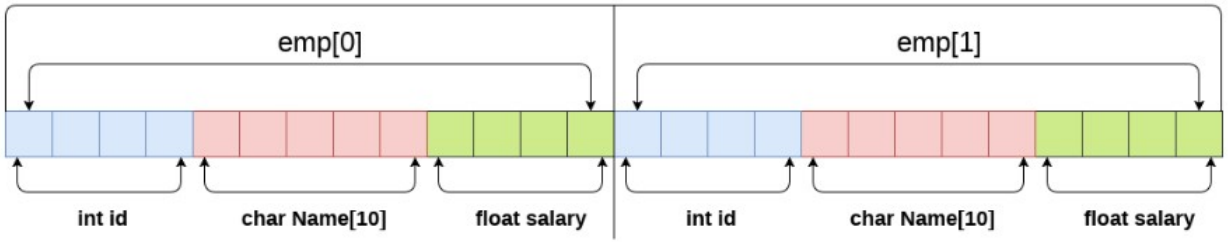
```
#include<stdio.h>
struct address
{
        char city[20];
        int pin;
        char phone[14];
};
struct employee
{
        char name[20];
        struct address add;
};
void main()
{
struct employee emp;
printf("Enter employee
information?\n");
scanf("%s %s %d %s",emp.name,
emp.add.city, emp.add.pin,
emp.add.phone);
printf("Printing the employee
information....\n");
printf("name: %s\n City: %s\n
Pincode: %d\n Phone: %s",
emp.name, emp.add.city,
emp.add.pin, emp.add.phone);
}
```

AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

Array of Structure

An array of structures in C can be defined as the collection of multiple structures variables where each variable contains information about different entities.

The array of structures is also known as the collection of structures.




```
struct employee
{
    int id;
    char name[5];
    float salary;
};
struct employee emp[2];
```

sizeof (emp) = 4 + 5 + 4 = 13 bytes

sizeof (emp[2]) = 26 bytes

AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

Array of Structure Example

```
#include<stdio.h>
#include <string.h>
struct student
{
int rollno;
char name[10];
};
int main()
{
int i;
struct student st[5];
  printf("Enter Records of 5 students");
for(i=0;i<5;i++)
{
printf("\nEnter Rollno:");
scanf("%d",&st[i].rollno);
printf("\nEnter Name:");
scanf("%s",&st[i].name);
}
printf("\nStudent Information List:");
for(i=0;i<5;i++)
{
printf("Rollno:%d, Name:%s",
st[i].rollno, st[i].name);
}
    return 0;
}
```

AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

Passing structure to function

A structure can be passed to any function from main function or from any sub function.

Structure definition will be available within the function only.

It won't be available to other functions unless it is passed to those functions by value or by address(reference).

Else, we have to declare structure variable as global variable. That means, structure variable should be declared outside the main function. So, this structure will be visible to all the functions in a C program.

Passing structure to function

```
#include <stdio.h>
#include <string.h>
struct student
{
        int id;
        char name[20];
        float percentage;
};
void func(struct student record);
int main()
{
        struct student record;
        record.id=1;
        strcpy(record.name, "Raju");
        record.percentage = 86.5;
        func(record);
        return 0;
}

void func(struct student record)
{
printf(" Id is: %d \n", record.id);
printf(" Name is: %s \n",
record.name);
printf(" Percentage is: %f \n",
record.percentage);
}
```

AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

Structure using Pointer

To use the array of structure variables efficiently, we use **pointers of structure type**. We can also have pointer to a single structure variable, but it is mostly used when we are dealing with array of structure variables.

Dot(.) operator is used to access the data using normal structure variable and arrow (->) is used to access the data using pointer variable.

Structure using Pointer Example

```
#include <stdio.h>
#include <string.h>
 struct student
{
    int id;
    char name[30];
    float percentage;
};
int main()
{
    int i;
    struct student record1 = {1,
"Raju", 90.5};
    struct student *recd;
    recd = &record1;
    printf("Records of STUDENT1:
\n");
    printf("  Id is: %d \n", recd->id);
    printf("  Name is: %s \n", recd-
>name);
    printf("  Percentage is: %f \n\n",
recd->percentage);
    return 0;
    }
```

AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY