Programming in C
(2-D Array)

Prepared By

Alok Haldar

Assistant professor
Department of Computer Science & BCA
Kharagpur College

C Structure

Why use structure?

In C, there are cases where we need to store multiple attributes of an entity. It is not necessary that an entity has all the information of one type only. It can have different attributes of different data types. For example, an entity **Student** may have its name (string), roll number (int), marks (float). To store such type of information regarding an entity student, we have the following approaches:

- Construct individual arrays for storing names, roll numbers, and marks.
- Use a special data structure to store the collection of different data types.

Let's look at the first approach in detail.

What is Structure :

Structure in c is a user-defined data type that enables us to store the collection of different data types.

Each element of a structure is called a member.

The **,struct** keyword is used to define the structure. Let's see the syntax to define the structure in c.

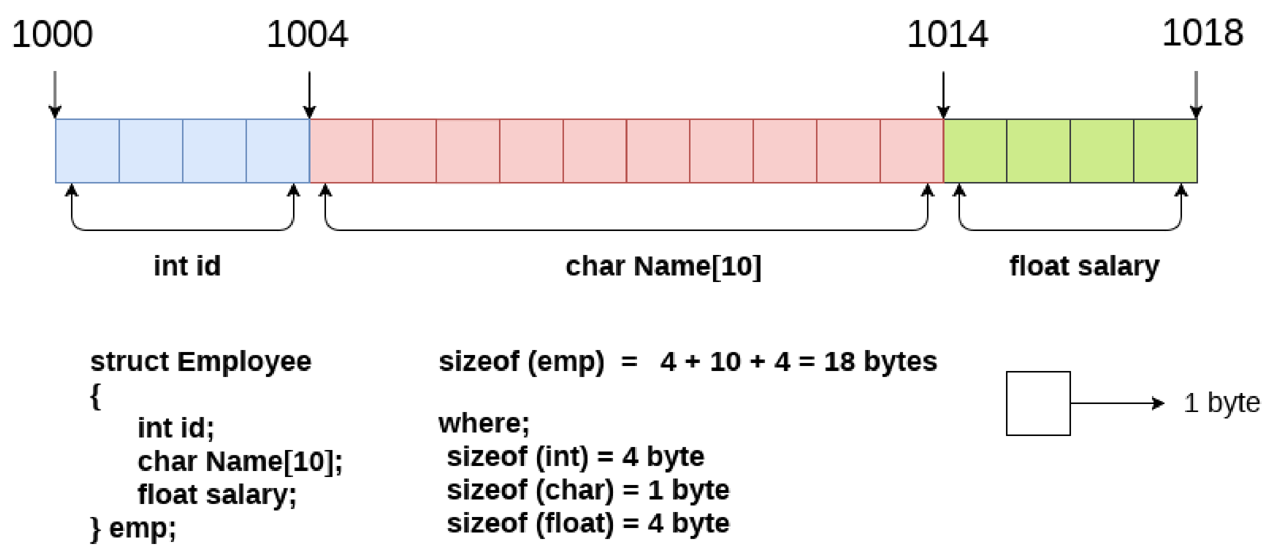
```
Struct structure_name
        {
        data_type member1;
        data_type member2;
        .
        .
        data_type memeberN;
        };
```

Let's see the example to define a structure for an entity employee in c.

```
Struct employee
        {
           int id;
           char name[20];
           float salary;
        };
```

The following image shows the memory allocation of the structure employee that is defined in the above example.

Here, struct is the keyword; employee is the name of the structure; id, name, and salary are the members or fields of the structure. Let's understand it by the diagram given below:



Declaring structure variable :

We can declare a variable for the structure so that we can access the member of the structure easily. There are two ways to declare structure variable:

1. By struct keyword within main() function
2. By declaring a variable at the time of defining the structure.

1st way:

Let's see the example to declare the structure variable by struct keyword. It should be declared within the main function.

```
Struct employee
        {
         int id;
         char name[50];
         float salary;
        };

struct employee e1, e2;
```

Let's see another way to declare variable at the time of defining the structure.

```
Struct employee
    {
      int id;
      char name[50];
      float salary;
    }e1,e2;
```

Accessing members of the structure

There are two ways to access structure members:

1. By . (member or dot operator)
2. By -> (structure pointer operator)

Let's see the code to access the *id* member of *p1* variable by. (member) operator.

```
p1.id

p1->id
```

C Structure example

```
#include<stdio.h>
    #include<string.h>
    struct employee
    {
      int id;
      char name[50];
    }e1;//declaring e1 variable for structure
    int main()
    {
    //store first employee information
    e1.id=100;
    strcpy(e1.name, "Amit Sharma");//copying string into char array
    //printing first employee information
    printf("employee  id :%d\n",e1.id);
    printf("employee  name:%s\n",e1.name);
    return 0;
    }
```

Output:

```
Employee id : 100
employee name : Amit Sharma
```

```
#include<stdio.h>
        void main ()
        {
        char names[2][10],dummy;//2-dimensioanal character array names is used to store the names
        of the students
        int roll_numbers[2],i;
        float marks[2];
        for(i=0;i<3;i++)
        {
        printf("Enter the name, roll number, and marks of the student %d",i+1);
        scanf("%s %d %f",&names[i],&roll_numbers[i],&marks[i]);
        scanf("%c",&dummy); // enter will be stored into dummy character at each iteration
        }
        printf("Printing the Student details ..\n");
        for(i=0;i<3;i++)
        {
        printf("%s %d %f\n",names[i],roll_numbers[i],marks[i]);
        }
        }
```

Output :

```
Enter the name, roll number, and marks of the student 1 Amit 80 81

Enter the name, roll number, and marks of the student 2 Sumit 45 66
Enter the name, roll number, and marks of the student 3 Romit 88 77

Printing the Student details...
Amit 80 81.000000
Sumit 45 66 .000000
Romit 88 77.000000
```