C Structure

- Structure is a user-defined data type, which is used to store logically similar data of different data types together.
- All the structure elements are stored at contiguous memory locations.
- Structure type variable can store more than one data item of varying data types under one name.
  - Syntax:

```
struct structureName {
   dataType member1;
   dataType member2;
   ...
};
```

Example:

```
struct Employee {
    int id;
    char name[50];
    float salary;
};
```

C Structure

- Creating a structure variable:

```c
struct Employee {
  // code
};

int main() {
  struct Employee e1, e2, e[20];
  return 0;
}
//------Another way of creating a struct variable is:
struct Employee {
   // code
} e1, e2, e[20];
```

  - e1 and e2 are variables of structure Employee.
  - e[] is an array of struct Employee type which is of size 20.

C Structure

- Accessing members of a structure:
  - Structure members can be accessed using two ways as follows:
    - By . (Dot or member operator)
    - By -> (structure pointer operator)

C Structure

- Accessing members of structure using dot(.) operator:

```c
#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=101;
   strcpy(e1.name, "Dennis Ritchie");//copying string into char array
   //printing first employee information
   printf( "employee 1 id : %d\n", e1.id);
   printf( "employee 1 name : %s\n", e1.name);
return 0;
}
```

C Structure

- typedef - It is a keyword which is used to give an alias name to an already existing variable.

```c
#include <stdio.h>
typedef struct student {
    char name[20];
    int age;
}stud;
int main()
{
    stud s1;
    printf("\nEnter the name of the student:");
    scanf("%s",&s1.name);
    printf("\nEnter the age of student:");
    scanf("%d",&s1.age);
    printf("\n Name of the student is : %s", s1.name);
    printf("\n Age of the student is : %d", s1.age);
    return 0;
}
```