Structures in C

Prof. Harish D.G.
Dept. of Computer and IT
College of Engineering,Pune
www.harishgadade.com

Structures in C

- Definition
- Define Structures
- Create a Structure variable.
- Accessing members of a Structures
- Why structures in C?

Definition

- a struct (or structure) is a collection of variables (can be of different types) under a single name.

Defining a Structure

- Before you can create structure variables, you need to define its data type. To define a struct, the `struct` keyword is used.

Syntax:

```
struct structName
{
        dataType var1;
        dataType var2;
        ……
};
```

Example:

```
struct student
{
        int prn;
        Char name[20];
        float per;
};
```

Here, a derived type `struct student` is defined. Now, you can create variables of this type.

Creating a Structure Variable

- When a struct type is declared, no storage or memory is allocated. To allocate memory of a given structure type and work with it, we need to create variables.

```
 struct student
 {
     int prn;
     Char name[20];
     float per;
 };

int main()
{
    struct student  s1,s2,s[20];
    return 0;
}
```

```
 struct student
{
     int prn;
     Char name[20];
     float per;
}s1,s2,s[20];
```

In both cases, two variables `s1,s2,` and an array variable `s` having `20` elements of type `struct student` are created.

Accessing Members of a Structure

- There are two types of operators used for accessing members of a structure.
  - . (dot) - Member operator
  - -> Structure pointer operator

- Suppose, you want to access the percentage of s2. Here's how you can do it.

```
S2.per;
```

Why structures in C?

- Suppose, you want to store information about a student: PRN. his/her name, percentage. You can create different variables `PRN,name,per` to store this information.
- What if you need to store information of more than one student? Now, you need to create different variables for each information per student : `prn1,name1,per1, prn2,name2,per2, etc.`
- A better approach would be to have a collection of all related information under a single name `student` structure and use it for every student.