PART-3

More on structure and union

1) *typedef:* typedefined for structure and union

One way to use typedef example1:

Line 1-   struct student{
Line 2-   char sname[50];
Line 3-   ....
Line 4-   };
Line 5-   *typedef* struct student stud;
Line 6-   int main(){
Line 7-   struct student s1;   // declaration of s1 as structure variable
Line 8-   struct student s2;   // declaration of s2 as structure variable
Line 9-   stud s3,   // declaration of s3 as structure variable but stud is defined at Line5
Line 10-   stud s4;   // declaration of s4 as structure variable but stud is defined at Line5
Line 11-   }

Another way to use typedef example1:

Line 1-   *typedef* struct student{
Line 2-   char sname[50];
Line 3-   ....
Line 4-   } *stud*;
Line 5-
Line 6-   int main(){
Line 7-   struct student s1;   // declaration of s1 as structure variable
Line 8-   struct student s2;   // declaration of s2 as structure variable
Line 9-   stud s3,   // declaration of s3 as structure variable but stud is defined at Line4
Line 10-   stud s4;   // declaration of s4 as structure variable but stud is defined at Line4
Line 11-   }

2) *Structure and function calls:*

The C language allows us to pass an entire structure to a function. In addition a function can return a structure back to its caller.

Example:

Line 1-   #include<stdio.h>

Line 2-   *typedef* struct{
Line 3-   int ecode;
Line 4-   char ename[30];
Line 5-   int esal;
Line 6-   }emp;

Line 7-   void show(emp p){
Line 8-   printf("Details of Employee\n");
Line 9-   printf("Name: \t%s\n",p.ename);
Line 10-   printf("Ecode: \t%d\n",p.ecode);
Line 11-   printf("Sal: \t%d\n",p.esal);
Line 12-   }

Line 13-   int main(){
Line 14-   emp e={1001,"RGG",2000};
Line 15-   show(e);
Line 16-   return 0;
Line 17-   }

Output:
Details of Employee
Name: RGG
Ecode: 1001
Sal: 2000

Explanation:
- Between Line 3- 7, we have created a structure and its *typedef* name *"emp"*
- Between Line 8-13, created a function named show(emp p), it has a structure "emp" as parameter.
- At Line 17, we are passing structure "emp" variable "e" as parameter in function show(e).

3) *Pointer to structure:*

We can also declare a pointer to structure and it will work similar to other pointers, only the difference is that with structure pointers we will access the member through arrow operator (->).

Here I am giving you example 1:

Line 1-   #include<stdio.h>

Line 2-   typedef struct{
Line 3-   int ecode;
Line 4-   char ename[30];
Line 5-   int esal;
Line 6-   }emp;

Line 7-   int main(){
Line 8-   emp e1;
Line 9-   emp *p=&e1;
Line 10-   printf("-- Enter the record of EMPLOYEE --\n");
Line 11-   printf("Emp Code, Name, Salary:\n");
Line 12-   scanf("%d%s%d",&p->ecode,p->ename,&p->esal);
Line 13-   printf("-- Details of Employee --\n");
Line 14-   printf("ECode: \t%d\n",p->ecode);
Line 15-   printf("EName: \t%s\n",p->ename);
Line 16-   printf("Sal: \t%d\n",p->esal);
Line 17-   return 0;
Line 18-   }

Output:
-- Enter the record of EMPLOYEE --
Emp Code, Name, Salary:
1001
RGG
2000
-- Details of Employee --
Ecode: 1001
EName: RGG
Sal: 2000

Explanation:
- At Line 9, we have declared a structure pointer "p" initialized the address of structure variable "e1"
- At Line 12, 14, 15 & 16; we have accessed the members of the structure through pointer using arrow (->) operator.

Line 1- #include<stdio.h>

Line 2- *typedef* struct{
Line 3- int ecode;
Line 4- char ename[30];
Line 5- int esal;
Line 6- }emp;

Line 7- void input(emp *p){
Line 8- printf("-- Enter the record of EMPLOYEE --\n");
Line 9- printf("Emp Code, Name, Salary:\n");
Line 10- scanf("%d%s%d",&p->ecode,p->ename,&p->esal);
Line 11- }

Line 12- void show(emp *p){
Line 13- printf("-- Details of Employee --\n");
Line 14- printf("Ecode: \t%s\n",p->ename);
Line 15- printf("EName: \t%d\n",p->ecode);
Line 16- printf("Sal: \t%d\n",p->esal);
Line 17- }

Line 18- int main(){
Line 19- emp e;
Line 20- input(&e);
Line 21- show(&e);
Line 22- return 0;
Line 23- }

Output:
-- Enter the record of EMPLOYEE --
Emp Code, Name, Salary:
1001
RGG
2000
-- Details of Employee --
Ecode: RGG
EName: 1001
Sal: 2000

Explanation:
- At Line 7 & 12, structure pointer to function is used in input(....) & show(....)
- At Line 20 & 21, these functions are called by passing the address of structure variable "e"