Follow Us

C Programming Structure and Function

In C, structure can be passed to functions by two methods:

1. Passing by value (passing actual value as argument)
2. Passing by reference (passing address of an argument)

Passing structure by value

A structure variable can be passed to the function as an argument as normal variable. If structure is passed by value, change made in structure variable in function definition does not reflect in original structure variable in calling function.

Write a C program to create a structure student, containing name and roll. Ask user the name and roll of a student in main function. Pass this structure to a function and display the information in that function.

```
#include <stdio.h>
struct student{
      char name[50];
      int roll;
};
void Display(struct student stu);
/* function prototype should be below to the structure declaration otherwise compil
int main(){
      struct student s1;
      printf("Enter student's name: ");
      scanf("%s",&s1.name);
      printf("Enter roll number:");
      scanf("%d",&s1.roll);
      Display(s1);    // passing structure variable s1 as argument
      return 0;
}
void Display(struct student stu){
   printf("Output\nName: %s",stu.name);
   printf("\nRoll: %d",stu.roll);
}
```

Output

```
Enter student's name: Kevin Amla
Enter roll number: 149
Output
Name: Kevin Amla
Roll: 149
```

Passing structure by reference

The address location of structure variable is passed to function while passing it by reference. If structure is passed by reference, change made in structure variable in function definition reflects in original structure variable in the calling function.

Write a C program to add two distances(feet-inch system) entered by user. To solve this program, make a structure. Pass two structure variable (containing distance in feet and inch) to add function by reference and display the result in main function without returning it.

```
#include <stdio.h>
struct distance{
      int feet;
      float inch;
};
void Add(struct distance d1,struct distance d2, struct distance *d3);
int main()
{
      struct distance dist1, dist2, dist3;
      printf("First distance\n");
      printf("Enter feet: ");
      scanf("%d",&dist1.feet);
      scanf("%f",&dist1.inch);
      printf("Second distance\n");
      printf("Enter feet: ");
      scanf("%d",&dist2.feet);
      printf("Enter inch: ");
      scanf("%f",&dist2.inch);
      Add(dist1, dist2, &dist3);

/*passing structure variables dist1 and dist2 by value whereas passing structure va
      printf("\nSum of distances = %d\'-%.1f\"",dist3.feet, dist3.inch);
      return 0;
}
void Add(struct distance d1,struct distance d2, struct distance *d3)
{
/* Adding distances d1 and d2 and storing it in d3 */
       d3->feet=d1.feet+d2.feet;
       d3->inch=d1.inch+d2.inch;
       if (d3->inch>=12) {      /* if inch is greater or equal to 12, converting i
             d3->inch-=12;
             ++d3->feet;
      }
}
```

Output

```
First distance
Enter feet: 12
Enter inch: 6.8
Second distance
Enter feet: 5
Enter inch: 7.5

Sum of distances = 18'-2.3"
```

Explaination

In this program, structure variables dist1 and dist2 are passed by value (because value of dist1 and dist2 does not need to be displayed in main function) and dist3 is passed by reference ,i.e, address of dist3 (&dist3) is passed as an argument. Thus, the structure pointer variable d3 points to the address of dist3. If any change is made in d3 variable, effect of it is seed in dist3 variable in main function.

« Structures & Pointers   C Unions »