# Passing Structure to a function

In C, [structure](https://www.programiz.com/c-programming/c-structures) can be passed to functions by two methods:

1. [Passing by value (passing actual value as argument)](https://www.programiz.com/c-programming/c-user-defined-functions)
2. [Passing by reference (passing address of an argument)](https://www.programiz.com/c-programming/c-pointer-functions)

**Passing structure by value**

A structure variable can be passed to the function as an argument as a normal variable.

If structure is passed by value, changes made to the structure variable inside the function definition does not reflect in the originally passed structure variable.

**C program to create a structure student, containing name and roll and display the information.**

#include <stdio.h>

struct student

{

char name[50];

int roll;

};

void display(struct student stu);

int main()

{

struct student stud;

printf("Enter student's name: ");

scanf("%s", &stud.name);

printf("Enter roll number:");

scanf("%d", &stud.roll);

display(stud);

return 0;

}

void display(struct student stu){

printf("Output\nName: %s",stu.name);

printf("\nRoll: %d",stu.roll);

}

**Output**

Enter student's name: Nisha

Enter roll number: 23

Output

Name: Nisha

Roll: 23

### Passing structure by reference

The memory address of a structure variable is passed to function while passing it by reference.

If structure is passed by reference, changes made to the structure variable inside function definition reflects in the originally passed structure variable.

**Example**

#include <stdio.h>

#include <string.h>

struct student

{

           int id;

           char name[20];

};

void display(struct student \*stu);

int main()

{

          struct student stud;

printf("Enter student's name: ");

scanf("%s", &stud.name);

printf("Enter id:");

scanf("%d", &stud.id);

          display(&stud);

          return 0;

}

void display(struct student \*stu)

{

printf("Output\nName: %s",stu.name);

printf("\nRoll: %d",stu.id);

}