**Syntax:**  
struct tag\_name  
{  
data type var\_name1;  
data type var\_name2;  
data type var\_name3;  
};

**Example:**  
struct student  
{  
int  mark;  
char name[10];  
float average;  
};

**Declaring structure using normal variable:**  
struct student report;

**Initializing structure using normal variable:**  
struct student report = {100, “Mani”, 99.5};

**Accessing structure members using normal variable:**  
report.mark;  
report.name;  
report.average;

#include <stdio.h>

struct student

{

           int id;

           char name[20];

           float percentage;

};

int main()

{

           struct student record;

           record.id=1;

           record.name= "Raju";

           record.percentage = 86.5;

           printf(" Id is: %d \n", record.id);

           printf(" Name is: %s \n", record.name);

           printf(" Percentage is: %f \n", record.percentage);

           return 0;

}

**Example program for C structure using pointer:**

In this program, “record1” is normal structure variable and “ptr” is pointer structure variable. As you know, Dot(.) operator is used to access the data using normal structure variable and arrow(->) is used to access data using pointer variable.

#include <stdio.h>

struct student

{

     int id;

     char name[30];

     float percentage;

};

int main()

{

     struct student record1 = {1, "Raju", 90.5};

     struct student \*ptr;

     ptr = &record1;

         printf("Records of STUDENT1: \n");

         printf("  Id is: %d \n", ptr->id);

         printf("  Name is: %s \n", ptr->name);

         printf("  Percentage is: %f \n\n", ptr->percentage);

     return 0;

}

**Another way to do structure**

#include <stdio.h>

struct student

{

            int id;

            char name[20];

            float percentage;

} record;

int main()

{

            record.id=1;

            record.name="Raju";

            record.percentage = 86.5;

            printf(" Id is: %d \n", record.id);

            printf(" Name is: %s \n", record.name);

            printf(" Percentage is: %f \n", record.percentage);

            return 0;

}

**How to declare structure:**

struct person

{

char name[50];

int citNo;

float salary;

};

int main()

{

struct person person1, person2, person3[20];

return 0;

}

struct person

{

char name[50];

int citNo;

float salary;

} person1, person2, person3[20];

#include <stdio.h>

struct Distance

{

int feet;

float inch;

} dist1, dist2, sum;

int main()

{

printf("1st distance\n");

// Input of feet for structure variable dist1

printf("Enter feet: ");

scanf("%d", &dist1.feet);

// Input of inch for structure variable dist1

printf("Enter inch: ");

scanf("%f", &dist1.inch);

printf("2nd distance\n");

// Input of feet for structure variable dist2

printf("Enter feet: ");

scanf("%d", &dist2.feet);

// Input of feet for structure variable dist2

printf("Enter inch: ");

scanf("%f", &dist2.inch);

sum.feet = dist1.feet + dist2.feet;

sum.inch = dist1.inch + dist2.inch;

if (sum.inch > 12)

{

//If inch is greater than 12, changing it to feet.

++sum.feet;

sum.inch = sum.inch - 12;

}

// printing sum of distance dist1 and dist2

printf("Sum of distances = %d\'-%.1f\"", sum.feet, sum.inch);

return 0;

}

**Output**

1st distance

Enter feet: 12

Enter inch: 7.9

2nd distance

Enter feet: 2

Enter inch: 9.8

Sum of distances = 15'-5.7"