## Structure Practice Programs

```
#include<stdio.h>
struct Car {
  char brand[50];
  char model[50];
  int year;
};
int main() {
  struct Car car1 = {"BMW", "X5", 1999};
  struct Car car2 = {"Ford", "Mustang", 1969};
  struct Car car3 = {"Toyota", "Corolla", 2011};
  printf("%s %s %d\n", car1.brand, car1.model, car1.year);
  printf("%s %s %d\n", car2.brand, car2.model, car2.year);
  printf("%s %s %d\n", car3.brand, car3.model, car3.year);
  return 0;
}
```

### Wrong way of Arithmatic

```
#include<stdio.h>
struct number
{
        float x;
};
int main()
{
        struct number n1,n2,n3;
        n1.x=4;
        n2.x=3;
        n3=n1+n2;
        return 0;
}
```

### Right Way Of Arithematic

```
#include <stdio.h>
struct number {
        float x;
};
int main()
{
        struct number n1,n2,n3;
        n1.x=4;
        n2.x=3;
```

```
        n3.x=(n1.x)+(n2.x);
        printf("\n%f",n3.x);
        return 0;
}
```

## Array of structures

```
#include<stdio.h>
struct Point
{
  int x, y;
};

int main()
{
  // Create an array of structures
  struct Point arr[10];
  // Access array members
  arr[0].x = 10;
  arr[0].y = 20;

  printf("%d %d", arr[0].x, arr[0].y);
  return 0;
}
```

```
        #include<stdio.h>
        #include <string.h>
        struct student{
        int rollno;
        char name[10];
        };
        int main(){
        int i;
        struct student st[5];
        printf("Enter Records of 5 students");
        for(i=0;i<5;i++){
        printf("\nEnter Rollno:");
        scanf("%d",&st[i].rollno);
        printf("\nEnter Name:");
        scanf("%s",&st[i].name);
        }
        printf("\nStudent Information List:");
        for(i=0;i<5;i++){
        printf("\nRollno:%d, Name:%s",st[i].rollno,st[i].name);
```

```
}
return 0;
}
```

## Passing Structure to function

```
#include <stdio.h>
#include <string.h>
struct Books {
   char  title[50];
   char  author[50];
   char  subject[100];
   int   book_id;
};
/* function declaration */
void printBook( struct Books book );
int main( ) {
   struct Books Book1;        /* Declare Book1 of type Book */
   struct Books Book2;        /* Declare Book2 of type Book */
   /* book 1 specification */
   strcpy( Book1.title, "C Programming");
   strcpy( Book1.author, "Nuha Ali");
   strcpy( Book1.subject, "C Programming Tutorial");
   Book1.book_id = 6495407;
   /* book 2 specification */
   strcpy( Book2.title, "Telecom Billing");
   strcpy( Book2.author, "Zara Ali");
   strcpy( Book2.subject, "Telecom Billing Tutorial");
   Book2.book_id = 6495700;
   /* print Book1 info */
   printBook( Book1 );
   /* Print Book2 info */
   printBook( Book2 );
   return 0;
}
void printBook( struct Books book ) {
   printf( "Book title : %s\n", book.title);
   printf( "Book author : %s\n", book.author);
   printf( "Book subject : %s\n", book.subject);
   printf( "Book book_id : %d\n", book.book_id);
}
```

## The structure can be nested in the following ways.

### 1)By separate structure

```
struct Date
{
  int dd;
  int mm;
  int yyyy;
};
struct Employee
{
  int id;
  char name[20];
  struct Date doj;
}emp1;
```

### 2)By Embedded structure

```
struct Employee
{
int id;
char name[20];
struct Date
{
int dd;
int mm;
int yyyy;
}doj;
}emp1;
```

## Nested Structure Example

```
#include<stdio.h>
struct address
{
char city[20];
int pin;
char phone[14];
};
struct employee
{
char name[20];
struct address add;
```

```
};
void main ()
{
struct employee emp;
printf("Enter employee information?\n");
scanf("%s %s %d %s",emp.name,emp.add.city, &emp.add.pin, emp.add.phone);
printf("Printing the employee information....\n");
printf("name: %s\nCity: %s\nPincode: %d\nPhone:
%s",emp.name,emp.add.city,emp.add.pin,emp.add.phone);
}
```

```
#include <stdio.h>
#include <string.h>
struct Employee
{
int id;
char name[20];
struct Date
{
int dd;
int mm;
int yyyy;
}doj;
}e1;
int main( )
{
//storing employee information
e1.id=101;
strcpy(e1.name, "Sonoo Jaiswal");//copying string into char array
e1.doj.dd=10;
e1.doj.mm=11;
e1.doj.yyyy=2014;
//printing first employee information
printf( "employee id : %d\n", e1.id);
printf( "employee name : %s\n", e1.name);
printf( "employee date of joining (dd/mm/yyyy) : %d/%d/%d\n",
e1.doj.dd,e1.doj.mm,e1.doj.yyyy);
return 0;
}
```

## Access Members of a Structure

**There are two types of operators used for accessing members of a structure.**

**. - Member operator**
**-> - Structure pointer operator (will be discussed in the next tutorial)**

## Pointer to structure

```
#include<stdio.h>
struct Point
{
  int x, y;
};

int main()
{
  struct Point p1 = {1, 2};
  // p2 is a pointer to structure p1
  struct Point *p2 = &p1;
  // Accessing structure members using structure pointer
  printf("%d %d", p2->x, p2->y);
  return 0;
}
```

## Structure pointer

```
#include <stdio.h>
#include <string.h>
struct Books {
  char  title[50];
  char  author[50];
  char  subject[100];
  int   book_id;
};

/* function declaration */
void printBook( struct Books *book );
int main( ) {
  struct Books Book1;          /* Declare Book1 of type Book */
  struct Books Book2;          /* Declare Book2 of type Book */
  /* book 1 specification */
  strcpy( Book1.title, "C Programming");
```

```
   strcpy( Book1.author, "Nuha Ali");
   strcpy( Book1.subject, "C Programming Tutorial");
   Book1.book_id = 6495407;
   /* book 2 specification */
   strcpy( Book2.title, "Telecom Billing");
   strcpy( Book2.author, "Zara Ali");
   strcpy( Book2.subject, "Telecom Billing Tutorial");
   Book2.book_id = 6495700;
   /* print Book1 info by passing address of Book1 */
   printBook( &Book1 );
   /* print Book2 info by passing address of Book2 */
   printBook( &Book2 );
   return 0;
}
void printBook( struct Books *book ) {
   printf( "Book title : %s\n", book->title);
   printf( "Book author : %s\n", book->author);
   printf( "Book subject : %s\n", book->subject);
   printf( "Book book_id : %d\n", book->book_id);
}
```