**Accessing Structure Members**

#include <stdio.h>

#include <string.h>

struct Books {

char title[50];

char author[50];

char subject[100];

int book\_id;

}Book1,Book2;

int main( ) {

// struct Books Book1; /\* Declare Book1 of type Books \*/

//struct Books Book2; /\* Declare Book2 of type Books \*/

printf("\n size of the structure is %d bytes",sizeof(Book1));

/\* book 1 specification \*/

strcpy( Book1.title, "C Programming");

strcpy( Book1.author, "Yashbant Kanitkar");

strcpy( Book1.subject, "C Programming Tutorial");

Book1.book\_id = 6495407;

printf("\n enter books title");

scanf("%s",Book2.title);

printf("\n enter books author");

scanf("%s",Book2.author);

printf("\n enter books subject");

scanf("%s",Book2.subject);

printf("\n enter books id");

scanf("%d",&Book2.book\_id);

/\* print Book1 info \*/

printf( "Book 1 title : %s\n", Book1.title);

printf( "Book 1 author : %s\n", Book1.author);

printf( "Book 1 subject : %s\n", Book1.subject);

printf( "Book 1 book\_id : %d\n", Book1.book\_id);

/\* print Book2 info \*/

printf( "Book 2 title : %s\n", Book2.title);

printf( "Book 2 author : %s\n", Book2.author);

printf( "Book 2 subject : %s\n", Book2.subject);

printf( "Book 2 book\_id : %d\n", Book2.book\_id);

return 0;

}

**Structures as Function Arguments**

#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, "Yashbant Kanitkar");

strcpy( Book1.subject, "C Programming Tutorial");

Book1.book\_id = 6495407;

/\* book 2 specification \*/

strcpy( Book2.title, "Basic Electronics");

strcpy( Book2.author, "M.Tailor");

strcpy( Book2.subject, "Electronics");

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);

}

Determine size of structure without using sizeof function

#include <stdio.h>

#include <string.h>

struct Books {

char title[50];

char author[50];

char subject[100];

int book\_id;

};

int main()

{

struct Books \*p=0;

p++;

printf("\n size of the structure is %d",p);

return 0;

}

**Union 1**

#include <stdio.h>

#include <string.h>

union book {

int id;

//double price;

char name;

};

int main( ) {

union book b1;

printf( "Memory size occupied by data : %d\n", sizeof(b1));

return 0;

}

**Union 2**

#include <stdio.h>

#include <string.h>

union book {

short id;

float price;

char name[20];

};

int main( ) {

union book b1={10};

// b1.id = 10;

printf( "b1.id : %d\n", b1.id);

strcpy( b1.name, "C Programming");

//b1.price = 220.5;

printf( "b1.price : %f\n", b1.price);

printf( "b1.name : %s\n", b1.name);

return 0;

}