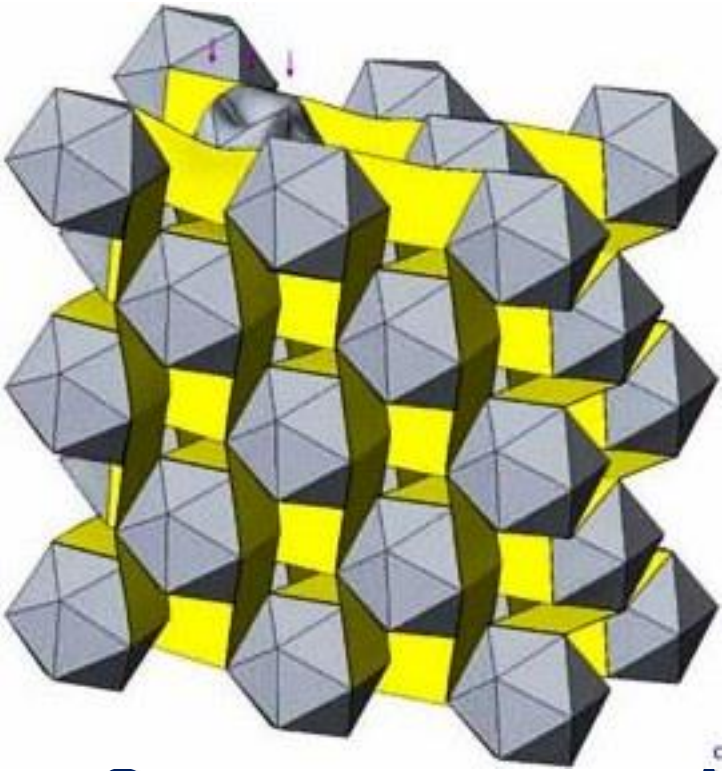
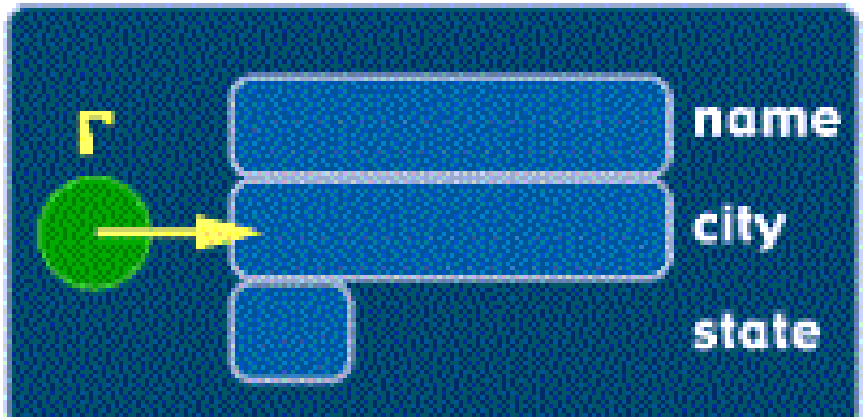




Structures within Structures, Structures and functions, Array of structures

Objectives

- To learn and appreciate the following concept
  - Array of structures

Session outcome

- At the end of session one will be able to
  - Understand the overall ideology of array of structures
  - Write programs using array of structures

Structures within Structures

Structure within structure means nesting of structures.

for instance see the following structure defined to store information about students

```
struct student{
   int rollno;
   char name[15];
   struct { // marks for 3 subjects under structure marks
        int sub1;
        int sub2;
        int sub3;
    }marks;
}fs[3]; //3 students
```

Structures within Structures

```
//Structure Definition
 struct student{
    int rollno;
    char name[15];
    struct m marks;
  }fs[3];
```

```
//Structure Definition
struct m{
   int sub1;
   int sub2;
   int sub3;
    };
```

Tag name is used to define inner structure **marks**

The members contained in the inner structure namely **sub1, sub2** and **sub3** can be referred to as:

```
fs[i].marks.sub1;
fs[i].marks.sub2;
fs[i].marks.sub3;
```

Structures and functions

```
void read(struct book x[]); // prototype
int main() {
int i;
struct book b1[2];
printf("\n Enter IBN, Author name & Price \n");
  read(b1); // function call

printf("\nThe book details entered are:\n");
for(i=0;i<2;i++){
  printf("\n Book %d", i+1);
  printf("\nIBN: \t\t%d", b1[i].ibn);
  printf("\nAuthor: \t%s", b1[i].author);
  printf("\nPrice: \t\t%f", b1[i].price);
 }
return 0;
}
```

```
//Structure Definition
struct book
{
  int ibn;
  char author[15];
  float price;
};
```

```
//function definition
void read(struct book a[])
{
 int i;
 for(i=0;i<2;i++){
   printf("\nBook %d\n", i+1);
   scanf("%d", &a[i].ibn);
   scanf("%s", a[i].author);
   scanf("%f", &a[i].price);
  }
}
```

Structures -Problems

Write programs to

1. Create a student record with name, rollno, marks of 3 subjects (m1, m2, m3). Compute the average of marks for 3 students and display the names of the students in ascending order of their average marks.
2. Create an employee record with emp-no, name, age, date-of-joining (year), and salary. If there is 20% hike on salary per annum, compute the retirement year of each employee and the salary at that time. [standard age of retirement is 55]

Structures – Solution for Q1

```
int main()
{
 struct student temp, fs[3] =
                        {{1,"manish",45,47,49},
                         {2,"ankur",43,44,45},
                         {3,"swati",46,42,43}};
  int i, n=3, total[3]={0}, avg[3]={0},tot=0;

  for(i=0; i< n; i++)   {
     total[i]=fs[i].marks.sub1+fs[i].marks.sub2+
     fs[i].marks.sub3; //students total

   avg[i] = total[i]/3;
 }
//display
printf("Total & Average of each student.\n“);
for(i=0;i<n;i++){
 printf("\nTotal of %s = %d & avg = %d", fs[i].name, total[i], avg[i]);
}
```

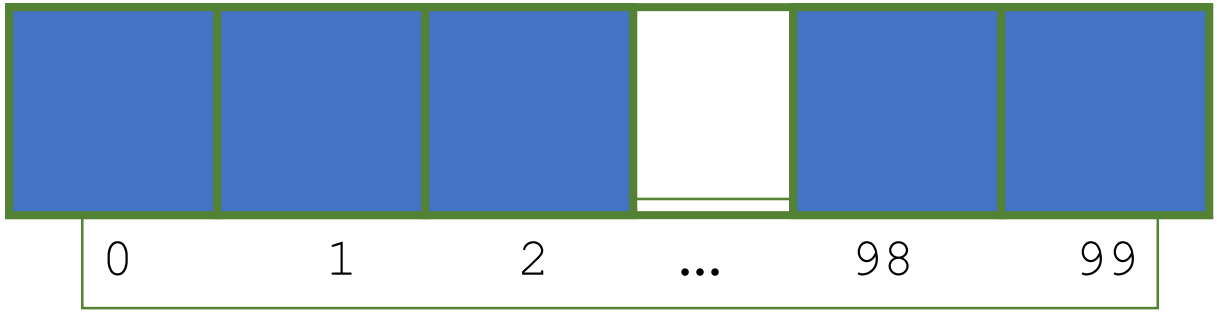
```
struct student{
int rollno;
char name[15];
    struct {
            int sub1;
            int sub2;
            int sub3;
            }marks;
};
```

Structures – Solution for Q1

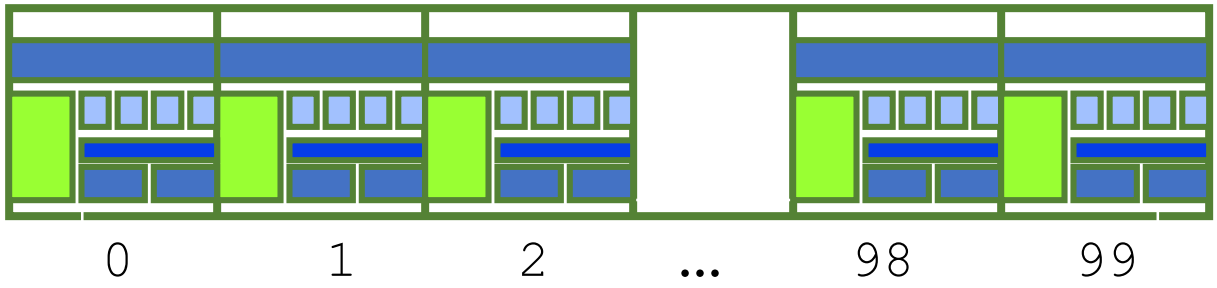
```
// sorting
for(i=0;i<n;i++)
  for(int j=i+1;j<n;j++)
  if(avg[i] > avg[j])
   {
     temp=fs[i]; //Swapping
     fs[i]=fs[j];
     fs[j]=temp;
    }
 for(i=0;i<n;i++) //Sorted list w.r.to average marks
  printf("\n%s\n",fs[i].name);
return 0;
} //end of main
```

Arrays of structures

- An ordinary array: One type of data



- An array of structs: Multiple types of data in each array element.



Array of structures

We can define single or multidimensional arrays as structure variables.

```
struct marks
  {
      int  subject1;
      int  subject2;
      int  subject3;
  } ;
marks student[80];
```

- Defines an array called student, that consists of 80 elements.

- Each element is defined to be the type marks.

Structures and functions

```
void read(struct book x[]); // prototype
int main() {
int i;
struct book b1[2];
printf("\n Enter IBN, Author name & Price \n");
  read(b1); // function call

printf("\nThe book details entered are:\n");
for(i=0;i<2;i++){
  printf("\n Book %d", i+1);
  printf("\nIBN: \t\t%d", b1[i].ibn);
  printf("\nAuthor: \t%s", b1[i].author);
  printf("\nPrice: \t\t%f", b1[i].price);
 }
return 0;
}
```

```
//Structure Definition
struct book
{
  int ibn;
  char author[15];
  float price;
};
```

```
//function definition
void read(struct book a[])
{
 int i;
 for(i=0;i<2;i++){
   printf("\nBook %d\n", i+1);
   scanf("%d", &a[i].ibn);
   scanf("%s", a[i].author);
   scanf("%f", &a[i].price);
  }
}
```

Structures as Function Arguments

You can pass a structure as a function argument in very similar way as you pass any other variable or 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;
```

Structures as Function Arguments

```
/* 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);
 }
```

Structures -Problem

Write a menu driven program for a "***BOOK MART***" with the following menu options

BOOKMART MENU

1. ***Availability***
2. ***Purchase***
3. ***Exit***

The details of the books are stored in a structure "***books***" with the member variables ***book_number, book_name, book_price, book_author, number_of_copies.***

Declare the different member variables (use meaningful abbreviations for the variables; e.g. **bno** for *book number*, **noc** for *number of copies* etc.) with appropriate data types. Use an array of structure ***book[ ]*** to insert details for at least 5 books. Your program shall run continuously for all the operations until you press ***Exit*** option in the menu. Purchase menu should be used to purchase a particular book using the book number as user input. [Hint: usage of SWITCH within WHILE statement (repeating loop)]

Summary

- Array of Structures
- Structures within Structures
- Structures and Functions