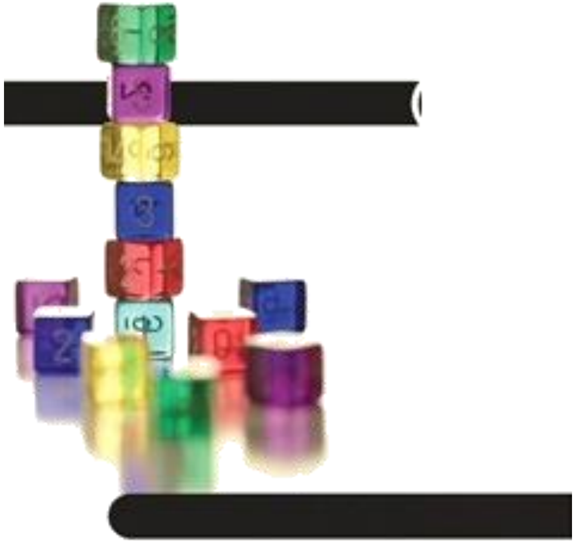


S21_2 Structures

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 Array of Structures
  - Understand the overall ideology of array of structures
  - Write programs using array of structures

Structure Initialization Methods

```
int main ( )
{
    struct Student
    {
       int rollno;
       int age;
    };
    Student s1={20, 21};
    Student s2={21, 21};
}
```

```
struct Student
 {
    int rollno;
    int age;
 } s1={20, 21};

 main ( )
 {
      Student s2={21, 21};
      …
      …
  }
```

Structure: Example

```
struct Book {                // definition
  char title[20];
  char author[15];
  int pages;
  float price;
};
int main( ){
    struct Book b1;
    printf(“Input values”);
    scanf(“%s %s %d %f”, b1.title, b1.author, &b1.pages,
              &b1.price);
    //output
   printf(“%s %s %d %f”, b1.title, b1.author, b1.pages,
              b1.price);
    return 0;
}
```

Structure: Example

```
struct Book {                    // definition
  char title[20];
  char author[15];
  int pages;
  float price;
};
int main( ){
    struct Book b1;
    printf(“Input values”);
    gets(b1.title); gets(b1.author);
    scanf(“%d %f”, &b1.pages,  &b1.price);
    //output
   printf(“%s %s %d %f”, b1.title, b1.author, b1.pages,
              b1.price);
    return 0;
}
```

Structures: overview

- Definition & structure variable declaration

```
struct student
  {   int rollno;
      int age;
      char name[20];
      }s1, s2, s3;
```

- Initialization

```
 int main( ){
 struct
  { int rollno;
       int age;
       }stud={20, 21};
       …
       …
       return 0;
}
```

- Giving values to members

Using dot operator '.'

```
s1. rollno = 25;
printf("%s",s1.name);
```

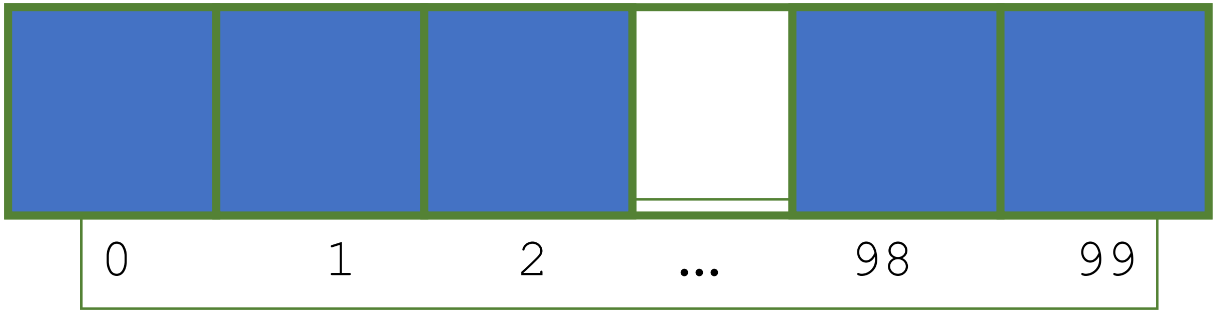
'.' operator acts as Link between member and a Structure variable.

- Assign & compare members

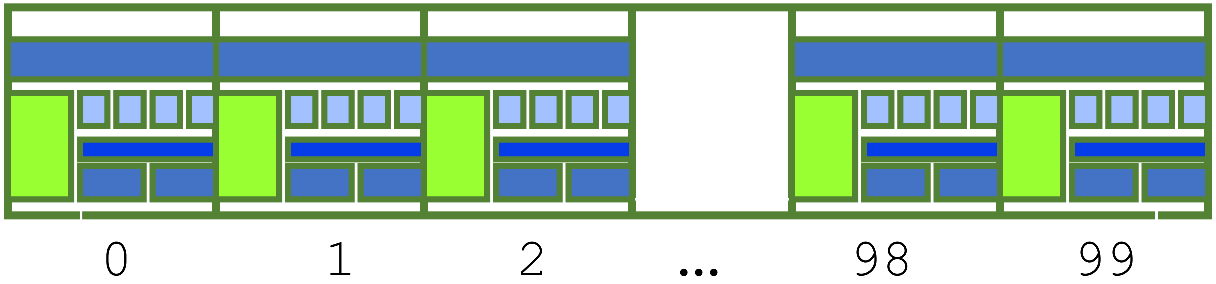
```
s1 = s2 ;    assignment  (allowed)
---------------------------
s1 == s2     comparison (not allowed)
s1!=s2         comparison (not allowed)
---------------------------
s1.rollno == s2.rollno;   (allowed)
s1.rollno!=s2.rollno;     (allowed)
```

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.

Array of structures – Initialization

```
struct marks {
  int  subject1;
  int  subject2;
  int  subject3;
}  ;

main(){
  marks student[]={
                  {45,47,49},
                  {43,44,45},
                  {46,42,43}
                };
```

	Memory
student[0].subject1	45
student[0].subject2	47
student[0].subject3	49
student[1].subject1	43
student[1].subject2	44
student[1].subject3	45
student[2].subject1	46
student[2].subject2	42
student[2].subject3	43

Array of Structure: Example

```
struct Book {                    //Structure Definition
        char title[20];
        char author[15];
        int pages;
        float price;
};
int main( ){
   struct Book b[10];
   printf("Input values");
   for (int i=0;i<3;i++)
   scanf("%s   %s   %d   %f",   b[i].title,   b[i].author,   &b[i].pages,
&b[i].price);
   for (int j=0;j<3;j++)
   printf("%s\t %s\t %d\t %f\n", b[j].title, b[j].author, b[j].pages,
b[j].price);
 return 0;
}
```

Arrays within Structures

We can define single or multidimensional arrays inside a structure.

```
struct marks
{      int  rollno;
       float  subject[3];
} student[2] ;
```

The member **subject** contains 3 elements; **subject[0], subject[1] & subject[2].**

student[1].subject[2];

- **Refers to the marks obtained in the third subject by the second student.**

Arrays within structures : example

```
#include<stdio.h>
 int main(){
  struct marks student[3] ={{0,45,47,49},
                            {0,43,44,45},
                            {0,46,42,43}};
  int i, j ;
  //students total
  for(i=0;i<=2;i++)  {
   for(j=0;j<=2;j++)
     student[i].total+=student[i].sub[j]; }
  printf("Grand Total of each student:");

for(i=0;i<=2;i++)
  printf("\nTotal of student[%d]= %d", i, student[i].total);
 return 0;
}
```

```
//Structure Definition
struct marks{
   int total;
   int sub[3];
};
```

Summary

- Simple problems using structures
- Array of Structures
- Arrays within Structures
- Structures within Structures