

STRUCTURES IN C



INRODUCTION

2

- ❑ Array's disadvantage is that all elements stored in array must be of same type.
- ❑ If we want collection of elements of different types then it is not possible with array.
- ❑ Structure can be used to store elements of different types.

WHAT IS STRUCTURE

3

- ❑ Structure is a user defined data type.
- ❑ A structure is a collection of variables under a single name.
- ❑ These variables can be of different types, and each has a name that is used to select it from the structure.
- ❑ The memory required to store structure is total of memory required to store each individual element.
- ❑ The proper place for structure declarations is in the global area of the program before main().
- ❑ It is not possible to compare structures for equality using '==', nor is it possible to perform arithmetic on structures.

DECLARING STRUCTURES

4

- A structure is declared by using the keyword `struct` followed by an optional structure tag followed by the body of the structure.

General Format:

```
struct <structure tag_name> {  
  
    <data _type member _ame1>  
    <data _type member _ame1>  
    .  
    .  
};
```

Example:

```
struct myStruct {  
  
    int a;  
    int b;  
    float c;  
  
};
```

- Structure tag name is name of the structure.

Different Ways Of Declaring Structures

```
struct myStruct {  
    int a;  
    int b;  
    float c;  
};
```

struct myStruct s1, s2
both contains a b,
and c value

```
struct {  
    int a;  
    int b;  
    float c;  
} s1,s2;
```

s1, s2 both
have the defined
structure, containing
a ,b, and c ,but
do not have a tag

```
struct myStruct {  
    int a;  
    int b;  
    float c;  
}s1, s2;
```

same as the other
two versions, but
united into one set
of code.

For the first and last sets of code, myStruct is a defined tag and can be used later, but in the middle code, there is no tag, so there is no way to reference more examples of this structure

ACCESSING THE MEMBERS OF STRUCTURE

6

- The members of a structure can be accessed by using the ‘.’, which is known as the ‘dot operator’.
- The general form of the statement for accessing a member of a structure is as follows:

< structure_variable > . < member_name > ;

Example:

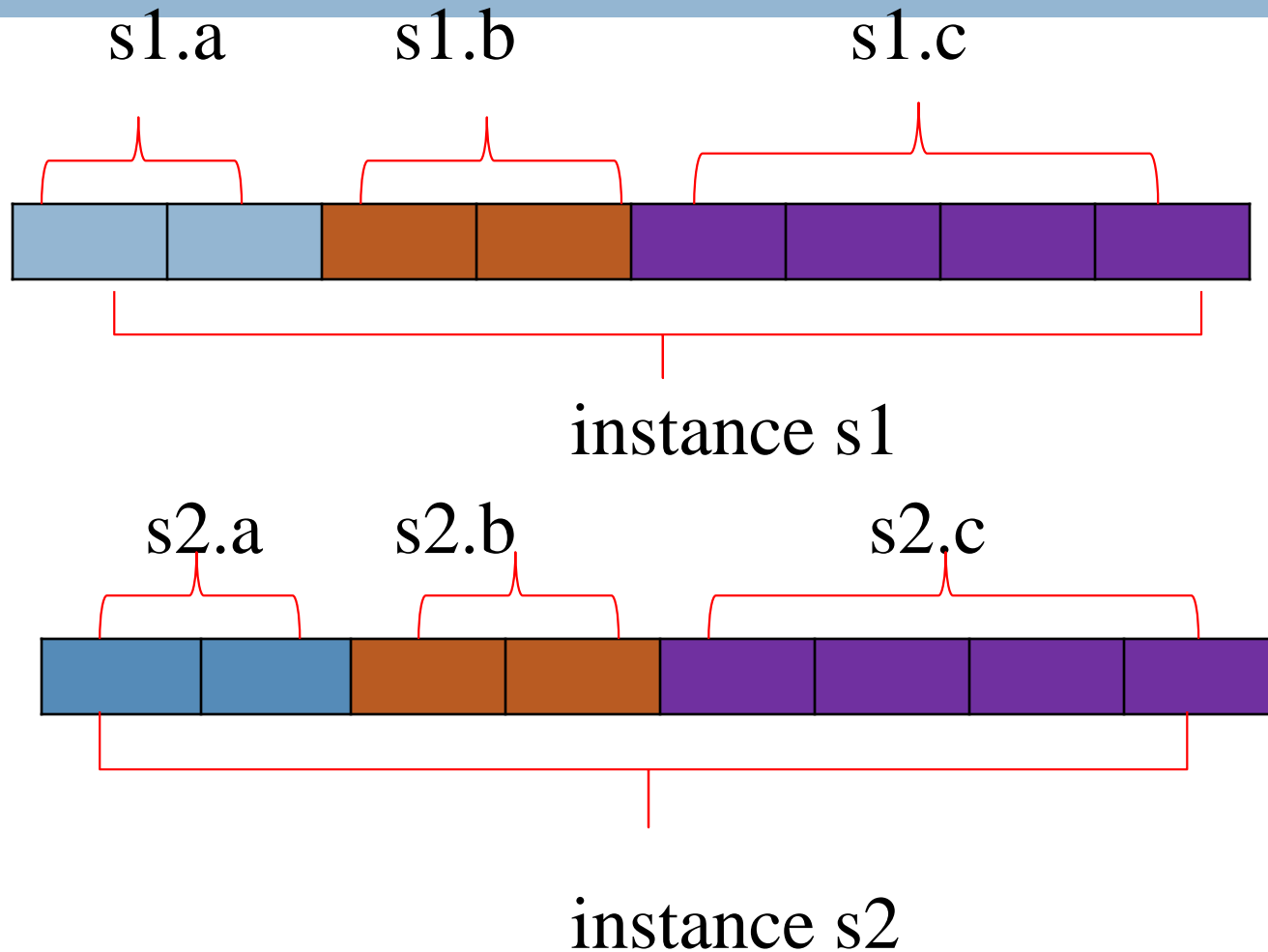
```
struct myStruct {  
    int a;  
    int b;  
    float c;  
  
}s1,s2;
```

a b and c can be accessed as :

```
s1.a  
s1.b  
s1.c
```

Conceptual View of memory allocated for instances s1 and s2

7



Initialization of Structures

8

- Structures can be initialized by one of the following ways :

```
struct myStruct {  
    int a;  
    int b;  
    float c;  
}s1={ 10,20,2.3};
```

OR

```
struct myStruct s1={ 10,20,2.3};
```


Demonstration Of Input And Output

9

```
#include<stdio.h>
#include<conio.h>
struct myStruct{
    int a;
    int b;
    float c;
}s1;
void main()
{
    clrscr();
    printf("Enter values for a , b and c");
    scanf("%d%d%f",&s1.a,&s1.b,&s1.c);
    printf("\na=%d",s1.a);
    printf("\nb=%d",s1.b);
    printf("\nc=%f",s1.c);
    getch();
}
```

OUTPUT

Enter values for a , b and c

2

3

4.5

a=2

b=3

c=4.500000

Define a structure named 'Person' which will store id, name and address . Write a program to display all these details.

10

```
#include<stdio.h>
#include<conio.h>
struct Person{
    int id;
    char name[20];
    char address[30];
}p1;

void main()
{
    clrscr();
    printf("Enter ID :\n");
    scanf("%d",&p1.id);
    printf("Enter Name :\n");
    scanf("%s",&p1.name);
    printf("Enter Address :\n");
    scanf("%s",&p1.address);
```

```
printf("\n PERSON INFORMATION
:");
printf("\nID = %d",p1.id);
printf("\nNAME = %s",p1.name);
printf("\nADDRESS =
%s",p1.address);
getch();
}
```

OUTPUT

11

Enter ID : 1

Enter Name : Riyan

Enter Address : Airoli

PERSON INFORMATION :

ID = 1

NAME = Riyan

ADDRESS = Airoli

Nested Structures

12

- A structure can be placed within another structure.
- In such cases, the dot operator in conjunction with the structure variables are used to access the members of the innermost as well as the outermost structures.

Example

```
struct First{  
    int a;  
    struct Second{  
        float b;  
    }s1;  
}f1;
```

a and b can be accessed as :

f1.a

f1.s1.b

Define a structure named 'Person' which will store id, name ,address and date of birth as a collection of day, month and year . Write a program to display all these details.

13

```
#include<stdio.h>
#include<conio.h>

struct Person{
    int id;
    char name[20];
    char address[30];
    struct DOB{
        int day;
        int month;
        int year;
    }d1;
}p1;
```

```
void main()
{
clrscr();
printf("Enter ID :\n");
scanf("%d",&p1.id);
printf("Enter Name :\n");
scanf("%s",&p1.name);
printf("Enter Address :\n");
scanf("%s",&p1.address);
printf("Enter Date of Birth in dd/mm/yyyy form :\n");
scanf("%d%d%d",&p1.d1.day,&p1.d1.month,&p1.d1.year);
printf("\n PERSON INFORMATION :");
printf("\nID = %d",p1.id);
printf("\nNAME = %s",p1.name);
printf("\nADDRESS = %s",p1.address);
printf("\nDATE OF BIRTH = %d:%d:%d",p1.d1.day,p1.d1.month,p1.d1.year);
getch();
}
```

OUTPUT

Enter ID : 1

Enter Name : Riyan

Enter Address : Airoli

Enter Date of Birth in dd/mm/yyyy form : 31

5

2012

PERSON INFORMATION :

ID = 1

NAME = Riyan

ADDRESS = Airoli

DATE OF BIRTH = 31:5:2012

Arrays of Structures

16

- The structure variable would be an array of objects, each of which contains the member elements declared within the structure construct.

Example:

```
struct Person{  
    int id;  
    char name[20];  
    char address[30];  
}p1[5];
```

Or

```
struct Person p1[5];
```


Example: Arrays of Structures

17

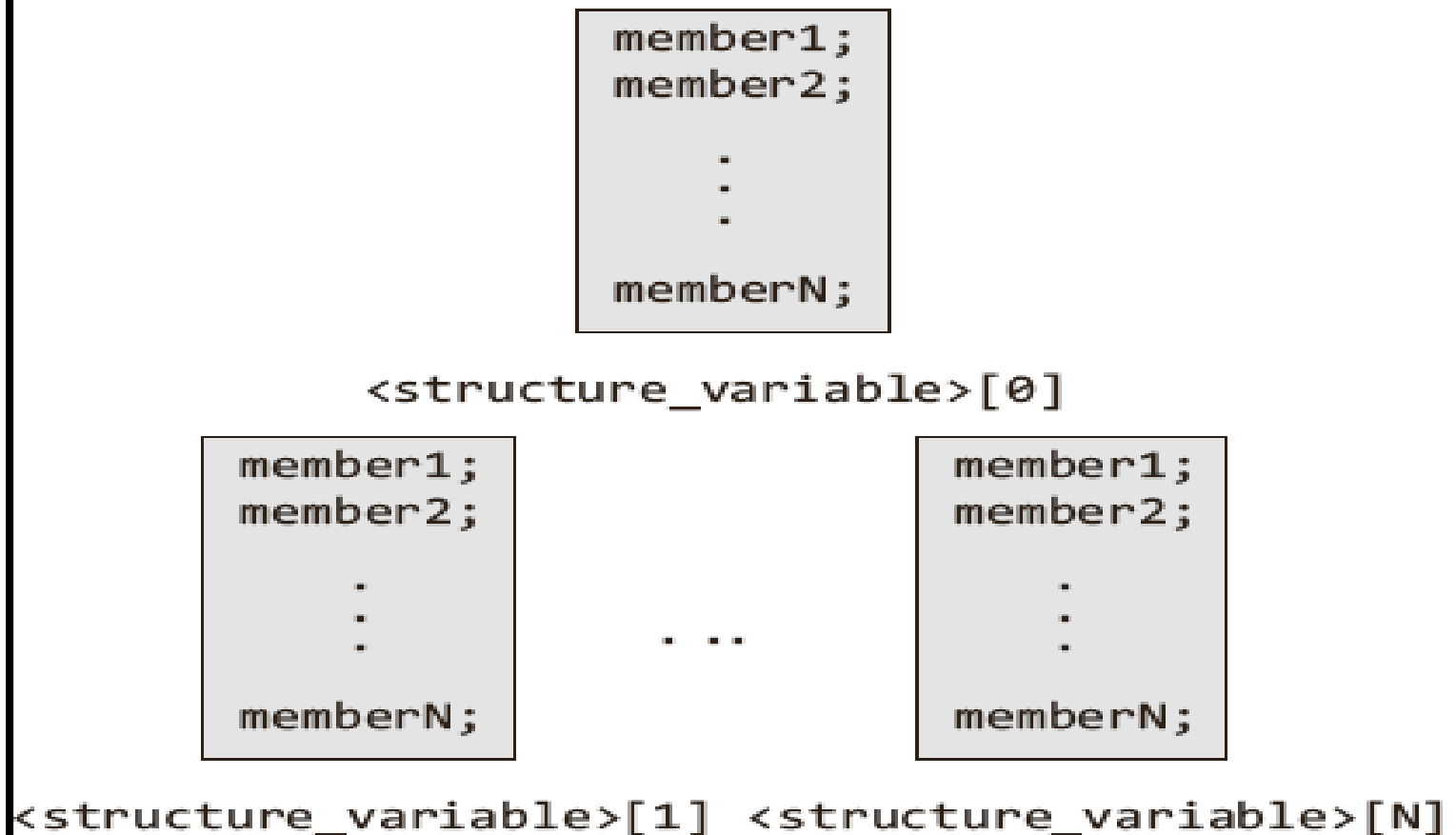


Figure Array of structures

Define a structure named 'Person' which will store id, name and address . Write a program to display all these details for 5 persons.

18

```
#include<stdio.h>
#include<conio.h>
struct Person{
    int id;
    char name[20];
    char address[30];
}p1[5];
void main()
{
    int i;
    clrscr();
    for(i=0;i<5;i++){
        printf("Enter Details for Person %d\n",i+1);
        printf("Enter ID : \n");
        scanf("%d",&p1[i].id);
        printf("Enter Name : \n");
        scanf("%s",&p1[i].name);
        printf("Enter Address : \n");
        scanf("%s",&p1[i].address);
    }
```

```
printf("\n PERSON INFORMATION : \n");
for(i=0;i<5;i++)
{
    printf("\nPerson %d\n",i+1);
    printf("\nID = %d",p1[i].id);
    printf("\nNAME = %s",p1[i].name);
    printf("\nADDRESS = %s\n",p1[i].address);
}
getch();
}
```