C_111 ⇒ Structures in C - Part 3

Initialization and accessing of structure Members.

* Structure can be initialized either at compile time or also at run time.

* First we will discuss about structure initialization at compile time.

normal datatype initial:

```
int a = 5;
Char name[20] = "Jerry";
```

```
struct student
{
  int rollno;
  Char name[20];
  float marks;
};
```

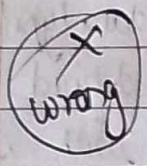
```
void main()
{
  struct student s = {1, "Jerry", 90.9};
}
```

* This is the simple compiler time initialization of memory of structure.

Note :-

* order should be same. Because in structure student the order is first int, then string and float ; so we should initialized it in same order.

Eg:-




```
struct student
{
    int rollno;
    char name[20];
    float marks;
};
void main()
{
    struct student S={"Jenny"
}
```

1st (int)
2nd (string)
3rd (float)
order doesn't matches.
1st
String

* Partial initialization will leads to null values.

```
Eg: struct student {
        int rollno;
        char name[20];
        float marks;
    };
    void main()
    {
        struct student S={ };
    }
```

So here int rollno is initialized with 1 and string to null and also float marks as null (i.e) 0.

* Another way of initialization of members of Structure using Object is given below:-

Eg:- struct student
{
int rollno;
Char name[20];
float marks;
} S={1, "Jessy", 90.91};

(correct with one object/variable of structure)

void main()
{
Comment line // struct student S1={1, "Jessy", 90.91};
}

for more than two Variable/Objects this is not the correct way of initialization

(wrong)

Eg:- struct student
{
int rollno;
Char name[20];
float marks;
} S1={1, "Jessy", 90.91}; S2={2, "Jivya", 85};

(after Semicolon we can't proceed.)

void main()
{
//---

}

Correct way of initialization of members of structure using objects/variables which are more than one.

Ex:- struct student
{
int rollno;
char name[20];
float marks;
};
void main()
{
struct student S1={1, "Jerry", 90.91};
struct student S2={2, "Jiya", 85};
:
:
}

* When we want to store 60 students information; then we can go for array of objects in structure.

* We can also initialize variables/objects in both outside and inside of main.

Eg:- struct student
{
int rollno;
char name[20];
float marks;
} S={35, "Anton", 70};
void main()
{
struct student S1={1, "Jerry", 90.91};
struct student S2={2, "Jiya", 85};
}

Initialized both inside and outside main

Date
Page

* Since the structure is declared outside main, then this structure can be accessed not only by main function but also by other functions.

* Ex:-
```
struct student
{
    int rollno;
    char name[20];
    float marks;
};
void main()
{
    struct student S1={90.91};
}
```

default initialized as int rollno as 90

first value

* How to print members of structure using object?

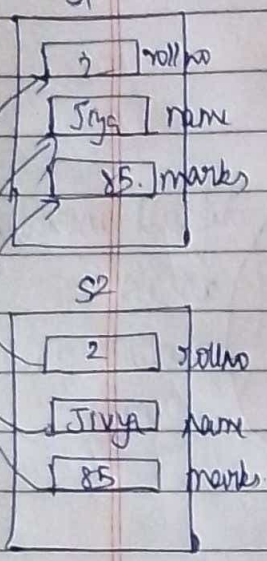
Ex:-
```
struct student
{
    int rollno;
    char name[20];
    float marks;
};
void main()
{
    struct student S1={};
    struct student S2={2,"Jiya",85};
    printf("%d", S1.rollno); ⇒ (1) o/p
}
```

S1
1 rollno
??? name
0 marks

S2
2 rollno
Jiya name
85 marks

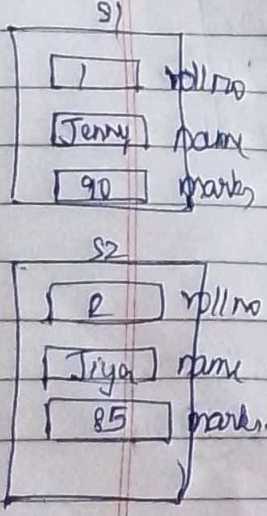
* We can also copy the values of one variable/objects of structure to another variable/objects. but valid only when S1 and S2 are of same type of structures (ie) student.

Ex:
```
struct student
{
    int rollno;
    char name[20];
    float marks;
};
void main()
{
    struct student S1;
    struct student S2 = {2, "Jiya", 85};
    S1 = S2;
    printf("%d", S1.rollno);
    printf("%s", S1.name);
}
```



* We can also initialize members of structure using the object separately.

Ex:
```
struct student {
    int rollno;
    char name[20];
    float marks;
};
void main()
{
    struct student S1;
    struct student S2 = {2, "Jiya", 85};
    S1.rollno = 1;
    S1.name = "Jerry";
    S1.marks = 90;
    strcpy(S1.name, "");
}
```



* How to initialize members of structures using objects/variables at runtime?

Ex:
```
struct student
{
    int rollno;
    char name[20];
    float marks;
};
void main()
{
    struct student S1;
    struct student S2 = {2, "Jiya", 85};
    printf("Enter information for student1 (S1)");
    scanf("%d", &S1.rollno);
    scanf("%s", S1.name);
    scanf("%f", &S1.marks);
}
```

* We cannot compare the objects/variables of structure but we can compare individual members of structure using objects separately.

Ex:
```
struct student
{ int rollno;
  char name[20];
  float marks;
};
void main ()
{
    struct student S1 = {1, "Jenny", 98};
    struct student S2 = {2, "Jiya", 85};
    if (S1 > S2) // if (S1 == S2) // if (S1 != S2)
}
```

We cannot directly compare objects/variables of structure

(x wrong)

Ex:-
```
struct student
{
    int rollno;
    char name[20];
    float marks;
};
void main()
{
    struct student S1 = {1, "Jessy", 98};
    struct student S2 = {2, "Jiya", 85};
    if (S1.rollno < S2.rollno)
    {
        printf("Hi");
    }
}
```

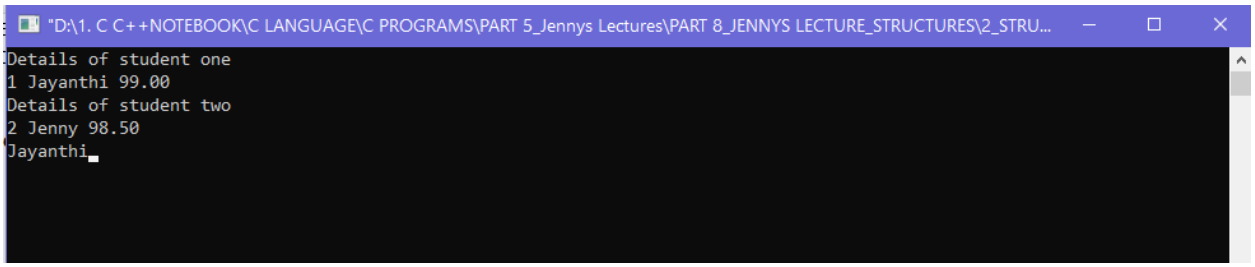
correct

We can Compare individual members of structure using objects

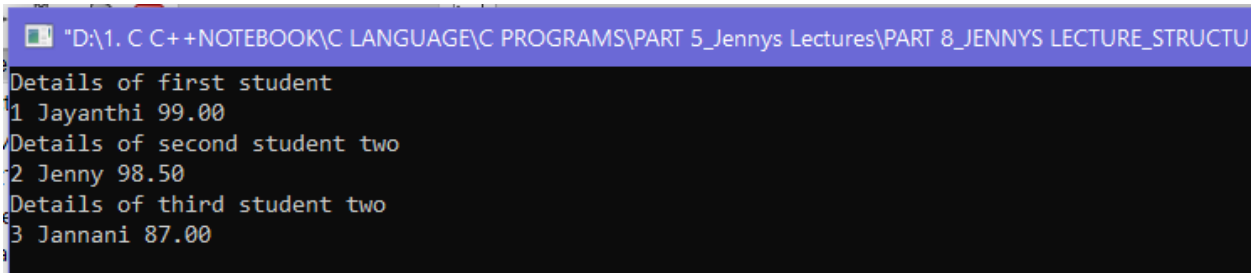
```c
#include <stdio.h>
#include <stdlib.h>
/** 1-STRUCTURE DECLARE, INITIALIZE, ACCESSING **/
/** Structure Declaration, Initialization, Accessing **/
struct student
{
 int rollno;
 char name[20];
 float marks;
};

int main()
{
struct student s1={1,"Jayanthi",98};
struct student s2={2,"Jenny",98.5};
printf("Details of student one\n");
printf("%d %s %0.2f\n",s1.rollno,s1.name,s1.marks);
printf("Details of student two\n");
printf("%d %s %0.2f\n",s2.rollno,s2.name,s2.marks);
getch();
}
```

```
"D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 8_JENNYS LECTURE_STRUCTURES\1_STRU...
Details of student one
1 Jayanthi 98.00
Details of student two
2 Jenny 98.50
```

```c
#include <stdio.h>
#include <stdlib.h>
/** 2-STRUCTURE DECLARE, INITIALIZE, ACCESSING **/
/** Structure Declaration, Initialization, Accessing **/
/** Another way **/
struct student
{
 int rollno;
 char name[20];
 float marks;
};
int main()
{
struct student s1;
s1.rollno=1;
strcpy(s1.name,"Jayanthi");
s1.marks=99;
struct student s2={2,"Jenny",98.5};
printf("Details of student one\n");
printf("%d %s %0.2f\n",s1.rollno,s1.name,s1.marks);
printf("Details of student two\n");
printf("%d %s %0.2f\n",s2.rollno,s2.name,s2.marks);

//Example
char string[20]="Sangeetha";
strcpy(string,"Jayanthi");
printf("%s",string);
getch();
```




```c
#include <stdio.h>
#include <stdlib.h>
/** 3-STRUCTURE DECLARE, INITIALIZE, ACCESSING **/
/** Structure Declaration, Initialization, Accessing **/
/** Another way **/
struct student
{
 int rollno;
 char name[20];
 float marks;
}s3;
int main()
{
struct student s1;
s1.rollno=1;
strcpy(s1.name,"Jayanthi");
s1.marks=99;
struct student s2={2,"Jenny",98.5};
struct student s3={3,"Jannani",87};
printf("Details of first student\n");
printf("%d %s %0.2f\n",s1.rollno,s1.name,s1.marks);
printf("Details of second student two\n");
printf("%d %s %0.2f\n",s2.rollno,s2.name,s2.marks);
printf("Details of third student two\n");
printf("%d %s %0.2f\n",s3.rollno,s3.name,s3.marks);

}
```



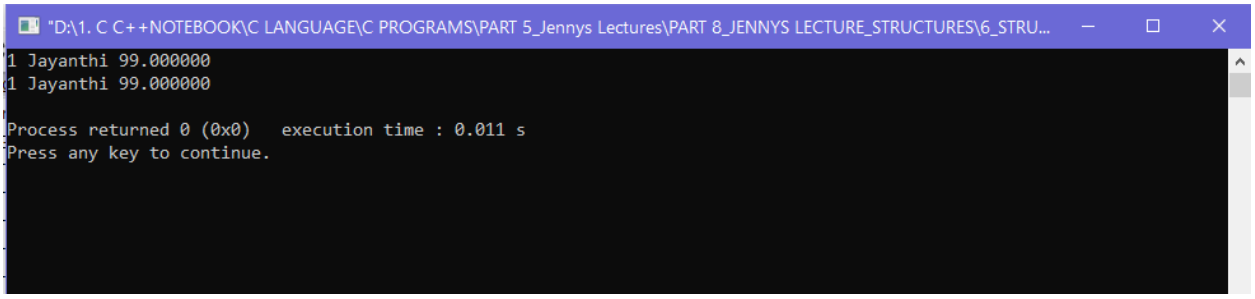
```c
#include <stdlib.h>
/** 4-STRUCTURE DECLARE, INITIALIZE, ACCESSING **/
/** Structure Declaration, Initialization, Accessing **/
/** Another way **/
struct student
{
 int rollno;
 char name[20];
 float marks;
}s3={3,"Jannani",87},s4={4,"JagaPriya",90};
int main()
{
struct student s1;
s1.rollno=1;
strcpy(s1.name,"Jayanthi");
s1.marks=99;
struct student s2={2,"Jenny",98.5};

printf("Details of first student\n");
printf("%d %s %0.2f\n",s1.rollno,s1.name,s1.marks);
printf("Details of second student two\n");
printf("%d %s %0.2f\n",s2.rollno,s2.name,s2.marks);
printf("Details of third student two\n");
printf("%d %s %0.2f\n",s3.rollno,s3.name,s3.marks);
printf("Details of fouth student two\n");
printf("%d %s %0.2f\n",s4.rollno,s4.name,s4.marks);
}
```

```
"D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 8_JENNYS LECTURE_STRUCTURES\4_STRU...
Details of first student
1 Jayanthi 99.00
Details of second student two
2 Jenny 98.50
Details of third student two
3 Jannani 87.00
Details of fouth student two
4 JagaPriya 90.00

Process returned 0 (0x0)   execution time : 0.043 s
Press any key to continue.
```

```c
#include <stdlib.h>
/** 5-STRUCTURE DECLARE, INITIALIZE, ACCESSING **/
/** Structure Declaration, Initialization, Accessing **/
/** Another way **/
struct student
{
 int rollno;
 char name[20];
 float marks;
};
struct student s3={3,"Jannani",87};

int main()
{
struct student s1;
s1.rollno=1;
strcpy(s1.name,"Jayanthi");
s1.marks=99;
struct student s2={2,"Jenny",98.5};

printf("Details of first student\n");
printf("%d %s %0.2f\n",s1.rollno,s1.name,s1.marks);
printf("Details of second student two\n");
printf("%d %s %0.2f\n",s2.rollno,s2.name,s2.marks);
printf("Details of third student two\n");
printf("%d %s %0.2f\n",s3.rollno,s3.name,s3.marks);
}
```

```
"D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 8_JENNYS LECTURE_STRUCTURES\5_STRU...
Details of first student
1 Jayanthi 99.00
Details of second student two
2 Jenny 98.50
Details of third student two
3 Jannani 87.00

Process returned 0 (0x0)   execution time : 0.042 s
Press any key to continue.
```

```c
#include <stdio.h>
#include <stdlib.h>
/** 6-STRUCTURE DECLARE, INITIALIZE, ACCESSING **/
/** Structure Declaration, Initialization, Accessing **/
/** Copy values of one object/variable to another object/variable which belongs
     to same structure data type **/

struct student
{
 int rollno;
 char name[20];
 float marks;
};

struct student s1={1,"Jayanthi",99};
struct student s2;
int main()
{
    s2=s1;
    printf("%d %s %f\n",s1.rollno,s1.name,s1.marks);
    printf("%d %s %f\n",s2.rollno,s1.name,s1.marks);
}
```



```c
#include <stdio.h>
#include <stdlib.h>
/** 7-STRUCTURE DECLARE, INITIALIZE, ACCESSING **/
/** Structure Declaration, Initialization, Accessing **/
/** We cannot directly compare two structure objects but we can compare
 members of structure individually using dot operator **/
struct student
{
 int rollno;
 char name[20];
 float marks;
};
struct student s1={1,"Jayanthi",99};
struct student s2;
int main()
{
    s2=s1;
    if(s2==s1)
     printf("Two students get equal marks");
    else
     printf("Two students get different marks");
    printf("%d %s %f\n",s1.rollno,s1.name,s1.marks);
    printf("%d %s %f\n",s2.rollno,s2.name,s2.marks);
```

Logs & others

Code::Blocks | Search results | Cccc | Build log | Build messages | CppCheck/Vera++ | CppCheck/Vera++ messages | Cscope

File	Line	Message
D:\1. C C++...	18	error: invalid operands to binary == (have 'struct student' and 'struct stu...
		=== Build failed: 1 error(s), 0 warning(s) (0 minute(s), 0 second(s)) ===

```c
/** 8-STRUCTURE DECLARE, INITIALIZE, ACCESSING **/
/** Structure Declaration, Initialization, Accessing **/
/** We cannot directly compare two structure objects but we can compare
 members of structure individually using dot operator **/
struct student
{
 int rollno;
 char name[20];
 float marks;
};
struct student s1={1,"Jayanthi",99};
struct student s2;
int main()
{
 s2=s1; //we can copy values of members of structure from one object/variable to other object
    if(s2.marks==s1.marks)
     printf("Two students get equal marks\n");
    else
     printf("Two students get different marks");
    printf("Marks of student 1\n");
    printf("%d %s %f\n",s1.rollno,s1.name,s1.marks);
    printf("Marks of student 2\n");
    printf("%d %s %f\n",s2.rollno,s2.name,s2.marks);
}
```

```
"D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 8_JENNYS LECTURE_STRUCTURES\8_STRU...
Two students get equal marks
Marks of student 1
1 Jayanthi 99.000000
Marks of student 2
1 Jayanthi 99.000000

Process returned 0 (0x0)   execution time : 0.040 s
Press any key to continue.
```

```c
#include <stdio.h>
#include <stdlib.h>
/** 9-STRUCTURE DECLARE, INITIALIZE, ACCESSING **/
/** Structure Declaration, Initialization, Accessing **/
/** We cannot change the order of initialization of structure members **/
strct student
{
 int rollno;
 char name[20];
 float marks;
}s1={"jayanthi",1,89};

int main()
{
    printf("Marks of student 1");
    printf("%s %d %f",s1.name,s1.rollno,s1.marks);
}
```

Logs & others

Code::Blocks | Search results | Cccc | Build log | Build messages | CppCheck/Vera++ | CppCheck/Vera++ messages | Cscope

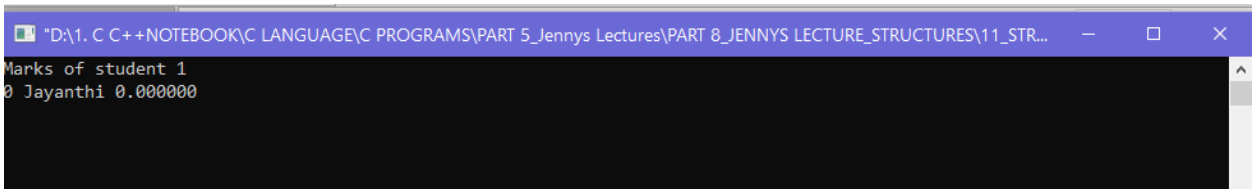
File	Line	Message
D:\1. C C++...	11	note: (near initialization for 's1')
D:\1. C C++...	11	error: initializer element is not computable at load time
D:\1. C C++...	11	note: (near initialization for 's1')
D:\1. C C++...	11	warning: excess elements in scalar initializer
D:\1. C C++...	11	note: (near initialization for 's1')
D:\1. C C++...	11	warning: excess elements in scalar initializer

```c
#include <stdio.h>
#include <stdlib.h>
/** 10-STRUCTURE DECLARE, INITIALIZE, ACCESSING **/
/** Structure Declaration, Initialization, Accessing **/
/** Initialization of structure members is optional,
    Uninitialized structure members are initialized with null values **/
struct student
{
 int rollno;
 char name[20];
 float marks;
}s1={1};

int main()
{
    printf("Marks of student 1\n");
    printf("%d %s %f",s1.rollno,s1.name,s1.marks);
    getch();
}
```

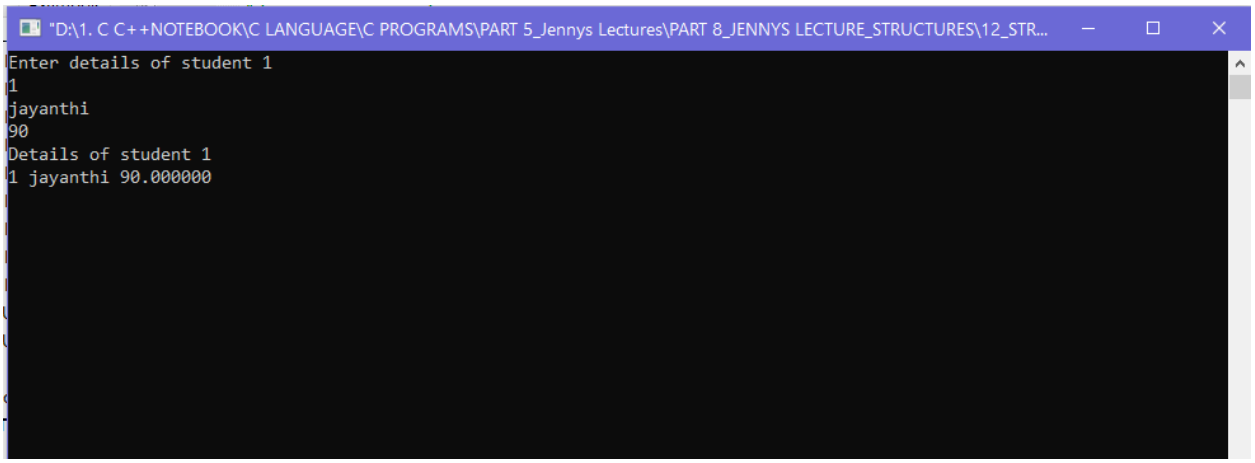
"D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 8_JENNYS LECTURE_STR

```
Marks of student 1
1  0.000000
```

```c
#include <stdio.h>
#include <stdlib.h>
/** 11-STRUCTURE DECLARE, INITIALIZE, ACCESSING **/
/** Structure Declaration, Initialization, Accessing **/
/** Initialization of structure members is optional,
    Uninitialized structure members are initialized with null values **/
struct student
{
 int rollno;
 char name[20];
 float marks;
};
struct student s1;
int main()
{

    strcpy(s1.name,"Jayanthi");
    printf("Marks of student 1\n");
    printf("%d %s %f",s1.rollno,s1.name,s1.marks);
    getch();
}
```




```c
#include <stdio.h>
#include <stdlib.h>
/** 12-STRUCTURE DECLARE, INITIALIZE, ACCESSING **/
/** Structure Declaration, Initialization, Accessing **/
/** Run time initialization of structure members using object/variable of structure datatype**/
struct student
{
 int rollno;
 char name[20];
 float marks;
};

int main()
{
    struct student s1;
    printf("Enter details of student 1\n");
    scanf("%d %s %f",&s1.rollno,s1.name,&s1.marks);
    printf("Details of student 1\n");
    printf("%d %s %f",s1.rollno,s1.name,s1.marks);
    getch();
}
```


"D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 8_JENNYS LECTURE_STRUCTURES\12_STR...
Enter details of student 1
1
jayanthi
90
Details of student 1
1 jayanthi 90.000000