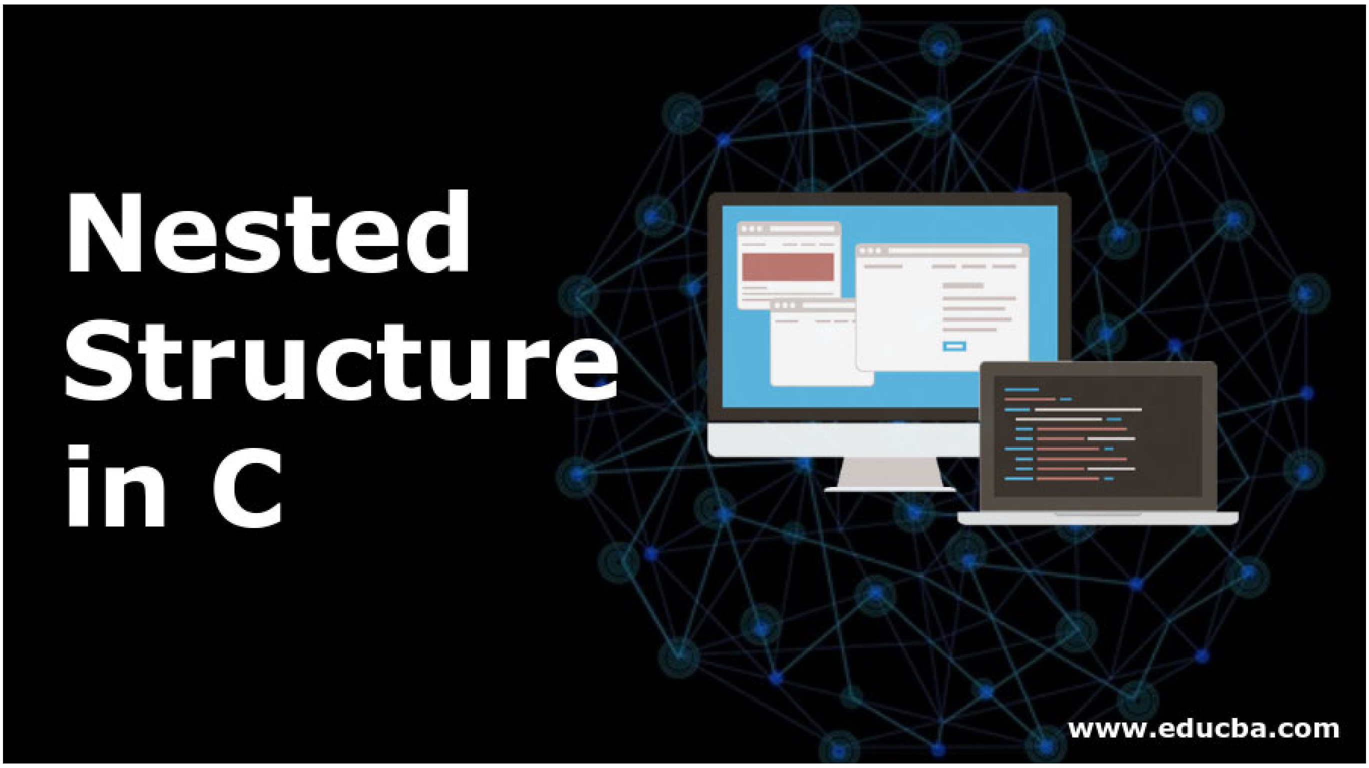
EDUCBA
(https://www.educba.com/software-development/)

←
(https://www.educba.com/structure-padding-in-c/)

→
(https://www.educba.com/continue-statement-in-c/)



Introduction to Nested Structure in C

Any programming language has its own way of defining and describing structures. So I structures as its name suggest in C is kind of defining one structure inside another structure. Any member variables can be defined inside a structure and in turn, that structure can further

EDUCBA (https://www.educba.com/software-development/)

Start Your Free Software Development Course

Web development, programming languages, Software testing & others

1. Structure inside a structure in C using the pointer variable.
2. Structure inside a structure in C using a normal variable.

Syntax:

Following is the syntax for creating a nested structure:

```
structure tagname_1
{
var_1;
var_2;
var_3;
.
.
.
.
var n;
structure tagname_2
{

var_1;
var_2;
var_3;
}; mem1
} mem2;
```

Working of Nested Structure in C

From the above syntax, we can infer the fact that mem1 structure nested inside member1 structure will contain the member or the variable to be accessed and everyone can be accessed in a nested manner by using. (dot) operator.

- **mem2.mem1.var_1:** This refers to the first member of the variable of the structure tagname_1.
- **mem2.mem1.var_2:** This refers to the second member of the variable of the structure tagname_2.

We will take more examples to get clarity on how the syntax satisfies the working of the nested structure.

Examples #1

```
struct employee
{
struct man
{
char name [20];

int age;
char dob[10];
} d;
```

In the above example, man structure is defined inside an employee structure which is a nested structure. Members within the nested structure which is a man can be accessed using the below syntax or format.

Offer
LEARN from Home

⚡ Popular Course in this category

C Programming Training (3 Courses, 5 Project)

3 Online Courses | 5 Hands-on Projects | 34+ Hours | Verifiable Certificate of Completion | Lifetime Access

★★★★★ 4.5 (8,604 ratings)

Course Price

$79 ~~$399~~

View Course

(https://www.educba.com/software-development/courses/c-programming-course/?btnz=edu-blg-inline-banner1)

Related Courses

C++ Training (4 Courses, 5 Projects, 4 Quizzes) (https://www.educba.com/software-development/courses/c-course/?btnz=edu-blg-inline-banner1)

Java Training (40 Courses, 29 Projects, 4 Quizzes) (https://www.educba.com/software-development/courses/java-course/?btnz=edu-blg-inline-banner1)

QUIZ

Like in the given example

- **employee.d .name:** It tells about the name of the man inside the employee structure.
- **employee.d .age:** It will tell about the age of the man as an employee.

Instead, a workaround for this can be:

We could have defined the structure outside and then could have declared the variable inside the structure wherever we want to access it throughout the code.

Examples #2

```
Struct man
{
char name[20];
int age;
char dob [10];
};
```

Also, this structure can be reused by the outer structure.

```
struct employee
{
struct man info;
int id;
char desg [10];
}
```

The advantage of using this type of structure declaration is that we can declare a variab type struct man anywhere throughout the program.

```
struct teacher
{
char name[20];
char address[100];
int age[];
struct teacher principal; // totally invalid way to create nested
structure.
}
```

Examples of Nested Structures in C

Below are the different examples of nested structure in C:

Example #1 – Initialization of nested structures:

Initialization of nested structures is possible at the time of declaration.

Code:

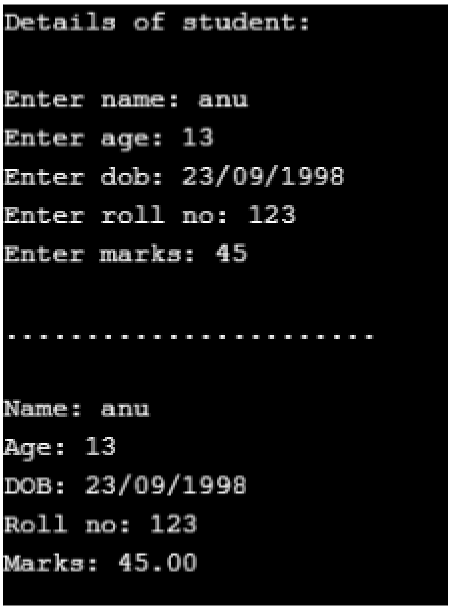
```
struct student
{
struct person info;
int rollno;
float marks[10];

}
struct student student_1 = {
{"Anji", 26, 1995},
```

Example

Code:

```
#include <stdio.h>
struct person
{
char name[20];
int age;
char dob[10];
};
struct student
{
struct person info;
int roll_no;
float marks;
};
int main ()
{
struct student p1;
printf("Details of student: \n\n");
printf("Enter name: ");
scanf("%s", p1.info.name);

printf("Enter age: ");
scanf("%d", &p1.info.age);
printf("Enter dob: ");
scanf ("%f", &p1.marks);
printf("\n......................\n\n");
printf("Name: %s\n", p1.info.name);
printf("Age: %d\n", p1.info.age);
printf("DOB: %s\n", p1.info.dob);
printf("Roll no: %d\n", p1.roll_no);
printf("Marks: %.2f\n", p1.marks);
return 0;
}
```

Output:



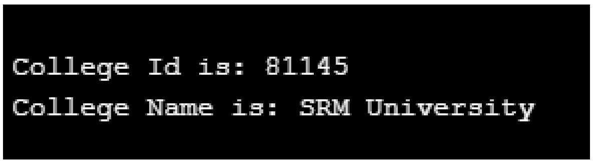
```
Details of student:

Enter name: anu
Enter age: 13
Enter dob: 23/09/1998
Enter roll no: 123
Enter marks: 45


......................

Name: anu
Age: 13
DOB: 23/09/1998
Roll no: 123
Marks: 45.00
```

Example #2 – Accessing of members inside nested struc using Pointers:

Code:

```
{
int college_id;
char college_name[50];
};
struct student_info
{
int id;
char name[20];
float percentage;
struct student_college_info clg_data;
} stu_data, *stu_data_ptr;
int main()
{
struct student_info stu_data = {2, "Anu", 92.5, 81145,
"SRM University"};
stu_data_ptr = &stu_data;
printf(" Id is: %d \n", stu_data_ptr->id);
printf(" Name is: %s \n", stu_data_ptr->name);
printf(" Percentage is: %f \n\n",
stu_data_ptr->percentage);
printf(" College Id is: %d \n",
stu_data_ptr->clg_data.college_id);
printf(" College Name is: %s \n",
stu_data_ptr->clg_data.college_name);
return 0;
}
```

EDUCBA
(https://www.educba.com/software-development/)



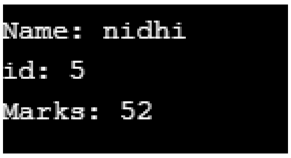
```
College Id is: 81145
College Name is: SRM University
```

Example #3 – Passing structure member as arguments to function:

Code:

```
struct teacher
{
char name [20];
int id;
int marks;
};
void print_struct (char name [], int id, int marks);
int main ()
{
struct teacher tea = {"nidhi", 5, 52};
print_struct (tea.name, tea.id, tea.marks);
return 0;
}
void print_struct (char name [], int id, int marks)
{

printf ("Name: %s\n", name);
printf ("id: %d\n", id);
printf ("Marks: %d\n", marks);
```

EDUCBA (https://www.educba.com/software-development/)



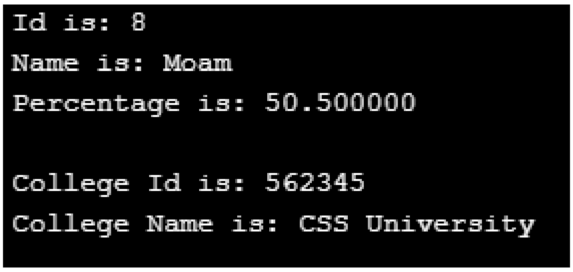
```
Name: nidhi
id: 5
Marks: 52
```

Example #4 – Structure inside structure using a normal variable.

Code:

```
#include <stdio.h>
#include <string.h>
struct student_college_detail
{
nt college_id;
char college_name[50];
};
struct student_detail
{
int id;
char name[20];
float percentage;
struct student_college_detail clg_data;
} stu_data;
int main()

{
struct student_detail stu_data = {8, "Moam", 50.5, 562345,
"CSS University"};
printf(" College Id is: %d \n",
stu_data.clg_data.college_id);
printf(" College Name is: %s \n",
stu_data.clg_data.college_name);
return 0;
}
```

Output:



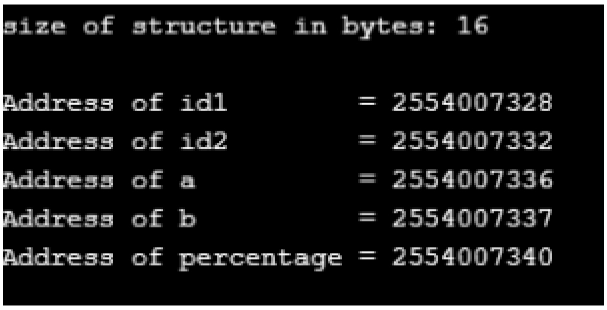
```
Id is: 8
Name is: Moam
Percentage is: 50.500000

College Id is: 562345
College Name is: CSS University
```

Example

Code:

```
#include <stdio.h>
#include <string.h>
struct student
{
int id1;
int id2;
char e;
char f;
float percentage;
struct student recrd1 = {3, 4, 'C', 'B', 80.5};
printf ("size of structure in bytes: %d\n",
sizeof(recrd1));
printf ("\nAddress of id1        = %u", &recrd1.id1);
printf("\nAddress of id2        = %u", &recrd1.id2 );
printf("\nAddress of a          = %u", &recrd1.e );
printf("\nAddress of b          = %u", &recrd1.f );
printf("\nAddress of percentage = %u”, &recrd1.percentage);
return 0;
}
```

Output:



```
size of structure in bytes: 16

Address of id1        = 2554007328
Address of id2        = 2554007332
Address of a          = 2554007336
Address of b          = 2554007337
Address of percentage = 2554007340
```

> **Note:** Although it is good to pass structure variables as an argument because it allows us to pass all members of structure to function but still this is not a conventional method to do so.

Conclusion

Structures in C is a very interesting way to cluster and group all user-defined member variables and functions into one entity. But still, it has some limitations like it does not allow structure

EDUCBA (https://www.educba.com/software-development/)

This is a guide to the Nested Structure in C. Here we discuss the working in Nested Structure in C along with different examples and code implementation. You may also look at the following article to learn more –

1. Patterns in C Programming (https://www.educba.com/patterns-in-c-programming/)
2. Left Shift Operator in C (https://www.educba.com/left-shift-operator-in-c/)
3. Expression Evaluation in C (https://www.educba.com/expression-evaluation-in-c/)
4. Decimal to Hexadecimal in C (https://www.educba.com/decimal-to-hexadecimal-in-c/)

C PROGRAMMING TRAINING (3 COURSES, 5 PROJECT)

☑ 3 Online Courses

☑ 5 Hands-on Projects

☑ 34+ Hours

☑ Verifiable Certificate of Completion

☑ Lifetime Access

Learn More

(https://www.educba.com/software-development/courses/c-programming-course/?btnz=edu-blg-inline-banner3)

QUIZ

EDUCBA (https://www.educba.com/software-development/)

Blog (https://www.educba.com/blog/?source=footer)

Who is EDUCBA? (https://www.educba.com/about-us/?source=footer)

Sign Up (https://www.educba.com/software-development/signup/?source=footer)

Corporate Training (https://www.educba.com/corporate/?source=footer)

Certificate from Top Institutions (https://www.educba.com/educbalive/?source=footer)

Contact Us (https://www.educba.com/contact-us/?source=footer)

Verifiable Certificate (https://www.educba.com/software-development/verifiable-certificate/?source=footer)

Reviews (https://www.educba.com/software-development/reviews/?source=footer)

Terms and Conditions (https://www.educba.com/terms-and-conditions/?source=footer)

Privacy Policy (https://www.educba.com/privacy-policy/?source=footer)

Apps

iPhone & iPad (https://itunes.apple.com/in/app/educba-learning-app/id1341654580?mt=8)

Android (https://play.google.com/store/apps/details?id=com.educba.www)

Resources

Free Courses (https://www.educba.com/software-development/free-cour
source=footer)

QUIZ

Java Tutorials (https://www.educba.com/software-development/software-development-tutorials/java-tutorial/?source=footer)

EDUCBA
(https://www.educba.com/software-development/)

Certification Courses

All Courses (https://www.educba.com/software-development/courses/?source=footer)

Software Development Course - All in One Bundle (https://www.educba.com/software-development/courses/software-development-course/?source=footer)

Become a Python Developer (https://www.educba.com/software-development/courses/python-certification-course/?source=footer)

Java Course (https://www.educba.com/software-development/courses/java-course/?source=footer)

Become a Selenium Automation Tester (https://www.educba.com/software-development/courses/selenium-training-certification/?source=footer)

Become an IoT Developer (https://www.educba.com/software-development/courses/iot-course/?source=footer)

ASP.NET Course (https://www.educba.com/software-development/courses/asp-net-course/?source=footer)

VB.NET Course (https://www.educba.com/software-development/courses/vb-net-course/?source=footer)

PHP Course (https://www.educba.com/software-development/courses/php-course/?source=footer)

© 2022 - EDUCBA. ALL RIGHTS RESERVED. THE CERTIFICATION NAMES ARE THE TRADEMARKS OF THEIR RESPECTIVE OWNERS.

QUIZ