Initializing and Accessing Members of a Structure Variable in C++

In this lesson, you will see the basic syntax for accessing the members of a structure.

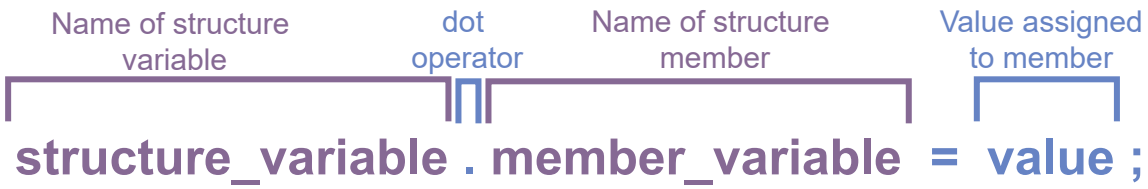
We'll cover the following

Introduction

We have seen how to define a structure and then declare a structure variable in a program. Let's see how we can store data in the member variables of the structure.

Basic syntax

The basic syntax for storing values in the member of the structure variable is given below:



To access the member of the structure variable, we write the name of the structure variable, followed by a dot operator, which is further followed by the name of the member. To assign a value to the member variable, we use the equal operator followed by the value and the statement terminator (i.e., semicolon).

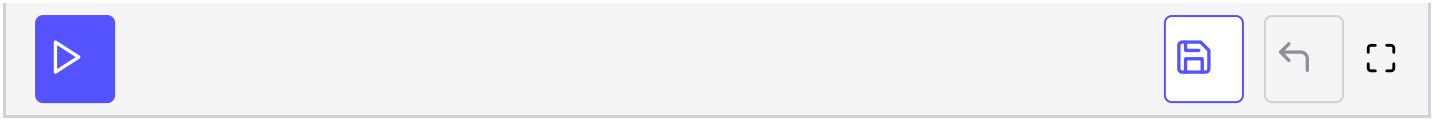
Example program

Let's store values in structure variables `s1`, `s2`, and `s3`.

```
#include <iostream>

using namespace std;
// Student structure
struct Student {
  string name;
  int roll_number;
  int marks;
};
// main function
int main() {
  Student s1, s2, s3;
// Assign value to members of s1
  s1.name = "John";
  s1.roll_number = 1;
  s1.marks = 50;
  cout << "s1 Information:" << endl;
// Print members of s1
  cout << "Name = " << s1.name << endl;
  cout << "Roll Number = " << s1.roll_number << endl;
  cout << "Marks = " << s1.marks << endl;
// Assign value to members of s2
  s2.name = "Alice";
  s2.roll_number = 2;
  s2.marks = 43;
// Print members of s2
  cout << "s2 Information:" << endl;
  cout << "Name = " << s2.name << endl;
  cout << "Roll Number = " << s2.roll_number << endl;
  cout << "Marks = " << s2.marks << endl;

  return 0;
}
```



Explanation

Line No. 14: We are accessing the member `name` of `s1` using the dot operator and then set it to **John**.

Similarly, we access the member's `roll_number` and `marks` of `s1` and set their values.

We repeat the same procedure to set the values for the rest of the structure variables.

> When we declare a structure variable, all of its members are initialized to their default values.

Initializing members in one line

You are probably thinking, setting each member of the structure variable is a tedious task. So, is there a way to set all the members of structure variables in one line?

Yes, there is. We can also initialize structure variables in one line using the initializer list.

```
structure_variable  =  {member1_value , member2_value, .....member(n)_value} ;
```

We will assign a comma-separated list of values enclosed in a curly bracket to the structure variable. The first member will be assigned the first value in the curly bracket; the second member will be assigned the second value, and so on.

> ℹ If the initializer list does not have some member of structure variables, then those members are automatically initialized to their default value.

See the program given below!

```
#include <iostream>

using namespace std;

struct Student {
  string name;
  int roll_number;
  int marks;
};

int main() {
  struct Student s1, s2, s3;

  s1 = {"John", 1, 50};

  cout << "s1 Information:" << endl;
  cout << "Name = " << s1.name << endl;
  cout << "Roll Number = " << s1.roll_number << endl;
  cout << "Marks = " << s1.marks << endl;

  s2 = {"Alice", 2, 43};

  cout << "s2 Information:" << endl;
  cout << "Name = " << s2.name << endl;
  cout << "Roll Number = " << s2.roll_number << endl;
  cout << "Marks = " << s2.marks << endl;

  return 0;
```

```
}
```

In the program above, we set the members of `s1` and `s2` in one line.

Quiz

Q

If we print the member values of `s1`, then what is the output of the following code.

```
struct Student {
  string name;
  int roll_number;
  int marks;
};

int main() {
  struct Student s1;
  s1 = {"Alex", 2};
```

Retake Quiz

Let’s study an array of structures in the upcoming lesson.

Stay tuned!