



# Defining Structure in C++

Learn about the basic syntax for defining structures in C++.

We'll cover the following ^

- Introduction
  - Defining structure
- Example program
  - Explanation

## Introduction#

Structure is a user-defined data type. Therefore, before using structure in a program, we must tell the compiler what our structure will look like.

## Defining structure#

The basic syntax for defining a structure in C++ is given below:



The diagram illustrates the syntax of a C++ struct definition. It shows the following code snippet:

```
struct struct_name {  
    datatype member1;  
    datatype member2;  
    .  
    .  
    .  
    datatype member(n);  
};
```

Annotations with brackets and labels identify the components:

- Keyword:** Points to the word `struct`.
- name of structure:** Points to `struct_name`.
- body of structure:** A large bracket on the right side of the opening curly brace `{` encompasses the entire block of members (from `datatype member1;` down to `};`).

To define a structure in a program, use the `struct` keyword followed by a structure name, which is followed by curly braces and a semicolon at the end. Inside the curly braces, we declare the data members of the structure.

✗ Forgetting a semicolon after the structure definition generates an error.

## Example program#

Suppose that we want to store the record of student name, roll number, and marks in a single location. Let's see in the example below how a struct can help us:

```
1  #include <iostream>
2
3  using namespace std;
4  // Student structure
5  struct Student {
6      string name;
7      int roll_number;
```



```
8   int marks;  
9   };  
10  // main function  
11  int main() {  
12  
13      return 0;  
14  }
```



## Explanation#

In the above program, we have defined the structure `Student` from **Lines No 5 to 9**. `name`, `roll_number`, and `marks` are the data members of the `Student`.



(/learn)

Have you noticed anything?

Here, we declare the variables of different data types under the same name.



We will use `struct_name` later in a program to create a structure variable.



A structure cannot contain a member of its own type.

### Quiz



Q

The code given below:



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

Your Answer



A) Generates an error

Explanation

Forgetting semicolon after the structure definition will generate an error.



B) Compiles successfully

Submit Answer

Reset Quiz ↺

Even though we have declared members in `Student` structure, the compiler has not allocated any memory to them yet.

Let's see how to allocate memory to the structure members in the upcoming lesson.

← Back

Next →