Defining Structure in C++

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

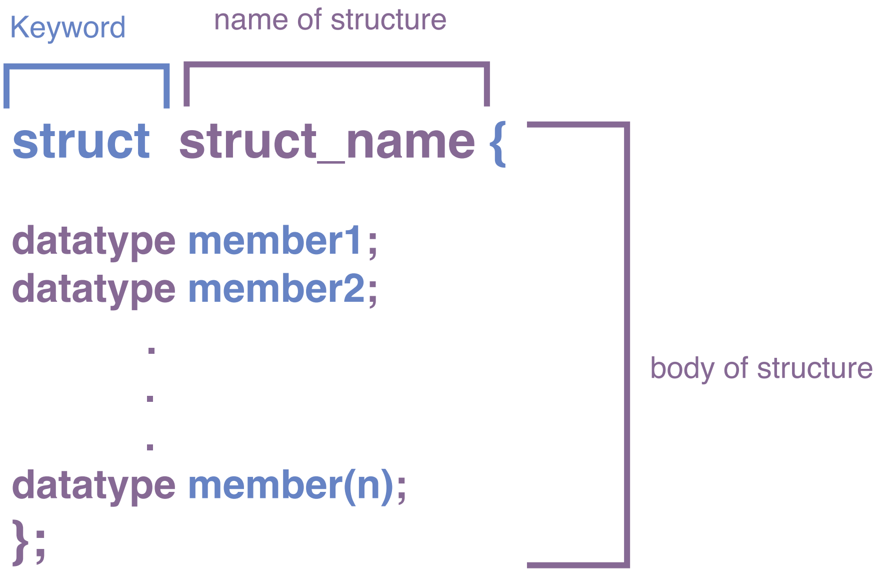
We'll cover the following

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:



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:

```
#include <iostream>

using namespace std;
// Student structure
struct Student {
  string name;
  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`.

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 →