Defining Structure in C++

Let's see the basic syntax for defining structure in C++.

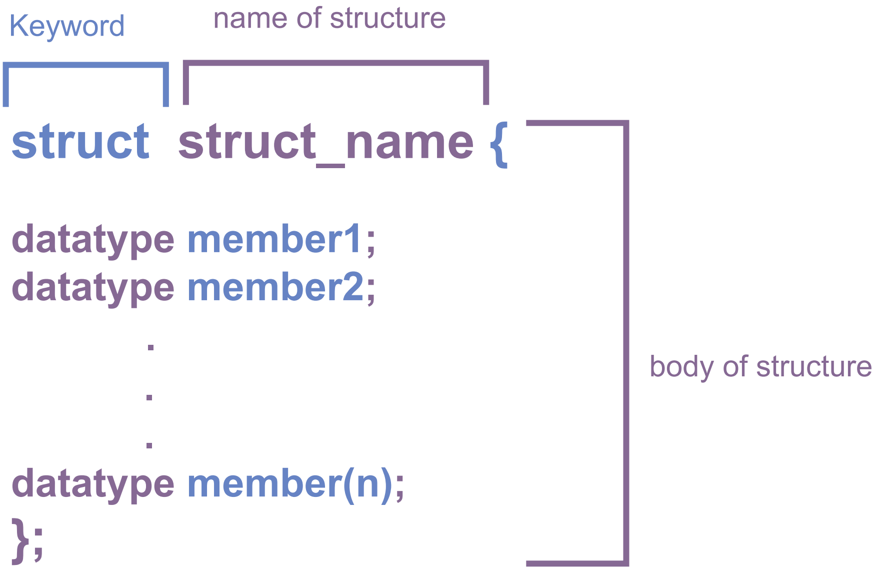
We'll cover the following

Introduction

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

Defining structure

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



```
struct  struct_name {

datatype member1;
datatype member2;
        .
        .
        .
datatype member(n);
};
```

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;
  int marks;
};
// main function
int main() {

  return 0;
}
```

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.

Quiz

Q

The code given below:

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

```
    int roll_number;
    int marks;
}
```

Retake Quiz

Even though we have declared the members in the `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.