Introduction to Structures

In this lesson, you will be introduced to the concept of structure.

We'll cover the following

- What is structure?
- Example

What is structure?#

Consider a **blueprint** used to construct a building. A blueprint is a guide that tells us what the basic architecture of the building is. For example, it will detail the number of floors, rooms, windows, etc. in the building.

We can use the same blueprint to construct multiple buildings, but each building will be different from others in properties. For example, if one building is of a white color, the other will be a red color.

Structure is just like a blueprint from which we can create a variable of our own data type.

> *The **Structure** is a user-defined data type that is used to store variables or arrays of different data types under a single name.*

Example#

Suppose there are **100** students in a class, and you want to store their names, roll numbers, and marks.

To store data of each student, we can create **3** variables for each student. In total, we have to create **300** variables, which does not make sense.

Arrays can be used to store data of a similar kind. Here, the student's name will be `string`, and their roll number will be `int`. So, we cannot use arrays here!

Structures let us store data of different types.

We will define a `Student structure` that will act as a blueprint in our program. The `Student structure` will have **3 members:** `name`, `roll_number`, and `marks` to store the information of `Student`.

We will then declare a variable whose type will be a `Student` for each student in the class that is known as the **structure variable**.

If the **structure** is like a blueprint on the page, the **structure variable** is like a building that has an actual physical existence and where we can live.

Let's get into the details of structure implementation in C++.

← Back

Solution: Mini Project 2

Next →

Defining Structure in C++

Mark as Completed

Report an Issue