

# Class (class) and Object in C ++

The main purpose of C ++ is to add object orientation to the C Programming Language and classes which are central features of C ++ that support object-oriented programming and are often called types user defined (user-defined).

The main purpose of C ++ is to add object orientation to the C Programming Language and classes which are central features of C ++ that support object-oriented programming and are often called types user defined (user-defined).

A class is used to determine the form of an object and it connects data representation and methods to manipulate that data into a neat package. Data and functions within a class are called members of that class.

## Class definition in C ++

When you define a class, you define a blueprint for a data type. This does not really define any data, but it defines what the meaning of the class name is, that is, what an object of that class will include and what activities can be performed on one that object.

A C ++ class definition begins with the class keyword, followed by the class name and class body, surrounded by a pair of curly braces. A class definition must be followed: either by a semicolon or a list of declarations. For example, we define Box data type using the **class** keyword C ++ as follows:

```
class Box { public : double chieudai ; // chieu dai cua hop double chieurong
```

The **public** keyword determines the access attributes of the class members that follow it. A public member can be accessed from outside the class anywhere within the scope of that class object. You can also define whether members of the class are **private** or **protected** to be discussed in the sub-chapter.

## Object Definition in C ++

A class provides blueprints for objects, so basically an object is created from a class. We declare the objects of a class exactly like we declare variables of the basic type. The following commands declare two objects of the Box class:

```
Box Box1 ; // Khai bao Box1 la cua kieu Box Box Box2 ; // Khai bao Box2 la cua B
```

Both Box1 and Box2 objects will have copies of their own Data Members.

## Access data members in C ++

Public data members of objects of a class that can be accessed using the member access operator directly are dots (.). You will see clearly when looking at the following example:

```
#include <string> using namespace std ; class Box { public : double chieudai ; // chieudai
```

Compiling and running the above C ++ program will produce the following results:

```
The tich cua Box1 la: 14.85
The tich cua Box2 la: 33.12
-----
```

The important point to remember is: private and protected members cannot be accessed directly by using this direct member access operator. We will learn how private and protected members can be accessed.

## Details about Classes & Objects in C ++

At this point, you have a fairly basic understanding of Classes and Objects in C ++. Here are some interesting concepts related to Classes and Objects in C ++ that you need to care about. [Click on the link to see details.](#)

### Concept Description

#### Class member functions in C ++

A member function of a class is a function with its definition or its prototype within the class definition like any other variable.

#### Access Modifier for class in C ++

A class member can be defined as public, private or protected. By default, the members will be private

#### Constructor & Destructor in C ++

A class constructor is a special function in a class that is called when a new object of that class is created. A destructor class is also a special function that is called when the created object is destroyed

#### Copy Constructor in C ++

Copy constructor is a constructor that creates an object by initializing it with an object of the same class, which was created earlier.

#### Friend function in C ++

The *friend* function is allowed to access members who are private and protected of a class

#### Inline function in C ++

With an inline function, the compiler tries to extend the code in the body of the function to replace a call to that function

#### Cursor this in C ++

Each object has a special pointer, which points to the object itself

Cursor to class in C ++

A pointer to a class is done exactly the same way a pointer to a structure. In fact, a class is a structure with functions in it

Static members in C ++

Both data members and function members can be declared static

**According to Tutorialspoint**

Last post: Struct in C / C ++

Next lesson: Function of class members in C ++

You finished reading the article "**Class (class) and Object in C ++**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.