

Access Modifier for class in C ++

Data Hiding is one of the important features of object-oriented programming that allows to prevent the function of a program from directly accessing the internal representation of a class type.

Data Hiding is one of the important features of object-oriented programming that allows to prevent the function of a program from directly accessing the internal representation of a class type. Limiting access to class members defined by public, private and protected areas is labeled inside the class body. The **public, private and protected** keywords are called Access Specifier.

A class can have multiple public, private and protected areas labeled. Each area remains effective until it meets: or another regional label or closing parenthesis of the class body. The default Access Specifier for members and classes is private.

```
class Base { public : // tai day la cac thanh vien public protected : // tai day
```

Members are public in C / C ++

Public members are accessible from anywhere outside the class but are inside a program. You can set and retrieve the values ??of public variables without member functions, as in the following example:

```
#include using namespace std ; class Line { public : double chieudai ; void
```

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

```
Do dai cua duong la: 50.2
Do dai cua duong la: 24.7
```

Private member in C / C ++

A variable or a **private** member function in C / C ++ cannot be accessed, or viewed from outside the class. Only classes and friend functions can access private members in C / C ++.

By default, all members of a class will be **private** , for example, **chieurong** is a private member, that is, when you label a member, it will be treated as a member. **private** .

```
class Box { double chieurong ; public : double chieudai ; void setChieuRong
```

In fact, we define data in the private area and related public functions so that they can be called from outside the class, like the following program:

```
#include using namespace std ; class Box { public : double chieudai ; void s
```

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

```
Chieu dai cua Box la: 50.2  
Chieu rong cua Box la: 22.4
```

Protected member in C / C ++

A **protected** variable or a member function is quite similar to a private member, but it provides an additional benefit that they can be accessed in subclasses, which are called inheritance classes.

You will learn about inheritance and inheritance in the next chapter. Now, you can check the following example, here, I have inherited a **SmallBox** subclass from a **Box** class called **Box** .

The following example is similar to the above example and here the **chieurong** member will be accessible by any member function of the SmallBox inheritance class.

```
#include using namespace std ; class Box { protected : double chieurong ; }; c
```

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

```
Do rong cua Box la : 15.3
```

According to Tutorialspoint

Previous article: Function of class members in C ++

Next lesson: Constructor and Destructor in C ++

You finished reading the article "**Access Modifier for class 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.