

Inheritance in C Plus Plus

Module 2 - Access Control and Inheritance in C Plus Plus



CBSE

Inheritance:

The ability to access members of one class by another class called inheritance. At least two classes are required for inheritance.



ICSE

1. Base class (Old class / Super class): A class that provides members to another class called base class.



NTSE

2. Derived class (New class / sub class): A class that can access members of base class called derived class. Derived class can access only protected and public members of base class.



Banking &
Insurance

Access specifiers in C++.

public - members are accessible from outside the class, and members can be accessed from anywhere.

private - members cannot be accessed (or viewed) from outside the class, i.e members are private to that class only.

protected - members cannot be accessed from outside the class, but, they can be accessed in inherited classes or derived classes.



Central Govt.
Service

Public, Protected, and Private inheritance in C++

public, protected, and private inheritance have the following features:

- **public inheritance** makes public members of the base class public in the derived class, and the protected members of the base class remain protected in the derived class.

- **protected inheritance** makes the public and protected members of the base class protected in the derived class.

- **private inheritance** makes the public and protected members of the base class private in the derived class



State Govt.
Services



LAW
Entrance

Public Inheritance

• **Description:**

When a class inherits publicly, all the public members of the base class remain public, and the protected members of the base class remain protected in the derived class. Private members of the base class are not directly accessible.

• **Syntax:**

```
class Base { /* ... */ };
```

```
class Derived : public Base { /* ... */ };
```



MBA
Entrance



Railways & Metro
Services

...many more

abhyasonline.in



**Course
&
Test Series**

 **CBSE**

 **ICSE**

 **NTSE**

 **Banking &
Insurance**

 **Central Govt.
Service**

 **State Govt.
Services**

 **LAW
Entrance**

 **MBA
Entrance**

 **Railways & Metro
Services**

...many more

abhyasonline.in

Introduction to 'C++' Language

Rules:

- Public members of Base → Public in Derived
- Protected members of Base → Protected in Derived

Use Case:

Use public inheritance when there is a clear "is-a" relationship between the base and derived classes.

Protected Inheritance

• **Description:**

When a class inherits protectedly, the public and protected members of the base class become protected in the derived class. Private members of the base class remain inaccessible.

• **Syntax:**

```
class Base { /* ... */ };
class Derived : protected Base { /* ... */ };
```

Rules:

- Public members of Base → Protected in Derived
- Protected members of Base → Protected in Derived

Use Case:

Use protected inheritance when you want to restrict access to the base class's members in the derived class while still allowing further inheritance.

Private Inheritance

• **Description:**

When a class inherits privately, all public and protected members of the base class become private in the derived class. Private members of the base class remain inaccessible.

• **Syntax:**

```
class Base { /* ... */ };
class Derived : private Base { /* ... */ };
```

Rules:

- Public members of Base → Private in Derived
- Protected members of Base → Private in Derived

Use Case:

Use private inheritance when you want to hide the implementation details of the base class from external code while still using its functionality within the derived class.