Making a private member inheritable

- Private member of a base class cannot be inherited.
- But it is possible to do so with the help of **'protected'** access specifier.
- A member declared as **protected** is accessible by the member functions within its class and any class *immediately* derived from it.
- It cannot be accessed by any other functions outside these two classes.
- When **protected** member is inherited in **public** mode, it becomes **protected** in the **derived** class and is accessible by the member functions of the derived class, also further inheritance is possible.
- When **protected** member is inherited in **private** mode, it becomes **private** in the **derived** class and is accessible by the member functions of the derived class, but further inheritance is impossible.
- In **protected** derivation, **public** and **protected** members become **protected**

```
class alpha
{
   private:                    //optional
   …….                         //visible to member functions within its class
   ……..
   protected:
   …….                         // visible to member functions of its on and derived class
   ……..
   public:
   ……                          //visible to all functions in the program
   ……..
};
```

Access Rights of Derived Classes (or Visibility of inherited members)

Derived class visibility

Base class visibility	private	protected	public
private	-	-	-
protected	private	protected	protected
public	private	protected	public

Example program

```
#include <iostream>
using namespace std;

class Base {
private:
    int pvt = 1;

protected:
    int prot = 2;

public:
    int pub = 3;
    int getPVT()                    // function to access private member
                {
                    return pvt;
                 }
};
```

```
class ProtectedDerived : protected Base {
public:
    int getProt()                       // function to access protected member from Base
        {
          return prot;
        }

    int getPub()                        // function to access public member from Base
        {
           return pub;
        }
 };

int main()
{
  ProtectedDerived object1;
  cout << "Private cannot be accessed." << endl;
  cout << "Protected = " << object1.getProt() << endl;
  cout << "Public = " << object1.getPub() << endl;
  return 0;
}
```

OUTPUT

Private cannot be accessed.

Protected = 2

Public = 3