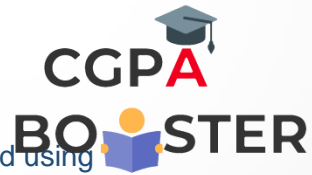


Solution Code



```
/* C++ Program to show access to Private Public and Protected using  
Inheritance */
```

```
#include<iostream>  
using namespace std;  
class Base  
{  
    private:  
        int basedata1 ;  
    protected:  
        int basedata2 ;  
    public:  
        int basedata3 ;  
    /*The member function of a class can directly access all other members (data  
+ methods) of the same class, no matter whether the data is public,  
protected or private */  
    void get_base_data()  
    {  
        cout << "\n Enter basedata1: ";  
        cin >> basedata1 ; // accessible  
        cout << "\n Enter basedata2: ";  
        cin >> basedata2 ; // accessible  
        cout << "\n Enter basedata3: ";  
        cin >> basedata3 ; // accessible  
    }  
    void display_base_data()  
    {  
        cout << "\n basedata1 = " << basedata1;  
        cout << "\n basedata2 = " << basedata2;  
        cout << "\n basedata2 = " << basedata3;  
    }  
}; // End of the class Definition
```

Solution Code



```
class Derive : public Base
{private:
    int derivedata1 ;
protected:
    int derivedata2 ;
public:
    int derivedata3 ;
void get_derive_data()
{
    cout << "\n Enter basedata1: ";
    cin >> basedata1 ; //not accessible; private data of base class
    cout << "\n Enter basedata2: ";
    cin >> basedata2 ; // accessible; protected data of base class
    cout << "\n Enter basedata3: ";
    cin >> basedata3 ; // accessible; public data of the base class

    cout << "\n Enter derivedata1: ";
    cin >> derivedata1 ; // accessible
    cout << "\n Enter derivedata2: ";
    cin >> derivedata2 ; // accessible
    cout << "\n Enter derivedata3: ";
    cin >> derivedata3 ; // accessible
}
void display_derive_data()
{
    cout << "\n basedata1 = " << basedata1;
    cout << "\n basedata2 = " << basedata2;
    cout << "\n basedata2 = " << basedata3;

    cout << "\n derivedata1 = " << derivedata1;
    cout << "\n derivedata2 = " << derivedata2;
    cout << "\n derivedata2 = " << derivedata3;
}
}; // End of the sub class definition
```

Solution Code



```
int main()
{
    Base bobj ;

    bobj.basedata1 = 10 ; // Not accessible
    bobj.basedata2 = 20 ; // Not accessible
    bobj.basedata3 = 30 ; // Accessible

    bobj.get_base_data(); // accessible
    bobj.display_base_data(); // accessible

    // Cont. on the next slide

    Derive dobj ;

    dobj.basedata1 = 5 ; // Not accessible
    dobj.basedata2 = 6 ; // Not accessible
    dobj.basedata3 = 7 ; // Accessible

    dobj.derivedata1 = 8 ; // Not accessible
    dobj.derivedata2 = 9 ; // Not accessible
    dobj.derivedata3 = 10 ; // Accessible

    dobj.get_base_data() ; // Accessible
    dobj.display_base_data() ; // Accessible
    dobj.get_derive_data(); // Accessible
    dobj.display_derive_data(); // Accessible

    return 0;
}
```

