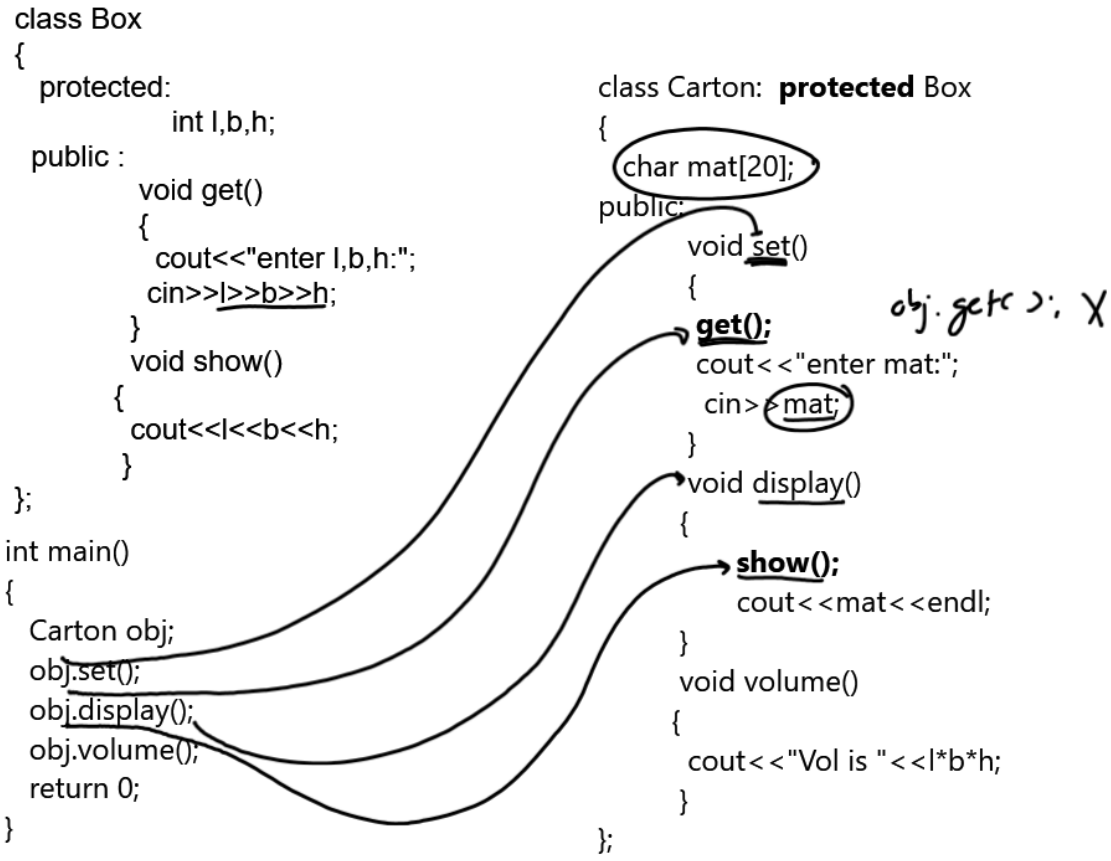
Accesibility Rules Of Base Class Members In "PROTECTED" mode of INHERITANCE
==========

1. All the **public** members of the base calss become **protected** members of the derived class . That is they can be accessed from the member functions of the derived class but not from outside the derived class.

2. All the **protected** members of the base calss become **protected** members of the derived class . That is they can be accessed from the member functions of the derived class but not from outside the derived class.

3. All the **private** members of the base class remain **private** to their own class and thus can neither be accessed by the object nor by the member functions of the derived class.



Accesibility Rules Of Base Class Members In "PRIVATE" mode of INHERITANCE
==========

1. All the **public** members of the base calss become **private** members of the derived class . That is they can be accessed from the member functions of the derived class but not from outside the derived class.

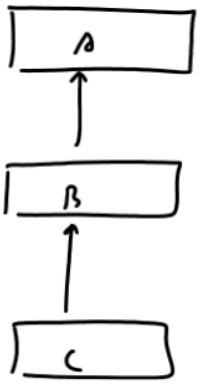
2. All the **protected** members of the base calss become **private** members of the derived class . That is they can be accessed from the member functions of the derived class but not from outside the derived class.

3. All the **private** members of the base class remain **private** to their own class and thus can neither be accessed by the object nor by the member functions of the derived class.

```
class Box
{
  protected:
              int l,b,h;
  public :
         void get()
         {
           cout<<"enter l,b,h:";
           cin>>l>>b>>h;
         }
          void show()
         {
          cout<<l<<b<<h;
          }
};
int main()
{
  Carton obj;
  obj.set();
  obj.display();
  obj.volume();
  return 0;
}
```

```
class Carton:  private Box
{
  char mat[20];
public:
        void set()
        {
        get();
         cout<<"enter mat:";
          cin>>mat;
        }
        void display()
        {
             show();
             cout<<mat<<endl;
        }
        void volume()
       {
         cout<<"Vol is "<<l*b*h;
        }
};
```

MULTILEVEL INHERITANCE




```
class Num
{
  protected:
              int a,b;
 public :
           void get()
           {
            cout<<"enter 2 int:";
            cin>>a>>b;
           }
           void show()
          {
           cout<<a<<","<<b;
          }
};

int main()
{
  DiffNum obj;
  obj.accept();
  obj.add();
  obj.diff();
  obj.print();
  return 0;
}
```

```
class AddNum: public Num
{
            int c;
 public :
          void set()
          {
           get();
          }
          void add()
          {
            c=a+b;
          }

         void display()
        {
         show();
         cout<<"Sum is "<<c;
        }
};
```

```
class DiffNum: protected
AddNum
{
   int d;
 public :
           void accept()
           {
            set();
           }
           void diff()
           { add();
             d=a-b;
           }

         void print()
        {
         display();
         cout<<"Diff is "<<d;
        }
};
```