Accessibility Rules Of Base Class Memebrs In "public" Mode Of nheritance

Whenever we inherit a class in public mode then:

1. All the public members of the bae class become public members of the derived class . That means they can be accessed via member functions of the derived class as well as by the main() function also using object of the derived class.

2. All the protected members of the base class become protected members of the derived class that means they can be accessed from the member functions of the derived class but not directly from main().

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.

Accessibility Rules Of Base Class Memebrs In "protected" Mode Of nheritance

Whenever we inherit a class in protected mode then:

1. All the public members of the bae class become protected members of the derived class . That means they can be accessed via member functions of the derived class only and not from main() using object of the derived class.

2. All the protected members of the base class become protected members of the derived class that means they can be accessed from the member functions of the derived class but not directly from main().

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
{
   int l,b,h;
public:
           void get()
           {
             cout<<"Enter l,b,h:";
             cin>>l>>b>>h;
           }
           void show()
           {
             cout<<l<<","<<b<<","<<h<<endl;
           }
};
```

```
class Carton: protected Box
{
char matr[20];
public:
           void set()
           {
             get();
             cout<<"Enter mat:";
             cin>>mat;
           }
           void display()
           {
             show();
             cout<<mat<<endl;
           }
};
```

```
int main()
{
  Carton obj;

  obj.set();


  obj.display();

  return 0;
}
```

Accessibility Rules Of Base Class Memebrs In "private" Mode Of nheritance

Whenever we inherit a class in private mode then:

1. All the public members of the base class become private members of the derived class . That means they can be accessed via member functions of the derived class only and not from main () using object of the derived class.

2. All the protected members of the base class become private members of the derived class that means they can be accessed from the member functions of the derived class but not directly from main().

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.

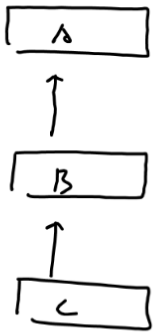
```
class Box
{
   int l,b,h;
public:
           void get()
           {
             cout<<"Enter l,b,h:";
             cin>>l>>b>>h;
           }
           void show()
           {
             cout<<l<<","<<b<<","<<h<<endl;
           }
};
```

```
class Carton: private Box
{
char mat[20];
public:
           void set()
           {
             get();
             cout<<"Enter mat:";
             cin>>mat;
           }
           void display()
           {
             show();
             cout<<mat<<endl;
           }
};
```

optional

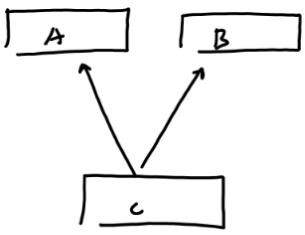
```
int main()
{
  Carton obj;

  obj.set();


  obj.display();

  return 0;
}
```

MultiLevel Inheritance



```
class Num
{
  protected:
            int a,b;
  public:
           void get()
          {
           cout<<"Enter a and b:";
           cin>>a>>b;
          }
          void show()
         {
          cout<<a<<","<<b<<endl;
         }
};
class AddNum:public Num
{
                int c;
    public:
           void set()
           {
             get();
           }
           void add()
          {
            c=a+b;
          }
     void display()
     {
       show();
       cout<<"Sum is "<<c;
     }
};
class DiffNum: public AddNum
{
   int d;
public:
         void accept()
         {
            set();
         }
         void diff()
        {
          d=a-b;
        }
        void print()
       {
         display();
         cout<<"Diff is "<<d;
       }
};
int main()
{
  DiffNum obj;
  obj.accept();
  obj.add();
  obj.diff();
  obj.print();
  return 0;
}
```

Multiple Inheritance




```
class A
{
  =
};
class B
{
  =
};
```

```
class C: public A, public B
{
  =
  =
  =
  =
};
```

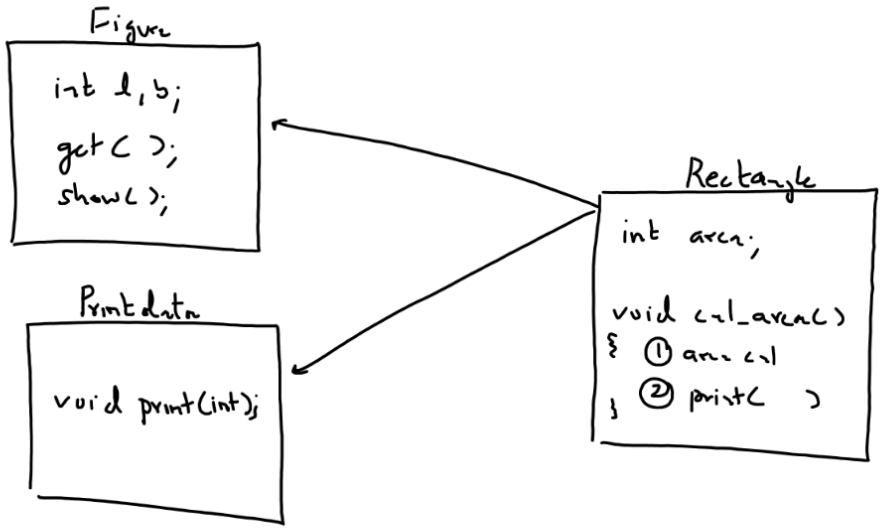


Figure
int l, b;
getC);
showC);
Rectangle
int area;
void cal_areaC)
{ ① area cal
② printC)
}
Printdata
void print(int);