

```

class A
{
    public:    int x;
    protected: int y;
    private:  int z;
};

class B : public A
{
    // x is public
    // y is protected
    // z is not accessible from B
};

class C : protected A
{
    // x is protected
    // y is protected
    // z is not accessible from C
};

class D : private A
{
    // x is private
    // y is private
    // z is not accessible from D
};
    
```

IMPORTANT NOTE: Classes B, C and D all contain the variables x, y and z. It is just question of access.

Member in base class :		Public	Protected	Private
Inheritance type : Object inherited as:				
Private	:	Private	Private	NoAccess
Protected	:	Protected	Protected	NoAccess
Public	:	Public	Protected	NoAccess

		class B : public A	class C: protected A	class D: private A
Base Class A	public : int X ;	X is public in B	X is protected in C	X is private in D
	protected : int Y ;	Y is protected in B	Y is protected in C	Y is private in D
	private : int Z;	Z is not accessible in B	Z is not accessible in C	Z is not accessible in D

1. Any private member of Base be inaccessible in Child
2. MoreRestricted(Access1 of Base member, Access2 of Base class declared in Child)
3. Not Accessible => not even compile