```cpp
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.

<table>
<thead>
<tr>
<th>Member in base class</th>
<th>Public</th>
<th>Protected</th>
<th>Private</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inheritance type: Object inherited as:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Private</td>
<td>Private</td>
<td>Private</td>
<td>NoAccess</td>
</tr>
<tr>
<td>Protected</td>
<td>Protected</td>
<td>Protected</td>
<td>NoAccess</td>
</tr>
<tr>
<td>Public</td>
<td>Public</td>
<td>Protected</td>
<td>NoAccess</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Base Class</th>
<th>public : int X;</th>
<th>class B : public A</th>
<th>X is public in B</th>
<th>X is protected in C</th>
<th>X is private in D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>protected : int Y;</td>
<td>class C : protected A</td>
<td>Y is protected in B</td>
<td>Y is protected in C</td>
<td>Y is private in D</td>
</tr>
<tr>
<td></td>
<td>private : int Z;</td>
<td>class D : private A</td>
<td>Z is not accessible in B</td>
<td>Z is not accessible in C</td>
<td>Z is not accessible in D</td>
</tr>
</tbody>
</table>

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