Access of Redeclared Member

R. Michael Anderson
Edison Design Group, Inc.
rma@edg.com
May 27, 1995

Problem

If a class member is declared more than once within a class definition — a situation that can arise with nested class and typedef declarations — the declarations are currently permitted to have different access controls. But which access applies? Here's an example:

```c
struct S {
    class A;
    class B;
private:
    class A { };
    class B;
};
S::A a;  // Is S::A accessible?
class S::B { };
S::B b;   // Is S::B accessible?
```

Proposal

Add to clause 11 [class.access] or 11.1 [class.access.spec] the following:

```c
When a member is redeclared within its class definition, the access specified at its redeclaration shall be the same as at its initial declaration. [Example:

```c
struct S {
    class A;
private:
    class A { }; // error: cannot change access
};
```

—end example]
```

Discussion

This was discussed a bit on the reflector, and there was strong support for a solution along these lines. Other options included giving priority to the defining declaration (if any) and/or to the initial declaration — but without an error when there is an inconsistency.