The Public and Private Keywords
SC22WG14 N653 — NCITS/J11 97-016

6 January 1997

Bill Seymour
wseymour@email.usps.gov

This paper proposes the addition of two new keywords, **public** and **private**, that may be used in the declaration of a structure that contains member functions. This was originally part of the Member Functions proposal (N652 / 97-015) but was removed from that paper so that it could be accepted or rejected independently. If the Member Functions proposal is not accepted, this paper will be withdrawn. The key points in this proposal are:

- Member visibility counts when determining whether two structure types are compatible.

- To say that an identifier has private visibility simply means that the identifier has block scope within a member function definition. The alternative would have been to create a new kind of "structure scope" which would have complicated the proposal unnecessarily.

- Private member objects may be initialized just like public member objects. In the absence of C++-like constructors, the only alternative would have been to disallow initialization of private members entirely.

6.1.2 Keywords

Add the two keywords, **private** and **public**.

6.1.2.1 Scopes of identifiers

Change the beginning of the first sentence in **[3]** to:

> Except for identifiers with private visibility as specified in 6.5.2.1, every other…

6.1.2.6 Compatible type and composite type

Change the second to last sentence of **[1]** from:

> For two structures, corresponding members shall be declared in the same order.

to:

> For two structures, corresponding members shall be declared in the same order
> and with the same visibility.

6.5.2.1 Structure and union specifiers

Change the grammar to:

struct-declaration-list:
struct-declaration
struct-declaration-list struct-declaration
| *visibility-specifier* **:** *struct-declaration-list*

| *visibility-specifier:*
| **private**
| **public**

And add the constraint:

A visibility-specifier shall not appear in a union specifier.

Change **[5]** from the following language adopted in the Member Functions proposal:

Structure and union specifiers have the same form, except that a structure specifier may declare member functions.

to:

Structure and union specifiers have the same form, except that a structure specifier may declare member functions and may include visibility specifiers.

Add between **[6]** and **[7]**:

Structure member identifiers whose declarations follow the **private** visibility specifier (without an intervening **public** visibility specifier) have *private visibility*. Except for the purpose of initialization, identifiers with private visibility have block scope within the outermost block of the definition of each member function of the same structure type. All other structure member identifiers have *public visibility* and scope as specified in 6.1.2.1.

Change the second sentence of **[9]** from:

If enough space remains, a bit-field that immediately follows another bit-field in a structure shall be packed into adjacent bits of the same unit.

to:

If enough space remains, a bit-field that immediately follows another bit-field in a structure, or is separated from the previous bit-field only by one or more visibility specifiers, shall be packed into adjacent bits of the same unit.

6.5.7 Initialization

The Member Functions proposal added to semantics between **[9]** and **[10]**:

> Member functions of structure objects do not participate in initialization.

Add the following sentence right before that:

> All named member objects of structure objects participate in initialization regardless of whether they have public or private visibility.