Document Numbers: X3J16/96-0150
WG21/N0968
Date: July 25, 1996
Reply To: Bill Gibbons
bill@gibbons.org

Working Paper Changes Eliminating Friend Name Injection

7.3.1.2 [namespace.memdef]

In paragraph 3, replace

> A friend function first declared within a class is a member of the innermost enclosing namespace

with:

> A friend class or function first declared within a class describes a member of the innermost enclosing namespace. Such a friend is not found by any name lookup until a matching declaration is explicitly provided in that scope (either before or after the class declaration.)

In paragraph 3, delete:

> The scope of class names first introduced in elaborated-type-specifiers is described in (3.3.1)."

3.3.1 [basic.scope.pdecl]

In paragraphs 6 and 7 , replace

> A class declared as a friend ... [Note: For point of instantiation of a template, see 14.7.1.]

with:

> Friend declarations describe functions or classes in the nearest enclosing namespace, but they do not introduce names into that namespace.

14.6.5 [temp.inject]

In paragraph 1, replace

> When a template ... -end example]

with:

> When a template is instantiated, its friends are treated as if the specialization had been explicitly declared at its point of instantiation. [7.3.1.2], [3.3.1]

Editorial suggestion: rename [temp.inject] since injection is gone.

13.3.1.1.1 [over.call.func]

In paragrah 3, after

> The name is looked up in the context of the function call following the normal rules for name lookup

add:

> ... and also looked up among the friends of any classes among the set of associated types for the arguments, using the rules for associated types as described in [over.match.oper].

13.3.1.2 [over.match.funcs]

In paragraph 5, add to list of lookups of operator@:

- For each type Z, where Z is a class type representing either T1 or T2, or a direct or indirect base class of one of these types, operator@ is looked up among the friend declarations in Z.