Bill Gibbons, John Spicer

1. Issues from 95-0183/N0783

1.1 Working paper changes for issue 1.1

Add to 7.3.1.2 [namespace.memdef] paragraph 3:

> When looking for a prior declaration of a friend, scopes outside the innermost enclosing namespace scope are not considered.

1.2 Working paper changes for issue 1.2

Add to 3.3.1 paragraph 4:

> If the elaborated type specifier has the form

```
friend class-key identifier ;
```

> the identifier is declared in the scope described above, but when looking for a prior declaration of the class, scopes outside the innermost enclosing namespace scope are not considered.

1.3 Working paper changes for issue 1.4

Add to end of the first sentence of 11.4 [class.friend] paragraph 5:

> and the function name is unqualified

1.4 Working paper changes for issue 3.1

Change 7.3.1.1 [namespace.unnamed] paragraph 1 as in the paper.

1.5 Working paper changes for issue 3.2

Add to 7.3.3 [namespace.udecl] paragraph 1:

> In a class or namespace scope, a name spacified in the using-declaration must not already be a member of that scope.

1.6 Working paper changes for issue 3.3

Add to 7.3.3 [namespace.udecl] paragraph 4:

> A using-declaration shall not refer to a destructor. [Note: Since constructors do not have names, a using-declaration cannot refer to a constructor.]

1.7 Working paper changes for issues 1.3 and 3.4

Note that the resolution of issue 1.3 was changed from what was described in the paper. A global qualifier is permitted in all declarations in which a qualified-name is permitted, not simply in friend declarations as described in the paper.

In 8.3 [dcl.meaning] change “A *declarator-id* shall not contain a *nested-name-specifier*” to “A *declarator-id* shall not be qualified”.

In 8.3 [dcl.meaning] add:

> When the declarator is qualified, the declaration must refer to a previously declared member of the class or namespace to which the qualifier refers. [Note: A global qualifier refers to the global namespace]

Modify syntax for declarator-id in clause 8 [dcl.decl] paragraph 4 as follows:

declarator-id:
$::_{opt}$ *id-expression*
$::_{opt}$ *nested-name-specifier*$_{opt}$ *type-name*

1.8 Working paper changes for issue 3.5

Add a new section after 3.4.4 [basic.lookup.classref]:

> 3.4.5 Using-directives and namespace-aliases

When looking up a *namespace-name* in a *using-directive* or *namespace-alias-definition* only namespace names are considered.