Document Numbers: X3J16/96-0157
WG21/N0975
Date: July 25, 1996
Reply To: Bill Gibbons
bill@gibbons.org

Working Paper Changes for Defaulting Inlines to Extern

3.5 [basic.link]

In paragraph 3, change:

> A name having namespace scope (3.3.5) has internal linkage if it is the name of
>
> - an object that is explicitly declared static or, is explicitly declared const and neither explicitly declared extern nor previously declared to have external linkage; or
>
> - a function that is explicitly declared static or, is explicitly declared inline and neither explicitly declared extern nor previously declared to have external linkage; or
>
> - a function template that is explicitly declared static or, is explicitly declared inline; or

to:

> A name having namespace scope (3.3.5) has internal linkage if it is the name of
>
> - an object, function or function template that is explicitly declared static; or
>
> - an object that is explicitly declared const and neither explicitly declared extern nor previously declared to have external linkage; or

7.1.1 [dcl.stc]

In paragraph 4, delete the following sentence:

> For a nonmember function, an inline specifier is equivalent to a static specifier for linkage purposes (3.5) unless the inline declaration explicitly includes extern as part of its decl-specifier or matches a previous declaration of the function, in which case the function name retains the linkage of the previous declaration.

In paragraph 7, Remove examples which use inline functions.

7.1.2.3 [dcl.fct.spec]

In paragraph 3, replace:

> For the linkage of inline functions, see 3.5 and 7.1.1.

with a footnote:

> The inline keyword has no effect on the linkage of a function.

11.4 [class.friend]

In paragraph 4, delete the note in the middle of the paragraph:

> A function first declared in a friend declaration has external linkage (3.5). ~~[Note: this is true even if the friend declaration declares the function to be inline.]~~ Otherwise, the function retains its previous linkage (7.1.1).