X3J16/95-0213
WG21/N0813
November 8, 1995
J. Stephen Adamczyk
jsa@edg.com

WP changes for (member) function typedefs

This document is a summary of WP changes related to 95-0192/N0792, *Issues Regarding Function Typedefs*, by Ben Schreiber. It includes some changes recommended by that document and others required to implement a previous committee decision that there are no "member function types."

In 5.1 [expr.prim]:

In paragraph 8, replace "If the qualified-id refers to a non-static member, its type is the data member type or function member type (9.2) ; if it refers to a static member, its type is an object or function type (9.4)." with "The type of the qualified-id is the type of the member."

In 5.2.4 [expr.ref]:

In paragraph 6, fourth bullet, replace "*cv* function of (parameter type list) returning T" with "function of (parameter type list) *cv* returning T" and replace "class X's *cv* member function of (parameter type list) returning T" with "function of (parameter type list) *cv* returning T".

In 5.3.1 [expr.unary.op]:

In paragraph 3, replace "the type ``nonstatic member function''" with "a qualified-id for a nonstatic member function".

In 8.1 [dcl.name]:

In paragraph 1, last sentence, replace "function having no parameters and returning pointer to integer" to "function of () returning pointer to int" and replace "pointer to function of double returning an integer" with "pointer to function of (double) returning int".

In 8.3.3 [dcl.mptr]:

Add to the end of the note in paragraph 3:

The type "pointer to member" is distinct from the type "pointer", and cannot be built from a combination of "pointer" and any type. That is, a pointer-to-member is declared only by the pointer-to-member declarator syntax, and never by the pointer declarator syntax.

In 8.3.5 [dcl.fct]:

In paragraph 1, change "*derived-declarator-list cv-qualifier-seq*$_{opt}$ function with parameters of type *parameter-declaration-clause* and returning T" to "*derived-declarator-list* function of (*parameter-declaration-clause*) *cv-qualifier-seq*$_{opt}$ returning T".

Delete the last two sentences of paragraph 3 and add the following as a new paragraph following paragraph 3:

A *cv-qualifier-seq* shall only be part of the function type of a nonstatic member function, or the function type to which a pointer to member function refers, or the top-level function type of a typedef declaration. Such a typedef may itself be used only in such permitted places. The *cv-qualifier-seq* is part of the function type. At all times in the determination of a type, if a type of the form "*cv-qualifier-seq* function returning T" is formed, the program is ill-formed. [Note: the

cv-qualifier-seq as part of the function type is not the same as a cv-qualifier applied to the function type itself. [Example:

```
typedef void F();
struct A {
  const F f;  // Error (not the same as void f() const;)
};
```

-- end example] -- end note]

After the last paragraph, add

[Note: a function can be declared (but not defined) using a typedef for a function type. [Example:

```
typedef void F(int);
F f;                    // same as void f(int);
```

-- end example] -- end note]

In 9.2 [class.mem]:

Add at the end of paragraph 8: "[Note: The type of a nonstatic member function is a normal function type, and the type of a nonstatic data member is a normal object type. There are no special member function types or data member types.]"

Delete paragraph 10.

In 9.3 [class.mfct]:

Add after the last paragraph:

[Note: A member function can be declared (but not defined) using a typedef for a function type. The resulting member function has exactly the same type as it would have if the function declarator were provided explicitly. [Example:

```
typedef void fv(void);
typedef void fvc(void) const;
struct S {
  fv memfunc1;
  void memfunc2(void);
  fvc memfunc3;
};
fv  S::* pmfv1 = &S::memfunc1;
fv  S::* pmfv2 = &S::memfunc2;
fvc S::* pmfv3 = &S::memfunc3;
```

-- end example] See also _temp.arg_. -- end note]

In 9.3.1 [class.mfct.nonstatic]:

Delete paragraph 3.

In 9.4 [class.static]:

Delete paragraph 5.

In 13.3.1.1.2 [over.call.object]:

In paragraph 2, replace “pointer to function with parameters of type P1, ... Pn and returning R” with “pointer to function of (P1, ..., Pn) returning R” and replace "reference to function with parameters of type P1, ..., Pn and returning R" with "reference to function of (P1, ..., Pn) returning R".