J16/97-0099
WG21/N1137
November 11, 1997
J. Stephen Adamczyk
jsa@edg.com

Core II WP Changes
(Morristown)

A. Enum overloading

1) In 13.6 [over.built], remove "or enumeration" in paragraphs 19 and 23; change "pointer type" to "pointer or enumeration type" in paragraph 16; change "pointer to member type" to "enumeration or pointer to member type" in paragraph 21.

2) In 13.3.1.2 [over.match.oper], paragraph 3, third bullet, add as fourth sub-bullet:

   ... , and

   -- do not have the same parameter type list as any non-template non-member candidate.

   In 13.6 [over.built], paragraph 2, add at the end inside the note:

   As described in _over.match.oper_, if there is a user-written candidate with the same name and parameter types as a built-in candidate operator function, the built-in operator function is hidden and is not included in the set of candidate functions.

3) In 13.3.1.2 [over.match.oper], paragraph 3, second bullet, add at the end:

   However, if no operand has a class type, only those non-member functions in the lookup set that have a first parameter of type T1 or "reference to (possibly cv-qualified) T1", or (if there is a right operand) a second parameter of type T2 or "reference to (possibly cv-qualified) T2" are candidate functions.

B. Ambiguous Conversion Sequence

1) In 13.3.3.1 [over.best.ics], paragraph 10, replace

   If several different sequences of conversions exist that each convert the argument to the parameter type, the implicit conversion sequence is a sequence among these that is not worse than all the rest according to 13.3.3.2 123). If that conversion sequence is not better than all the rest and a function that uses such an implicit conversion sequence is selected as the best viable function, then the call will be ill-formed because the conversion of one of the arguments in the call is ambiguous.

   with

   If several different sequences of conversions exist that each convert the argument to the parameter type, the implicit conversion sequence associated with the parameter is the unique conversion sequence designated the /ambiguous conversion sequence/. For the

cv-unqualified version is the same type as T or is a derived class
thereof are candidate functions.

2) In 13.3.1.5 [over.match.conv], paragraph 1, bullet 1, replace

   Those that are not hidden within S and yield type "cv2 T" or a type
   that can be converted to type "cv2 T" via a standard conversion
   sequence (_over.ics.scs_), for any cv2 that is the same cv-
   qualification as, or lesser cv-qualification than, cv1, are candi-
   date functions. Conversion functions that return a nonclass type
   "cv2 T" are considered to yield cv-unqualified T for this process of
   selecting candidate functions.

   with

   Those that are not hidden within S and yield type T or a type
   that can be converted to type T via a standard conversion
   sequence (_over.ics.scs_) are candidate functions. Conversion
   functions that return a cv-qualified type are considered to yield
   the cv-unqualified version of that type for this process of
   selecting candidate functions.

(end)

•