1 Introduction

This paper updates the proposed wording of N2581. It deletes one section as requested by the Evolution Working Group, and incorporates all changes requested by the Core Working Group.

2 Proposed wording

This section’s proposed wording is with respect to N2741. This proposal is purely an extension to N2741; except for very small additions to the underlying grammar, it requires no changes to any existing wording.

In 14.9.3 [concept.refine], augment the grammar definition of requirement-specifier; in 14.10.1 [temp.req], augment the grammar definition of requirement and add a grammar definition of concept-instance-alias-def. Text to be added is indicated in red:
refinement-specifier:

\[\text{concept-instance-alias-def} \text{opt} \mathbf{::} \text{opt} \text{nested-name-specifier} \text{opt} \text{concept-id} \]

requirement:

\[\text{concept-instance-alias-def} \text{opt} \mathbf{::} \text{opt} \text{nested-name-specifier} \text{opt} \text{concept-id} \]

\(! \mathbf{::} \text{opt} \text{nested-name-specifier} \text{opt} \text{concept-id}\]

concept-instance-alias-def:

\[\text{identifier} = \]

Append the following new paragraph to 3.3.1 [basic.scope.pdecl]:

The point of declaration for the \textit{identifier} in a \textit{concept-instance-alias-def} is immediately after the \textit{concept-id} of its \textit{requirement} or \textit{refinement-specifier}.

Append the following after 14.10.1 [temp.req] p5 (“A negative requirement requires...”):

A \textit{concept-instance-alias-def} defines its \textit{identifier} to be an alias of the concept instance given in its \textit{requirement} or \textit{refinement-specifier}. When the \textit{concept-instance-alias-def} appears in a \textit{member-requirement} (9.2), the potential scope of the \textit{identifier} begins at its point of declaration and terminates at the end of the constrained member’s declaration. When the \textit{concept-instance-alias-def} appears in the optional \textit{requires-clause} of an \textit{axiom-definition} (14.9.1.4), the potential scope of the \textit{identifier} begins at its point of declaration and terminates at the end of the \textit{axiom-definition}. Otherwise, a \textit{concept-instance-alias-def} inserts the \textit{identifier} as a name in the scope of:

- the template parameters of the concept, when the \textit{concept-instance-alias-def} appears in a \textit{refinement-specifier} (14.9.3);
- the enclosing concept, when the \textit{concept-instance-alias-def} appears in an \textit{associated-requirement} (14.9.1.3); or
- the template parameters declared in the \textit{template-parameter-list} immediately before the \textit{requires} keyword, when the \textit{concept-instance-alias-def} appears in the optional \textit{requires-clause} of a \textit{template-declaration}.

| Example: |

```cpp
concept A<typename X, typename Y, typename Z> {
    typename result_type;
}

concept B<typename X, typename Y> {
    typename result_type;
}

concept C<typename T> {
    typename R;
}

template< typename T >
    requires J = C<T>
```
15 J::R f(T);
16 // qualified lookup finds type name R within the concept C (3.4.3.3)
18 auto concept D<typename Op, typename Elem> {
19   requires a = A<Op, Elem, Elem>;
20   requires B<a::result_type, Elem>;
21   typename result_type = a::result_type;
22 }

—end example ]

If a concept-instance-alias-def appears in a requirement that is the pattern of a pack expansion, the program is ill-formed. [ Example:

1 concept C<typename ... Ts> {}
3 template<typename ... Ts>
4   requires a = C<Ts>... // error: requirement aliases may refer only
5     // to requirements that are not pack expansions
7 void
8   f(Ts...);

—end example ]

3 Acknowledgments

We thank the Fermi National Accelerator Laboratory’s Computing Division, sponsor of Fermilab’s participation in the C++ standards effort, for its past and continuing support of efforts to improve C++ for all our user communities.