Wording for `std::numeric_limits<T>::lowest()`

Document #:	WG21/N2348 = J16/07-0208
Date:	2007-07-18
Revises:	None
Project:	Programming Language C++
Reference:	ISO/IEC IS 14882:2003(E)

Reply to: Marc Paterno<paterno@fnal.gov>
IDS Dept., Computing Division
Fermi National Accelerator Laboratory
Batavia, IL 60510-0500

1 Introduction

Document N1880 presents the case for adding a member function `lowest` to `std::numeric_limits`. This function is intended to return the smallest finite value (in the mathematical sense, that x is smaller than y if $x < y$).

At Mont Tremblant, Pete Becker noted that the wording was flawed, because not all floating-point representations have the feature that the most negative representable finite value is the negative of the most positive representable finite value.

This paper provides alternative wording, to fix this flaw.

This document is based on the current (as of this writing) working paper, N2315.

2 Proposed wording

2.1 Addition to [numeric.limits]

Immediately after

```
static constexpr T max() throw();
```

add

```
static constexpr T lowest() throw();
```

2.2 Addition to [numeric.limits.members]

Immediately after the description of `max()`, add:

```
static constexpr T lowest() throw();

A finite value x such that there is no other finite value y where y<x.

Meaningful for all specializations in which is_bounded != false.
```

2.3 Additional footnote

Add a footnote to “A finite value . . . y < x.”:

> lowest() is necessary because not all floating-point representations have a smallest (most negative) finite value that is the negative of the largest (most positive) finite value.

3 Acknowledgments

Fernando Cacciola, the author of N1880, did the real work in noticing the need for, and providing the motivation for, this proposal.

I thank the Fermi National Accelerator Laboratory’s Computing Division, sponsor of my participation in the C++ standards effort, for its past and continuing support.