Proposal for C2y WG14 N3259	
Title:	Support ++ and on complex values
Author, affiliation:	Aaron Ballman, Intel
Date:	2024-05-07
Proposal category:	New features
Target audience:	Developers
Date: Proposal category: Target audience:	2024-05-07 New features Developers

**Abstract:** Proposed relaxing a constraint requiring real types for the ++ and -- operators to instead require an arithmetic type.

**Prior art:** Clang, GCC, ICC, IBM XL

# Support ++ and -- on complex values

Reply-to: Aaron Ballman (aaron@aaronballman.com) Document No: N3259 Date: 2024-05-07

#### Summary of Changes

N3259

• Original proposal

### Introduction and Rationale

Complex types have been supported in C since C99 and support the full range of expected mathematical operations, including addition, subtraction, multiplication, et al. However, there is one notable exception to this support: increment and decrement operators require their operand to be a real or pointer type, which excludes complex types.

Given that ++E is defined to be equivalent to (E += 1), which is supported for complex types (and similar for --E), it stands to reason that it is safe to increment or decrement a complex value directly.

This functionality is supported as an extension in every C compiler supporting complex types on Compiler Explorer: Clang, GCC, ICC, and CL430. The IBM XL compilers all (including for non-IEEE formats) support increment and decrement as well. Other compilers that were available were tested, but do not support complex types.

There is one minor difference between postfix operators and the compound assignment operators in that the previous value needs to be held somewhere for a postfix operator. However, all implementations supporting this extension get the semantics correct based on empirical testing, so this is not expected to be a significant impediment for an implementation intending to support complex types.

#### Proposal

This paper proposes allowing a complex type as the operand to the increment and decrement operators.

C23	C2y
<pre>#include <complex.h></complex.h></pre>	<pre>#include <complex.h></complex.h></pre>
<pre>void func() {   complex float f = 1.0f;   ++f; // constraint violation  f; // constraint violation   f++; // constraint violation   f; // constraint violation }</pre>	<pre>void func() {    complex float f = 1.0f;    ++f; // Okay    assert(creal(f) == 2.0f);   f; // Okay    assert(creal(f) == 1.0f);    f++; // Okay    assert(creal(f) == 2.0f);    f; // Okay    assert(creal(f) == 1.0f); }</pre>

# Proposed Wording

The wording proposed is a diff from WG14 N3220. Green text is new text, while red text is deleted text.

Modify 6.5.3.5p1:

The operand of the postfix increment or decrement operator shall have atomic, qualified, or unqualified real-arithmetic or pointer type, and shall be a modifiable lvalue.

Modify 6.5.4.1p1:

The operand of the prefix increment or decrement operator shall have atomic, qualified, or unqualified real arithmetic or pointer type, and shall be a modifiable lvalue.

## Acknowledgements

I would like to recognize the following people for their help in this work: Rajan Bhakta and Jens Gustedt