Member access of an incomplete struct
WG14 N3532

Title:	Member access of an incomplete struct
Author, affiliation:	Aaron Ballman, Intel
Date:	2025-04-21
Proposal category:	Issues
Target audience:	WG14 members

Abstract: The C standard is currently unclear on whether the LHS of a . or -> operator is required to be a complete type or not. This clarifies that a complete type is required.

Member access of an incomplete object

Reply-to: Aaron Ballman (aaron@aaronballman.com)
Document No: N3532
Date: 2025-04-21

Summary of Changes

N3532

- Initial version

Introduction and Rationale

Consider the following code:

```
struct A {
  int a;
  int x : (sizeof((struct A*)0)->a * 8);
};
```

Does this code violate any constraints because of the use of `A` before it is complete? This is the subject of an issue raised to Clang in https://github.com/llvm/llvm-project/issues/9843.

The standard implies that this code is strict conforming. 6.5.3.4p2 says: The first operand of the `->` operator shall have type "pointer to atomic, qualified, or unqualified structure" or "pointer to atomic, qualified, or unqualified union", and the second operand shall name a member of the type pointed to. The wording for the `.` operator is similar. In either case, the first operand is a structure type and the second operand names a member of the structure type. However, no C compiler found will accept this code because the structure type is incomplete: https://godbolt.org/z/o677Tb1nh

This paper proposed clarifying the wording to match existing practice.

Proposed Wording

The wording proposed is a diff from WG14 N3467 applied. Green text is new text, while ~~red~~ text is deleted text.

Modify 6.5.3.4p1:
The first operand of the `.` operator shall have an atomic, qualified, or unqualified complete structure or union type, and the second operand shall name a member of that type.

Modify 6.5.3.4.p2:
The first operand of the `->` operator shall be a pointer to an ~~have type "pointer to~~ atomic, qualified, or unqualified complete structure or union type~~" or "pointer to atomic, qualified, or unqualified union",~~ and the second operand shall name a member of the type pointed to.