

**Proposal for C2x
WG14 N3102**

Title: US30-115 on declarators
Author, affiliation: Aaron Ballman, Intel
Date: 2023-02-02
Proposal category: Proposed NB Comment Response
Abstract: Clarifies what declarators actually do.

US30-115 on declarators

Reply-to: Aaron Ballman (aaron@aaronballman.com)

Document No: N3102

Date: 2023-02-02

Summary of Changes

N3102

- Original proposal

Introduction and Rationale

From US30-115:

“... and asserts that when an operand of the same form as the declarator appears in an expression, it designates a function or object with the scope, storage duration, and type indicated by the declaration specifiers.” Consider:

```
static int *g;
void f() {
    int x;
    g=&x;
    // ...
}
```

How does the object `g` designate a scope indicated by the declaration specifiers? Declaration specifiers don't indicate a scope – the lexical location of the declarator indicates the scope.

Also, the declarator is `*g` and not `g`; `*g` does not have static storage duration, `g` does.

I think we can find a more clear way to specify that a declarator declares one identifier and that identifier has properties from the declaration specifiers associated with it.

Proposed Straw Poll

Does WG14 want to resolve US30-115 by adopting the proposed wording from N3102?

Proposed Wording

All proposed wording in this document is a diff from WG14 N3054. **Green** text is new text, while **red** text is deleted text.

Modify 6.7.6p2:

Each declarator declares **an identifier for a single variable, function, or type, within a declaration. The preceding specifiers indicate the type, storage class, or other properties of the identifier or identifiers being declared. Each declarator specifies one declaration and names it and/or modifies the type of the specifiers with operators such as * (pointer to) and () (function returning).** ~~one identifier, and asserts that when an operand of the same form as the declarator appears in an expression, it designates a function or object with the scope, storage duration, and type indicated by the declaration specifiers.~~