

C9X Revision Proposal

Title: Nonzero Default Initial Values

Author: David Keaton

Author Affiliation: Keaton Consulting

Postal Address: 1630 30th St. No. 311, Boulder, CO 80301 USA

E-mail Address: dmk@dmk.com

Telephone Number: +1 500 FIND DMK

Fax Number:

Sponsor:

Date: 1995-05-29

Proposal Category:

- ☐ Editorial change/non-normative contribution
- ☐ Correction
- ☒ New feature
- ☐ Addition to obsolescent feature list
- ☐ Addition to Future Directions
- ☐ Other (please specify) _____

Area of Standard Affected:

- ☐ Environment
- ☒ Language
- ☐ Preprocessor
- ☐ Library
 - ☐ Macro/typedef/tag name
 - ☐ Function
 - ☐ Header

Prior Art: C++

Target Audience:

Those who would like part of the C++ constructor functionality but without the run time overhead. Users wishing to shorten the repetitive portions of Compound Literals.

Related Documents (if any):

C++ Draft Standard.

Classes in C -- Part 1: Basic Classes, C9X Proposal, Jervis.

(SC22WG14.1168) *Thoughts on Mining C++ Features*, Jaeschke.

Designated Initializers, C9X Proposal, Prosser & Keaton.

Compound Literals, C9X Proposal, Prosser & Keaton.

Proposal Attached:

- ☐ Yes
- ☒ No, but what's your interest?

Abstract

One of the uses of C++ constructors is to supply default values for some members, and allow other members to be initialized via arguments to the constructor. The static portion of this functionality can be emulated in C without the run time overhead by providing a mechanism to specify a nonzero

default initial value on a per-type basis. Then at each variable definition, any member values that are desired to be different from the default can be filled in using Designated Initializers.

In addition, nonzero default initial values could be used to shorten the specification of Compound Literals for a given type, with only the nondefault members being filled in via Designated Initializers.

Note that there are other uses for C++ constructors as well; these are not addressed here.

However, nonzero default initial values may relieve some of the pressure to implement full-blown constructors in C, with their attendant run time overhead.

Proposal

The author will monitor interest in this subject before providing a more detailed proposal.