

Wording for New Bit-field Syntax

Document No.: D0187R1

Project: Programming Language C++ - Evolution

Author: Andrew Tomazos <andrewtomazos@gmail.com>

Date: 2016-06-28

Summary

We propose a new syntax for bit-fields that allows them to have default member initializers.

The consensus after discussing P0187R0 at EWG Oulu was to “add a new syntax for being able to provide both a bitfield-width and an initializer”, with 21 for and 0 against.

Further to this, we have designed a new syntax that is simple, easy to teach, requires no disambiguation rules, is easy to parse and requires only a one line addition to the grammar.

Example

```
struct S {  
    int name : width;  
  
    int name : width : = init;  
  
    int name : width : { init };  
};
```

Explanation

The syntax can be taught as follows: “To use a default member initializer for a bit-field, separate the initializer from the bit-field width with a second colon.”

Background

For background and motivation on the problem we are solving see P0187R0.

Wording

Add to grammar:

member-declarator:

declarator virt-specifier-seq_{opt} pure-specifier_{opt}

declarator brace-or-equal-initializer_{opt}

identifier_{opt} attribute-specifier-seq_{opt}: constant-expression

*identifier attribute-specifier-seq_{opt}: *

constant-expression : brace-or-equal-initializer

Modify [class.bit]:

A *member-declarator* of **one of the forms:**

identifier_{opt} attribute-specifier-seq_{opt}: constant-expression

*identifier attribute-specifier-seq_{opt}: *

constant-expression : brace-or-equal-initializer

specifies a bit-field; [...]