Editor's Report

• X3J16/WG21 March 1992

Jonathan E. Shapiro
shopiro@usl.com

Editorial policy (same as last time)

Much of the RM needs improvement

• Resolutions of the committee (if any) are incorporated into the working paper
• Messages (preferably email) to the editor
  • Typos and thinkos are fixed with thanks
  • Substantive or controversial issues are referred to the appropriate subcommittee
• Issues brought up on the reflectors that seem clearly correct and non-controversial are incorporated
• The committee can undo any of the editor's decisions at the next meeting
C Compatibility

- Chapter 16, "Preprocessing Directives"
- Taken from C Standard
- Modified appropriately for C++
- Chapter 2, "Lexical Conventions"
- Phases of translation

Digraphs and New Keywords

- Appear in Chapter 2
- Distinguished from originals (in separate lists)
Other changes

- Basic concepts
  - Added "function prototype scope" for argument names in function declarations.
  - Make it explicit that `exit()` doesn't destroy any local variables.
- Standard conversions
  - Tighten up the discussion of pointer conversions
- A few syntax fixes

Expressions

- Tighten up the function-style cast description
- Fix explanation of allocation of zero-length arrays.
- Arithmetic on pointers to functions prohibited.
- Fix conditional expression (allowing pointer-to-member as first expression, and throw expression as second or third).
Chapter changes

• Statements
  • Allow pointer-to-member in if, etc. statements
• Declarations
  • Clean up description of enumerations (they are no longer integral types).
• Declarators
  • Included Scott Turner's pointer to pointer to const example

More chapter changes

• Classes, derived classes, and access control
  • No significant changes
• Special member functions
  • Prohibited X::operator X&()
  • Added base classes to memberwise initialization (virtual base classes initialized once) What about assignment?
• Improve description of default assignment and construction
Still more chapter changes

• Overloading
  • Subsequences are always preferred.
  • Fix operator ++ and -- description

• Templates
  • A template argument is taken as a type argument rather than an arg declaration.

• Preprocessing
  • Replaced the chapter with the chapter from the C Standard (almost).