Document Number: X3J16/92-0129
WG21/N0206
Date: 15 January, 1993
Project: Programming Language C++
Reply to: Dan Saks
dsaks@wittenberg.edu

X3J16 Meeting No. 10
WG21 Meeting No. 5
November 2-6, 1992

Boston Park Plaza Hotel
64 Arlington St.
Boston, MA 02116 USA

1 Opening activities

Lenkov convened the meeting as chair at 9:00 (EST) on Monday, 2 November, 1992. Clamage was the vice-chair, and Saks was the secretary.

OSF, represented by Johnson, hosted the meeting.

1.1 Opening comments

1.2 Introductions

1.3 Membership, voting rights, and procedures for the meeting

Clamage asked members to notify him of corrections to the membership list. Saks circulated an attendance list each day, which is attached as Appendix A of these minutes.

Lenkov reminded the attendees that this is a joint meeting of WG21 and X3J16. (The joint committee is denoted WG21+X3J16 in these minutes.) In straw votes, all WG21 technical experts may vote, even if this is the first meeting they've attended; however, X3J16 attendees can vote only if they are the voting representative of a member organization that has been represented at either of the previous two meetings. (The voting representative is the principal member, or an alternate if the principal is not present.) In WG21 formal votes, only the head of each national delegation may vote. In X3J16 formal votes, only one representative from each X3J16 member organization may vote if the organization meets the aforementioned attendance requirement.

1.4 Distribution of position papers, subgroup deliverables, and other documents not distributed before the meeting

1.5 Approval of the minutes from the previous meeting

Saks submitted the minutes from the previous meeting (92-0078 = N0155) for approval.

Steinmuller said he recalled that there was a vote on dynamic array classes at the previous meeting that was not recorded in the minutes. He asked that it be included. Saks asked Steinmuller if he recalled the specifics of the vote. Steinmuller did not. Hartinger said he also recalled a vote on dynamic arrays, but he did not recall the specifics either.

Steinmuller agreed to note his disagreement in these minutes and hold the vote (again) at some future meeting.

Motion by Saks/Schwarz:

> Move we approve 92-0078 = N0155 as the minutes of the previous meeting.

Motion passed X3J16: lots yes, 0 no.
Motion passed WG21: 5 yes, 0 no.

Action item: Saks will post the Toronto meeting minutes to the editorial reflector, and post a note to the all reflector announcing the posting.

1.6 Agenda review and approval

Lenkov submitted the proposed agenda (92-0103 = N0180) for approval, and suggested adding these items:

- 1.9 International fees
- 1.10 Two-week rule
- 1.11 WG11 questionnaire on cross-language standards
- 1.12 Ad-Hoc character set meeting
- 3.1 Rationale
- 9.1 National delegation caucuses (at 13:30pm Wednesday)
- 13.4 ISO schedule for the C++ standard

Saks asked if there should be an item 13.1 for New Business. No one said yes.

Saks asked that the committee conduct the general sessions (agenda items 10, 11 and 12) in three passes:
1. for presentation of issues where the presenter intends to request a formal vote.
2. for presentation of issues leading to straw votes.
3. informational presentations.

He added that a subgroup could skip any pass for which it had no presentations.

Swan suggested combining the informational presentations with the straw votes. Saks adamantly opposed that suggestion. He felt that committee time had been doled out unfairly at past meetings. Schwarz thought two passes were adequate. Plauger strongly supported Saks, not only out of fairness, but also out of a greater sense of urgency. He thought we should be more orderly about getting things done. Plum also agreed with Saks, adding that if requiring three passes is too bureaucratic, we can always change it.

Straw vote: Who favors holding the general sessions in these three passes? lots yes, 3 no, 3 abstain.

Saks explained that there is a measure of good faith here: you may start a presentation expecting to recommend a formal vote, only to realize you're not "ready for prime time".

[Secretary's note: Apparently, despite the discussion, members were still unclear whether technical discussions belonged in pass 2 or 3.]

Motion by Saks/Bruns: "Move that we accept the proposed agenda with these additions."

Motion passed X3J16: lots yes, 0 no, 1 abstain.
Motion passed WG21: 5 yes, 0 no, 0 abstain.

1.7 Report on the previous evening's WG21 meeting

Carter presented six items from the WG21 meeting:

1. Language Independence. WG21 must respond to WG11's questionnaire on cross-language standards (see agenda item 1.11)

2. Interpretations. Carter explained that SC22 now has procedures for handling defects in approved standards.

3. Internationalization. Carter explained that WG20 had trouble starting up, but those problems seem to be passed. WG21+X3J16 members who asked to be added to WG20's mail reflector were added. Carter said that others who want to get WG20 mail (electronic or postal) should contact him.

4. Character Sets (see agenda item 1.12)

5. C++ Project Schedule (see the following discussion and agenda item 13.4)

6. Convener Resignation. Carter explained that he must resign as WG21 convener. He has his employer's approval to continue for another six months, and plans to attend the next meeting. Sam Harbison of Tartan Labs has offered to be the new convener.

Carter said he posted his projected C++ standardization schedule on the "all" reflector, and a printed copy will be distributed at this meeting (N0190 = 92-0113).

Schwarz asked Carter how he arrived at this schedule. He also asked if the committee must formally approve it.

Carter explained that the current project schedule says the CD (committee draft) will be ready in December 1993. Carter said Lenkov advised him that the committee is probably three to six months behind that schedule. So Carter suggested extending the CD date to July 1994. He

said he consulted with the SC22 secretariat about the length of the approval process. If everything goes smoothly -- we make every ballot as soon as possible, we don't have to reballot due to a number of objections from other countries, and we don't need to rewrite much of the document for technical or normative reasons -- we would have an IS (international standard) by June 1996. If we miss the July 1994 target by three months, then we would miss the June 1996 target by at least three months.

Carter said this committee needs to endorse the schedule, not for SC22 or JTC1, but as an acknowledgment of the urgency of getting the standard out by June 1996. He added that, in the past, JTC1 has been tolerant of WGs that missed their schedule commitments. They may not be as tolerant in the future.

Schwarz expressed concern that each public review lasted only three months. Plum said that as a type I project we had to follow ISO rules, and they call for two-month reviews. He suggested appealing to X3 if we want to change it.

Plauger said it's important to realize that there are people at the U.N. level that take this work VERY seriously, much more seriously than you might expect. These schedules are important. We have to publish a schedule, and we have to stick to it because a lot of people depend on it. There's serious money at stake.

Saks asked Carter and Plauger what they meant when they said it's really important to meet the June 1996 date. If we say that we can get it done by July 1998, is there a risk that SC22 would say "forget it"?

Plauger said think about this as if you had a big bank loan. The first thing you need to do is set up a realistic repayment schedule. If you give a different story every six months, that's not good. If you suddenly renege, that's a lot worse. We need to give an honest estimate.

Koenig said we need to reaffirm what we are trying to accomplish. We must look at what has happened so far in the light of what we are trying to do, then see if we can do this in a way that ISO will find acceptable.

Several members suggested that Carter's schedule is wrong. Plum said he was confused by the discussion. He said people seem to be suggesting that Carter is imposing an aggressive schedule on us. In fact, he's slipping it by six months.

Plauger said we need to reevaluate the schedule at this meeting, and at each subsequent meeting.

Becker said he saw the situation as if he was working on a project and his management said "we need money." When this happens, you make the necessary trade-offs to get the job done. He suggested voting on the schedule after the subgroups presented all of their formal votes.

Straw Vote: Who wants to move agenda item 13.4 to after first pass through general session? lots yes, 0 no, 2 abstain.

Saks explained that WG21 selects a drafting committee for each meeting. Each motion must be presented in writing to the drafting committee for approval and possible rewording before the committee can vote on it. This insures that members whose native language is not English (and even those whose native language is English) have an opportunity to read the motion and understand it before voting. The drafting committee for this meeting was Saks and Shopiro.

Saks asked that each subgroup delegate a member to bring the subgroup's motion(s) to the drafting committee. The drafting committee will meet at the end of Thursday's session to finalize the wording of formal motions to be presented Friday.

1.8 Liaison reports

=== WG14 (ISO C) ===

Plauger reported that WG14 has not met since the last WG21+X3J16 meeting. WG14's next meeting will be in Washington, DC in December. They will try to vote in the normative addendum. Plauger said that after WG14 approves the addendum, he will ask WG21+X3J16 for committee time to bring C++ in synch with ISO C.

WG14 must also respond to WG11's questionnaire. Plauger suggested that it might be good idea for the two committees to least compare notes and maybe combine responses.

Plauger also reported that Simonsen would be happy if someone replaced him as liaison to WG14. Plum offered. (He is already liaison from WG14.)

=== WG20 (Internationalization) ===

Plauger reported that WG20 met last month in Quebec, Canada. They are preparing a work item. One of their first documents will be a technical report on what they think they can do. They may circulate a questionnaire to all affected technical committees as did WG11.

Plauger said that at the last WG21+X3J16 meeting, he expressed concern that WG20 would simply rubber-stamp WG15's (POSIX) approach to locales. Plauger wrote a letter to WG20 expressing WG21+X3J16's concern that WG20 not do anything contrary to the technology standardized in C and adopted by C++.

Plauger said he will not be able to attend future WG20 meetings. Their next meeting will be in Copenhagen in December. Simonsen is our official liaison, and he intends to continue.

1.9 International Participation Fees

Lenkov explained that JTC1 changed the way it funds certain activities. ANSI assumed responsibility for funding some JTC1 groups. ANSI decided to meet its budget shortfall by billing everyone involved in ANSI activities. Lenkov described some of the ways that other committees and individuals have protested the fee.

Price noted that X3J16 members can apply for fee waivers.

1.10 Two-week rule

Lenkov said that X3's procedures (X3/SD-2) are not explicit about the meaning of the two-week rule. He said each committee decides if documents must be mailed two weeks in advance of a meeting or received two weeks in advance. Carter added that ISO has no two-week rule; they only have a rule about when meeting agenda must be distributed.

Lenkov said that if a document received two weeks in advance of a meeting was modified at that meeting, the committee can it still vote on it. It's up to committee. Any member who thinks the change was too significant can move to table the proposal, but the committee can defeat such a motion.

Pennello observed that, in effect, there is no two-week veto on a document that was mailed and then modified. Noting that we have had problems with mail delays, he suggested establishing rules that allow for such delays.

Plum recalled suggesting that if the documents are postmarked by three weeks prior to the meeting, then this satisfies the two-week rule, even if postal strikes, bad weather, or sloppy mailrooms delay delivery. No one objected to this understanding.

1.11 WG11 Questionnaire on Cross-Language Standards

Carter explained that WG11 has eight cross-language (language independent) standards pending. SC22 wants each programming language standards committee to review these documents to assess their impact on its own work. SC22 wants feedback to decide if the cross-language work is worth doing.

With this in mind, WG11 distributed a questionnaire on cross-language standards. Carter asked for volunteers to read one or more of the pending standards and answer questions with respect to that standard's impact on C++. The deadline for the answering questions is Feb. 28, 1993. The replies are understood to be the position of the convener, not the position of the committee (there's not enough time for a whole committee to meet and discuss all of these). The SC22 ad hoc group on cross-language standards will write a report on the results in May and forward it to SC22.

Colvin and McLay volunteered to help Carter.

1.12 Ad-Hoc Character Set Meeting

Carter reported that the ISO 10646 is now a formal character set standard. It specifies multiple conformance levels. He said the lowest level is acceptable for the purposes of programming languages such as C++, but the higher ones are not. SC22 needs to assess 10646's impact on programming languages. SC22 might set a policy that all programming languages will conform to level 1 of 10646, and that might be the end of it. This might surprise the framers of 10646, because they might be expecting that programming languages must conform to more of the levels.

Carter explained that SC22's ad hoc group on character sets will meet in Copenhagen in April 1993. He hoped WG21+X3J16 will send a representative to the meeting who will study the effects of 10646 on C++ and keep us informed about character set issues. We need at least one, if not more, volunteers.

Shopiro asked if we need a copy of 10646. Carter suggested that the people who attend the meeting should hold a technical session when they return.

Plauger said that sending someone from WG21+X3J16 to this meeting is not an option. They should go to get educated, and report back to us. There isn't anything that can be done about 10646. SC22 is adamant that there's to be no more criticism. It doesn't do us any good to talk about how much of 10646 we will adopt; we must adopt it all. We must look upon this as a mandatory education.

Allison and Colvin offered to represent WG21+X3J16 at the character set ad hoc group meeting.

2 Subgroup reports

Lenkov asked each subgroup to include time in their work for this week to think about schedules.

Lenkov opened the committee of whole.

[The following subgroup presentations are listed in the order they appeared in the agenda, which is not chronological order.]

2.1 Core Language Subgroup

Koenig said he hoped to wrap up name lookup so we can have formal report to vote on next meeting. He said he needs a volunteer to write that report. He said the subgroup also needs to discuss how to improve their progress.

Schwarz noted that the was substantial discussion of lifetime of temporaries at the last meeting. Four or five people volunteered to write papers, but none appeared to be in the mailings. Koenig and Lenkov said there's no way to make people do them.

Schwarz urged people to be honest with themselves, and not volunteer for more than they can do. He said we put too much pressure on people to volunteer, and not enough pressure to work.

The committee discussed ways of motivating people. Becker said (as someone who did not write a promised paper) a gentle reminder might have helped. He and Eckel stressed the need for positive, not negative, feedback as motivation.

Charney asked if the subgroup would discuss his paper on access control. Koenig said the subgroup needed to clear up name lookup. Schwarz thought lifetime of temporaries was more important.

2.2 Extensions Subgroup

Knuttila said the proposal to add operators new and delete for arrays (N0170 = 92-0093) may come up for a formal vote. The subgroup will also work on run-time type identification, cast notations, namespaces, and *~const*. He also listed other proposals the subgroup may have time to consider:

-- indirect/evolvable classes
-- extended character sets
-- exponentiation operator
-- forward declaration of nested classes
-- operator .()
-- member initializers within class declaration
-- restricted pointers
-- further relaxation of virtual return type
-- further template refinements

Vilot wanted the Extensions subgroup to address those things that impact libraries, like namespaces. Clamage thought the *operator.()* had already been considered and rejected. Knuttila said this is a different proposal. Shopiro asked if, given the pressure from schedules, work on some proposals should be abandoned.

2.3 Libraries Subgroup

Vilot said the subgroup will submit their proposals for the input/output (iostreams) library and language support library for formal votes. He noted that Schwarz prepared a 15-page appendix to the iostreams proposal (N0193 = 92-0116) listing all the functions that are either in the proposal or in the AT&T 2.1 library. The appendix compares the libraries and explains the differences.

2.4 Environments Subgroup

Stone reported that Chapin is the subgroup chair, but won't be at the meeting. He said that Price had done more work specifying the order of static initialization using the *after* keyword, and that he himself made progress on translation limits. He planned to present his proposal on translation limits for a straw vote.

Stone said the subgroup's primary work items remain: the order of static initialization and translation limits. He said that Chapin plans to present a paper on the one-definition rule at the March meeting.

Stone mentioned hosted vs. freestanding environments, but said the subgroup had not done much on it. Saks suggested looking at environments section of the C standard, at least to identify topics they should be considering. Vilot asked about progress on mixed C and C++ environments. Stone said there was none.

Koch offered to chair the subgroup at this meeting. Saks noted that Chapin has had problems attending meetings, and asked Koch to consider being permanent subgroup chair.

2.5 Formal Syntax Subgroup

Roskind said the subgroup will present two proposals for a formal vote: Roskind's proposal to eliminate context sensitive elements in the grammar (N0173 = 92-0096), and Krohn's paper on qualified class names (N0175 = 92-0098).

2.6 C Compatibility Subgroup

Plum said the subgroup will submit their "impressionistic list of differences" (N0174 = 92-0097) for formal approval as a non-normative appendix to the Working Paper, replacing the current section 19.2. The list is still "impressionistic" because the Working Paper still does not incorporate all the terminology needed to determine if they are indeed incompatibilities. He will also ask to committee to approve his proposed layout-compatibility rules appearing on pages 28 and 29 of the Toronto meeting minutes (N0155 = 92-0018).

3 Working Paper for Draft Proposed Standard

Shopiro presented the editor's report (N0192 = 92-0115) for the September '92 Working Paper (92-0091 = N0168). He explained that pages 9-2 and 9-3 were missing from the pre-meeting mailing, as was the library section. The missing pages will be distributed at this meeting.

Shopiro explained that he changed all occurrences of "illegal" in the Working Paper to "ill formed". He hoped this would help illuminate the difference between compile-time and run-time errors. He asked for guidance.

Plum suggested listing required diagnostics in separate section of the draft. He argued that many requirements for diagnostics are buried inside words, like "must" and "only". Somebody needs to go through the draft sentence-by-sentence to determine the diagnostic requirements.

Schwarz noted there's a difference of opinion about whether certain diagnostics must be issued at compile time or may be issued at run time.

Price asked about errors that occur at link time. Shopiro said he intended "ill-formed" to refer to all errors that can be detected before the program is run (including link time). Even errors that might not be detected, such as violations of the one-definition rule, would be described as ill-formed.

Charney advocated using words like "undefined" and "indeterminate." Several members disagreed.

Action item: Plum will produce a list of erroneous C++ constructs requiring diagnostics.

Shopiro restated his intent to use "ill-formed" in the draft. He said that some programs have errors that can be determined by inspection (compile-time errors). Others are run-time errors. He wants to use "ill formed" to explain rules that are compile time errors. No one objected.

Saks asked if Plum's list would distinguish diagnostics that must be issued at compile time from those that may be deferred to run time. Plum said he'd compile the list and then decide if distinctions should be made.

Saks noted that a problem in the grammar for brace initializers, previously reported by O'Riordan at an earlier meeting, was not fixed.

3.1 Rationale

Waggoner explained that she created a troff superstructure for the rationale by copying the structure of the Working Paper. She started compiling a list of changes that the committee has made to C++. She took a stab at writing down the committee's goals, based on committee papers from 1990.

Waggoner reported that she has resigned from US West and will not likely be at any future meetings. Therefore, we need a new editor for the rationale. She estimated that it might occupy as much as 25% of the editor's time. No one volunteered.

Straw Vote: Who thinks we need a rationale? lots yes, 1 no.

Lenkov asked for a volunteer. Silence. He said he'd ask again on Friday.

Anderson took exception to the new wording for the scope of friends in section 3.2p1. He thought it should say "local" scope instead of "function" scope. Shopiro said that the words in the Working Paper are those agreed to in Toronto. Pennello thought the Core subgroup should look at it again.

4 Scheduling subgroup sessions

Lenkov closed the committee of the whole.

5 Subgroup sessions

The committee recessed to subgroups at 14:55 on Monday.

6 Subgroup sessions

7 Subgroup progress reports

8 Subgroup sessions

9 Subgroup sessions

The committee reconvened at 14:00 on Wednesday.

Lenkov opened the committee of the whole.

10 General session

=== Extensions ===

Stroustrup presented Yaker's proposal to add *operator new[]* and *operator delete[]* to C++ (N0170 = 92-0093). These operators allows users to control free store allocation for arrays. Under this proposal, users can define:

```
void *operator new[](size_t)

void *X::operator new[](size_t)
```

as well as other signatures that support placement, and

```
void operator delete[](void *)

void X::operator delete[](void *)
- or -
void X::operator delete[](void *, size_t)
```

Schwarz asked if the proposal says anything about where the compiler finds the additional space to store the number of allocated array elements. Stroustrup said no, as long as array *new* and *delete* match, everything's OK. Yaker said the compiler is responsible for keeping track of such additional information. She said the Working Paper currently requires that a *new* expression pass the exact size of the allocated object to *operator new*. She suggested that this requirement (brought over from the ARM) may be a mistake. The Core subgroup may decide to change it. In any case, the requirement does not affect this proposal.

Roskind asked if *new[]* is a token. Yaker replied that whitespace can appear between *new* and [].

Plum thought the proposal leaned heavily toward preventing the implementation from storing bookkeeping information in the allocated storage. Yaker replied that the proposal lets implementations work either way.

Straw vote: Who approves of this proposal? lots yes, 3 no, 2 abstain.

Roskind objected to the proposal because allowing space between *new* and [] conflicts with existing practice. Gibbons explained that there are digraphs for [and] (<: and :> respectively) and the subgroup did not want keywords containing digraphs.

Stroustrup presented a revised proposal to add run-time type identification (RTTI) to C++ (N0198 = 92-0121). He said this proposal differs from the previous RTTI proposal (N0145 = 92-0068) in only two ways:

-- It provides greater respect for abstraction. You can no longer do a run-time checked cast to a private base class. (You cannot use a dynamic cast for "breaking and entering").
-- it replaces the checked cast notation *(?T*)(p)* with *dynamic_cast<T*>(p)*. This eliminates the potential mistake of writing *(T*)p* when you meant *(?T*)p*. The dynamic cast notation mirrors the template notation.

Stroustrup noted an addition to class *Type_info* that did not make latest version of the RTTI paper:

```
class Extended_type;
class Type_info {
    ...
public:
    int before (const Type_info &) const;
    const Extended_type* more_info() const;
    ...
};
```

He explained that *before* is for an ordering relationship, not a type relationship in a hierarchy.

Steinmuller noted that the spelling of *Type_info* conflicts with the proposal for naming conventions coming from the Libraries subgroup (namely, that all names should be lowercase without underscores). Stroustrup said he'd lobby against that convention. He said wanted the namespace concept so that these type names can be written as *Base* and *Extended*.

Becker asked if a friend function can use a dynamic cast to convert to a base. Stroustrup said yes, provided it can do so statically; that is, you can use a dynamic cast to cast to an accessible base class, but not an inaccessible base class. The rules for dynamic matching are taken from exception handling.

Stroustrup said this is a large proposal, and it will take time for Shopiro to integrate it into the Working Paper. Some of this belongs to the language proper, some belongs to the library, some belongs to runtime support.

Becker and Kendall expressed surprise that they were being asked to vote on the proposal. It did not appear to them to be ready. Stroustrup said that, except for the dynamic cast syntax, the proposal has not changed significantly in the last eight months.

Scian asked if Stroustrup had listed the proposed changes to the draft. Stroustrup said no, adding that if he wrote up all of the changes, they would be scattered in several places. He thought it should be done editorially. Scian said he'd like to see the changes beforehand.

Stroustrup made two points: (1) we don't like to push people into decisions, and (2) the recommendation from the Extensions subgroup was unanimous.

Plum expressed concern that templates and exception handling raised definitional issues that we are still grappling with. He asked how many new issues RTTI will introduce relative to the number created by templates and exception handling. Stroustrup replied that he didn't know of anyone who implemented either templates or exception handling in less than six months, yet he added RTTI to his research compiler in only two mornings. He implemented the run-time stuff with a page and a half of code; the change to the guts of the compiler was of the same magnitude.

Stroustrup later added that he can't promise that there won't be any problems. However, shying away from all change gives away some of the language's advantages. He said everyone is already "fudging" this feature. This is the best design we've been able to come up with. It is backed with implementation experience.

Kendall said this proposal has too much power. There are other features that are easier to implement and more powerful, but also too dangerous. Stroustrup said neither he nor anyone in the Extensions subgroup agrees with Kendall. He explained that most of the need for this type of RTTI comes when you don't have control over all of your source code, so you can't redesign your base classes. Many programs don't need this, and you can misuse it, but the subgroup has weighed this.

Steinmuller expressed mixed feelings about the proposal. Libraries need this feature, but it's not good to have a new language features every year. Gibbons said the issue is not whether we will have RTTI; every library already has it. The question is whether we want to have it standardized in every library or implemented differently everywhere.

Plum said that at some point, we have to simply say, "This is it. We're going to standardize this language." Maybe that point will only be an hour before we vote the draft out for public comment. He also said that hearing about Stroustrup's implementation experience is valuable, but noted that Stroustrup's compiler only had templates but not exceptions.

Stroustrup answered that templates is a purely compile- and link- time feature. RTTI is purely a run-time feature. Plum responded that we need to look ahead to the interaction between RTTI and exceptions. Stroustrup expressed confidence that exception handling needs a superset of this information. Stroustrup added that this proposed facility is safer and better specified than any of the library macro hacks that he has seen for RTTI, and he has seen about a dozen. If we don't have this one, we are going to have several dozen inferior hacked versions.

Schwarz expressed concern that different implementors will implement RTTI with different run-time support.

Saks did not want the Extensions subgroup to leave it to Shopiro to figure out how to translate the proposal into changes into the Working Paper. He thought Shopiro was not completely caught up, and would get even more work from other subgroups at this meeting. Stroustrup acknowledged that it is a hard job and offered to help. Shopiro offered to produce two versions of the draft for the next meeting -- one that has RTTI in it and one that does not.

Armstrong expressed concern that the cast syntax is much different from the C cast syntax. Bjarne said it's intentionally different because it has different meaning.

Kendall stated his technical objections:
-- a dynamic cast and an ordinary cast return different results when a definition of the derived class is not visible (only a forward declaration is visible).
-- it's dangerous for a dynamic cast to return null in case of failure.
-- it's inconsistent for pointer dynamic casts to return null while reference dynamic casts throw an exception.

Charney wanted to see more implementation experience before voting on the proposal. Schwarz said he's more likely to implement this after we vote on it, not before.

The committee discussed whether the changes should be written in precise form before taking a vote. Plum summarized the issue by noting that Stroustrup presented an overview of the RTTI feature. He asked if we should expect to approve the details, or should we expect to approve the concept and learn the details later.

Straw vote: Who favors accepting the substance of the RTTI proposal? lots yes, 2 no, 6 abstain.

Straw vote: Who wants a formal vote on RTTI on Friday? 17 yes, 15 no, 15 abstain.

=== Small Issues ===

Lajoie explained that the subgroup drew issues from two lists: one compiled by Gray, and the one by (former vice-chair) Miller. She said the subgroup dealt with six issues that were possibly substantive. They wanted to be sure the committee agreed with their decisions.

1. "forbid *operator op* as identifier names except as function names" (from Gray's list)

The subgroup realized that more general restrictions were needed. They proposed the following wording:

> A declarator-id shall be a simple identifier, except when it appears in:
>
> 1) the declaration of some special member functions (12.3, 12.4, 13.4).
> 2) the definition of a member function (9.3).
> 3) the definition of a static data member (8.4).
> 4) the declaration of a friend function that is a member of another class (11.4)

Plum explained that these restrictions ban using *operator+* as an identifier as in

```
int operator+ = 1;
```

Straw vote: Who favors this? lots yes, 0 no, 6 abstain.

2. "static data members and member functions aggregate initialization" (from Gray's list)

Lajoie said the subgroup proposed adding this rule to 8.4.1p2: "Initialization can only be performed on non-static data members."

Winder asked if this means you can aggregate-initialize a static data member. Pennello said the wording as presented wasn't quite right. You want to skip over the static data member, but this sentence doesn't say that. Pennello gave this example:

```
class C {
    int i;
    static int j;
    int k;
} c = {1, 2};
```

He wanted to be clear that *i* gets 1 and *k* gets 2, and *j* is skipped.

Lajoie wrote new words: "Aggregate initializers shall only be used to initialize the aggregate's non-static data members."

Straw vote: Who favors this clarification? lots yes, 0 no, 0 abstain.

Charney noted that the example clarifies the words immensely. McLay asked if this example be included in the draft. Clamage said it's up to Shopiro.

3. "1. Must virtual functions be defined?" (from Miller's list)

The subgroup's answer was "yes."

Currently the Working Paper (3.1) says "If a function is never called and its address is never taken, it need not be defined." The subgroup wanted to add "...unless it is a virtual member function that isn't pure virtual (10.2)"

Currently, the Working Paper (10.2) says: "A virtual function in a base class must be defined or be declared pure (10.3). A virtual function that has been defined in a base class need not be defined in a derived class." The subgroup wants to change these sentences to "A virtual function shall have a function definition or be declared pure (10.3). A virtual function that has been defined in a base class need not be redefined in a derived class."

Straw vote: Who favors this clarification? lots yes, 0 opposed.

[Secretary's note: Outside of committee, Lajoie said the subgroup will reconsider this proposal due to objections raised by Gibbons.]

4. "21. What is the interpretation of constructors and destructors for volatile objects? Must all constructors and destructors be compiled with volatile semantics?" (from Miller's list)

Lajoie said the subgroup's answer to second question was "no." The subgroup proposed to clarify the Working Paper by adding the following footnote to sentences 12.1p2.2 and 12.4p1.5: "Volatile semantics might or might not be used."

Lajoie gave this example:

```
class C {
    int i;
public:
    C();    // "C() volatile;" not allowed
};

volatile C j;

C::C()
{
    i = 3;
    i = 3;  // not optimized ?
}
```

The redundant assignment in *C::C()* may be optimized away.

Stroustrup explained that the semantics were designed for *const* and that *volatile* was grafted on.

Straw vote: Who favors this clarification? lots yes, 0 no, 6 abstain.

5. "Explicit cast to *void* not permitted." (from Gray's list)

Lajoie explained that section 3.6.1 of the Working Paper already says: "Any expression may be explicitly converted to type *void*." However, section 5.4 implicitly disallows an explicit cast to *void*. The subgroup proposal adding copying that sentence from 3.6.1 to 5.4.

Straw vote: Who favors this change? lots yes, 0 no.

Lajoie reported the subgroup considered another issue (from neither Gray's nor Miller's list). At the last meeting, the committee voted (as part of cleaning up the wording for friend declarations) to change a sentence in section 3.2 to "A name first declared by a friend declaration belongs to the global scope or a function scope." But, using "function scope" means the friend name would be in scope to the end of function, even if declared in an inner block. The subgroup proposed to change "function scope" to "local scope."

Straw vote: Who favors this change? lots yes, 0 no.

=== C Compatibility ===

Restating his proposal from page 7 of the previous meeting minutes (N0155 = 92-0078), Plum asked the committee to amend the Working Paper to use the following "spelled-out" names for the previously approved type categories: FT -> function type, COT -> completely-defined object type, IOT -> incompletely-defined object type.

Straw vote: Who favors this proposal? lots yes, 0 no, 8 abstain.

Plum also asked the committee to reaffirm its desire to add the lexical grammar from the C standard to the Working Paper.

Straw vote: Who favors this proposal? 30 yes, 0 no, 15 abstain.

Referring to pages 28 and 29 of the previous minutes, Plum asked the committee to approve adding both the definitions for C-struct and C-union and the accompanying layout-compatibility rules to 8.4.1p10 in the Working Paper.

Plum said he sent e-mail to all who stated objections to this proposal in Toronto, but no one replied. Schwarz said he objected to the term "C-struct". He also said he replied to Plum, but it apparently didn't get through.

Vilot was astonished that the notion of C-like structures, which was defined just to implement *offsetof*, would be extended to the entire language. He also said we need to see if linkage applies to data structures, an issue which seemed to have been dropped. He wanted to see this discussed in terms of the language, not just C compatibility.

Shopiro said this is highly relevant to structural compatibility. He thought it was a very complex issue that we don't have the answers yet. Plum replied that everything we're doing is a complex issue we don't have the answers to. He said the question is: are there low-level manipulations you can do in C that you can't do in C++?

Plum reminded the committee that, at the X3J16 organizational meeting in 1989, Stroustrup encouraged the committee to not leave room for a lower-level language. Plum thought failure to heed this advice would be a bad marketing decision. He said that if we want to be able to manipulate memory with C++ as we can with C, we must accept this proposal. He didn't want the desire to do more to keep us from adopting this clear proposal.

Koenig pointed out that the two C compilers on his workstation have mutually-incompatible layout schemes. Saks said Plum is only considering the behavior of C code when recompiled as C++. He thought this was well within the subgroup's charter.

Stroustrup agreed we should not leave room for a lower-level language like C. But we shouldn't let that lower-level language corrupt the type system of C++. He didn't think these two statements were contradictory, and he didn't think this proposal is the way to achieve these ends. Stroustrup said several people felt uncomfortable about the proposal, and maybe we need more time.

Schwarz said he'd vote against this proposal no matter when. Plum's item 6 says a cast from a *struct* to the type of its first member is well-defined. He thought this violated the type system beyond what's needed for low-level manipulation.

Bruck said he thought these rules do not give an real practical support for low-level programming.

Koenig explained that Ritchie intended to be able convert between pointers to two *structs* with a common initial sequence. He said that without this rule, you can't translate C++ inheritance into C. Schwarz said he'd show Koenig how.

Straw vote: Who favors this proposal? 16 yes, 8 no, 23 abstain.

Saks asked the abstainers what do they need to change their votes. Several members said they'd like to see a paper discussing the issues and different options.

Straw vote: Who would like to see a paper? 22 yes.

Stroustrup said the topic of the paper should be: What are the layout guarantees? He thought C++ could provide layout guarantees as strong as those in C. Bruns urged the people who opposed the proposal to get their reasons into the paper.

Plum summarized his understanding of the objections: There are certain objects in C++ that, once constructed, you must treat only as the type that it really is. Programs can't just treat these objects as arrays of bytes. Going further than that, one can imagine a tagged architecture environment where every object, no matter how created, would come into being as a particular type, and cannot "well-formedly" be looked at as anything other than its intrinsic type. The problem with rule 6 is that it punches a whole in this system, allowing an object to be both, say, a *struct* and an *int*.

Schwarz said Plum was close. He said he had no problem with words that require the first element and the struct be at the same location; he can do that without allowing casts. Schwarz did not object to the constraints, but to the methods used to take advantage of those constraints.

Action item: Plum will write a paper on layout-compatibility rules for C++.

Plum then proposed adding the "impressionistic" list of incompatibilities with C (N0174 = 92-0097) to the Working Paper as a non-normative appendix, as requested by SC22. Plum suggested that the list replace the existing section 19.2. The document is formatted in the style requested by SC22. Everything from 19.2 has been incorporated, except for things discussed in 19.2 that don't apply to a strictly-conforming C program.

Vilot questioned the format used for the entries in the list. He thought the last two items of each entry were too subjective. He thought each entry should explain how to change conforming C to C++. Other members also questioned the format. Plum acknowledged that the last two items of the format are subjective, but explained that SC22 dictated that format.

Stroustrup expected this list to change from meeting to meeting. He thought this should be a running document, not part of the draft. He said he was things documented as incompatibilities that really just need clarification.

McLay asked if we could we vote separately on this appendix when we submit the Working Paper as CD (committee draft). Plum said no, it must all be together when we submit the CD to ISO.

Straw vote: Who favors putting adding this to the draft now? lots yes, 1 no, 10 abstain.

Plum then introduced Nelson's paper on incompletely-defined object types (N0177 = 92-0100). Plum thought the proposed changes were editorial. Shopiro thought most were editorial.

Lenkov asked if anyone objected to treating this as editorial. No one objected.

Schwarz noted that the paper leaves questions unanswered, such as: Can I declare a function with an argument that is a reference to an incomplete type? Lenkov asked him to take this question offline.

Lenkov closed the committee of the whole.

The committee recessed at 17:45 on Wednesday and reconvened at 8:35 on Thursday.

11 General session

Lenkov opened the committee of the whole.

=== Libraries ===

Vilot introduced the Library subgroup report (N0159 = 92-0082), which summarizes the group's work up to and including the Toronto meeting. The group worked on language support (operators *new* and *delete*, etc.), strings, i/o, and containers. He said language support and i/o are ready for a formal vote of approval.

Vilot explained that the subgroup intended to define more than one string class providing at least two levels of support: a "low level" string class implementing arrays of characters, and a "high level" text class with national character set support (locales, *wchar_t*).

Vilot said the subgroup will propose a formal vote to accept N0161 = 92-0084 as the language support library specification. He said the subgroup made changes to the document during the week, but they felt that none of the changes were substantive.

Vilot said the subgroup will also propose a formal vote to accept N0179 = 92-0102 as the i/o library specification. Vilot related Schwarz's observation that the committee should have adopted AT&T's iostreams library specification as another base document. In effect, that's what the Library subgroup did. The subgroup sought to minimize the differences between the proposed iostreams and AT&T's version.

Vilot explained that, during the week, the subgroup decided that the iostreams proposal did not specify native language support (locales and shift states) properly, so they removed it from the proposal.

Vilot listed related issues under study by other subgroups whose resolution affects the Library group's work:

Category / Issue	Subgroup
General	
Namespace	Extensions
Language support	
RTTI/Extended RTTI classes	Extensions
operator new[], delete[]	Extensions
String	
Lifetime of temporaries	Core
Input/Output	
Order of initialization	Environment
Exception specifications	Core/Extensions
Containers	
Template friends (bits)	Core/Extensions
Lifetime of temporaries (dynarray)	Core

Yaker noticed small discrepancies between the language support proposal and the Working Paper. She asked Vilot if he considered them only editorial issues. Vilot did.

Straw vote: Who wants to add language support (N0161 = 92-0084) to the Working Paper? lots yes, 0 no, 1 abstain.

Stroustrup encouraged the Library subgroup to be aggressive in using language features to specify the library. If they find problems in language, they should toss them back to the Core or Extensions subgroups.

Lajoie wanted to see a statement of the relationship between iostreams and FILEs (from stdio). Vilot said version 3 of the iostreams proposal says the i/o on iostreams and FILEs can be interleaved character-by-character. Schwarz said the underlying file model for iostreams is the same as for FILEs. Whether *streampos* and *fpos* should be related is an idea that hadn't occurred to him, but should be considered.

Plum said he liked the comments in the iostreams proposal and would like them preserved. Vilot said they should go in the rationale. Plum wondered if we would have a rationale. Stroustrup suggested leaving the annotations in the Working Paper for now. Carter said he wanted to send a copy of the Working Paper to ITTF for preliminary review next month, but he can't send it with the annotations. Stroustrup suggested putting the annotations in, and then stripping them out of the copy sent to ITTF.

Straw vote: Who wants to add the iostreams proposal, revision 5 (N0179 = 92-0102) plus corrections, to the Working Paper? lots yes, 0 no, 0 abstain.

=== Environments ===

Stone presented the Environments subgroup's proposal to add translation limits to the Working Paper (N0178 = 92-0101). The proposal called for two sets of limits:

1. "combo" (combined) limits: a translator must accept a program that meets all of these limits
2. "solo" limits: a translator must accept a program that reaches each solo limit.

The proposal also introduced a definition for what it means for a program to conform to these limits.

Stone presented wording intended as the preface to the section on translation limit values:

> _._._ Environmental Limits
>
> _._._._ Translation Limits
>
> 1. The implementation shall be able to translate and execute at least one program that contains one instance of every "combo" limit.
>
> 2. The implementation shall be able to individually translate and execute all 44 programs, each of which contains one instance of the "solo" limits.

Clamage said he didn't understand this preface: Does the standard supply the 44 programs, or does each the vendor supply their own? Stone said it could be either way.

Stone explained that the "combo" limits, though numerically different, are conceptually the "rubber teeth" limits used by the C standard. The "solo" limits are a new idea.

Koenig noted that there's no way to verify that an implementation can handle all programs that stay within these limits. He was also concerned that vendors will treat limits as maxima, because:

1. they want to reject any program another compiler might reject, or
2. they might be lazy and used fixed allocations for compile-time resources.

So minima become maxima.

Stone explained that the subgroup is trying to determine what a strictly conforming program can and cannot do. Thus they introduced the idea of "twofold" conformance (see page 8 of N0178 = 91-0101).

Vilot objected that if we must write the 44 programs, that puts us in the certification business. He said he understood the need for environmental limits, like the maximum size of an object, or the number of external identifiers, but not for language limits, like the maximum number of friends, or the maximum number of base classes.

Stone replied that some of these categories are probably language issues, but they do address portability issues. He offered that if the numbers are too restrictive, we can change them.

Vilot restated his point: That environmental limits are turning into language limits. There may be an indirect relationship between what a programmer can say in a program and what the environment allows.

Plauger mostly agreed with Vilot. But he said there are some concrete things that limits do. People who write validation suites will make up numbers if we don't. Also, it's difficult to have an implementation that can uphold all of these limits by cheating.

Plum said that X3 re-emphasized last year to all technical committees that all conformance requirements must be specified in a standard. Someone not on the committee is supposed to be able to look at a program and tell whether it conforms, and look at translator and tell whether it conforms. X3 wants us to be very clear about conformance. Plum thought these limits were a step in the right direction, and wanted to approve them as is, with the intent to revisit conformance issues later. Plum added that Johnson is responsible for the conformance sections of draft, and he should be involved. Plum also argued that all these limits are environmental because there are two environments: the translation environment and the execution environment.

Shopiro did not want to include all 44 programs in the standard. He argued that you can't have conforming programs, just conforming processors. He said the notion that you can certify a program by a few tests is silly; a translator must be able to translate all correct programs.

Stroustrup said he has written programs that have broken these limits, and he suspects that many other people have. He suggested the solo limits should be unreasonably high. Koenig agreed, adding that high limits pressure vendors into flexible implementations. He suggested setting a limit anywhere a vendor might use a fixed array.

Carter said this proposal is very important because such a document will exist regardless of what this committee does. Governments around the world will create them. It would be better if they were all the same, and if they were set by this group. People should to talk to Stone to help him set the numbers.

Bruck disagreed with Plum about accepting the limits; he wanted to work out the meaning for conformance first. Saks agreed with Bruck, because it's hard to choose the numbers if you don't know what the numbers mean. Saks also suggested new wording for item #2 in the preamble:

2. For each of the "C++ solo" limits, the implementation shall be able to translate and execute at least one program that contains an instance of that limit.

Plum argued that limits have practical value. He suspected that many programs that exceed these limits might not run under limited environments, like MS-DOS. He said he didn't know if we would be allowed to produce a standard that didn't define conformance, or if the community would tolerate it. He suggested that if we say something about conformance, it should be close to what the C standard says.

Straw vote: Is the proposal ready for a vote? lots yes, 7 opposed, 14 abstain

Straw vote: Who understands what they are voting for? lots yes, 9 no.

Straw vote: Who wants to add the translation limits, along with the preamble, to the Working Paper? 32 yes, 10 no, 9 abstain.

Straw vote: Who wants to add the twofold conformance definition to the Working Paper? Several members expressed a lack of understanding about what they were voting on. 1 yes, lots no, 10 abstain.

=== Syntax ===

Roskind presented his proposal to eliminate context sensitive elements of the C++ syntax (N0173 = 92-0096).

Koenig liked the idea, and asked the subgroup to look for ways to collapse the set {class-name, enum-name, typedef-name} into type-name. He said this is important for the future, especially in describing templates. Turner agreed.

Stroustrup was concerned that we don't lose the context-sensitive information needed to parse *A*B*. Roskind said we don't.

Straw vote: Who favors this proposal? lots yes, 0 no, 2 abs.

Action item: Roskind will consider Koenig's suggestion (to collapse the set {class-name, enum-name, typedef-name} into type-name).

Krohn presented his proposal on qualified class names (N0173 = 92-0098). He gave this example from the paper:

```
struct A {
    struct B {
        struct C {
            int i;
        } C;
    } B;
    enum E {
        e = 1;
    } E;
};
```

The problem is that the member *C* in *A::B* hides the *struct C* in *A::B*. Thus *A::B::C* refers to the data member, not the nested type. Krohn proposed using the notation *struct A::B::C* to refer to the *struct C* in *A::B*. Koenig noted that we need *struct A::B::C* to disambiguate *C*, but not *B* because the :: following *B* means it must be a class-name. Krohn agreed.

Plum noted that this is not technically a C compatibility issue, as suggested by Krohn. Koenig said it's a C interoperability question. Stroustrup said it's a practical problem, but didn't know if it needed to be solved. People do write stuff like this in C++. Stroustrup said we can either say "tough luck", which was fine with him, or we can use struct as a disambiguator. It is a practical problem, not a legalistic or compatibility problem.

Roskind claimed this is not an extension; that there is existing practice. Stroustrup countered that if it's not in the manual [the ARM], it's an extension.

Koenig gave an example supporting the need for this notation in writing templates:

```
1:  template <class T> class X {
2:      T::t a;
3:      // ...
4:  };
```

He said compilers will complain that they can't parse line 2 because *T::t* is not a type. If we change line 2 to

```
2:  class T::t a;
```

then it's clear that *T::t* must be a type.

Plum wanted to know how many vendors already implement this syntax, and how many solve the problem differently. Turner said Liant implemented this feature as proposed. Krohn said Pennello said that Metaware, Borland and/or Microsoft, and Zortech have also implemented this. Schwarz said Lucid took a slightly different approach, but he had no problem with accepting this change. Stroustrup said he thought cfront also does this.

Someone suggested that this change would allow forward declaration of nested classes. Stroustrup said it explicitly does not.

Straw vote: Who favors Krohn's proposal? lots yes, 0 no, 4 abstain.

=== Core Language ===

Koenig summarized the Core subgroup's discussions during the week. He said the name lookup issue looks very simple on the surface, but it is very complex. He wanted to get the committee's impressions of the subgroup's alternative approaches.

Koenig said the BIG issue is: What does *b.A::z* mean? That is, how do you look up names involved in this expression? Koenig added that the meanings of *b.z* and *A::z* were not at issue. Neither was *b.A::B::C::D::Z* because once we understand what *A* means, we understand everything else.

Koenig said there were three proposals:
1. by Armstrong, to look up *b.A::z* in the scope of *B* (the class of *b*)
2. by Pennello, to pretend *A::z* appeared in a member function body of *B*
3. by Gibbons, to modify lookup rules so that base classes are found as members

Koenig summarized the work assignments:
Action items:
Lajoie will track open issues.
Plum will help Lajoie establish a database and tracking procedures.
Armstrong will write a proposal for looking up *A* in the scope of *B*.
Pennello will write a proposal for looking up *A::z* "as if" in member of *B*.
Gibbons will write a proposal for looking up base classes "as if" they were members.
Turner will write a summary of agreements on name lookup starting with the Lund meeting.
Scian will summarize some specific minor issues regarding *A::B*, *enum { x = sizeof(x) }*, etc.

Koenig reported that the subgroup made no progress on the lifetime of temporaries. He suggested that in the absence of a strong consensus for an alternative, the wording in the draft (letting implementors choose) may stand. Koenig said he didn't think most people favored the status quo. Stroustrup agreed that users don't want the choice left to each implementor.

Shopiro said that if we leave the draft as is, programmers must assume short lifetimes of temporaries. Koenig said no, it's worse than that; they must assume that the implementation will do what is the least favorable for them.

The discussion returned briefly to name lookup. Koenig said the are two reasons for spending so much time on name lookup:
1) We need to know how to handle template parameters, and
2) environments that store C++ programs in non-text form must be able to reconstruct programs for display purposes.

Vilot thought that the group was spending too much time perfecting something that need not be perfect. He said this is a classic situation where 5% of the cases take 95% of the time. Pennello replied that compromise rules take longer because you must list each special case you're not willing to solve. Pennello also thought the subgroup was on the verge of a simple rule that covered all cases well.

Koenig returned to lifetime of temporaries. He listed alternative policies for destroying temporaries: Unconditionally constructed temporaries are destroyed
1. at end of statement (EOS)
2. at end of block (EOB)
and temporaries created in conditional expressions are destroyed
a. at the end of the conditional branch (EOCB)
b. with other temporaries (which may requires setting runtime flags)

Vilot wondered if a straw vote might help the subgroup determine the committee's preferences.

Plum suggested another alternative: The lifetime of temporaries is implementation-defined, but temporaries hang around at least until end of statement.

Schwarz thought the lifetime of temporaries is the most important issue before the Core subgroup. He also thought that discussing the details of the issue in full committee would be unproductive. If we vote on anything, it should be on setting the Core subgroup's priorities.

Koenig thought a straw vote would provide useful guidance. He wanted to know how many people would accept the status quo.

Straw vote: Who is...

	willing to accept ?	cannot accept ?
1a (= EOS + EOCB)	38	0
2a (= EOB + EOCB)	25	6
1b (= EOS + EOS)	34	1
2b (= EOB + EOB)	28	8
3 (= any time)	6	32

Straw vote: Who will not be able to vote on this issue next time unless they see a paper? 21 yes.

=== Small Issues ===

Lajoie asked the committee's consent on the following issue regarding *operator void*. She gave this example:

```
class X {
public:
    operator void();
    X operator+(int);
} x;
```

She said it's not clear when *operator void* would be used. For example:

```
x;          // called implicitly here?
(void)x;    // called explicitly here
x+3;        // called implicitly here?
```

Lajoie noted that the current language definition implies:

```
class C {
public:
    operator C(); // illegal (12.3.2)
    operator C&() // illegal (12.3.2)
};
```

The subgroup recommended disallowing *operator void* as well, because
-- you can only guarantee it will be invoked by calling it explicitly
-- the Working Paper doesn't describe when void expressions exist, so we can't describe when *operator void* would be called implicitly.

Koenig suggested banning *operator const void*, *volatile void*, and *const volatile void* as well.

Straw vote: Who favors this change? lots yes, 0 no, 1 abstain.

Lajoie explained that the committee needs a database for tracking open issues. She presented a standard format for submitting issues to that database. Each submission should be of the form:
- -- Email number
- -- Title
- -- Section of the Working Paper
- -- Status (active, received, closed)
- -- Description
  - Problem Description
  - Resolution
- -- Requestor
- -- Seconder
- -- Technical Owner
- -- Related Emails
- -- Related Papers

No one objected.

Action item: Lajoie will post to the core reflector the format for submitting core issues.

Lajoie also said the subgroup will fill out forms like this for issues already on their issues lists.

=== Extensions ===

Stroustrup reported that the subgroup has been working to determine the legitimate range of values for an enumerated type. He presented their proposal. They were not ready for a formal vote because they didn't have the exact wording and they wanted to give the Core subgroup a chance to see the proposal.

The subgroup suggested:
** The values of an enumeration type are the values of the smallest bit-field that will hold all the enumerators.
   -- the values are signed only if an enumerator is negative
   -- casting a value in this range to the enumerated type is well-defined and value preserving
** *sizeof* an enumerated type is the sizeof the smallest integral type that will hold all the values.
   -- a program is ill-formed if no such integral type exists or is supported
** the constant-expression that assigns a value to an enumerator may be any integral type and is not restricted to int.

Straw vote: Who favors this proposal? lots yes, 0 no, 3 abstain.

Plauger suggested that we adopt a formal resolution on the "three-week rule" -- any papers not mailed at least three weeks before a meeting can be challenged by any member and not considered. Lenkov requested wording for this resolution.

Action item: Shopiro will incorporate all of the changes that were proposed by the Small Issues subgroup and approved by straw votes as editorial clarifications:

1. A declarator-id shall be a simple identifier, except when it appears in...
2. Aggregate initializers shall only be used to initialize the aggregate's non-static data members.
3. Add words to 3.1 and 10.2 clarifying when virtual functions must be defined.
4. add a footnote to sentences 12.1p2.2 and 12.4p1.5 clarifying volatile semantics for constructors and destructors: "Volatile semantics might or might not be used."
5. Copy "Any expression may be explicitly converted to type *void*." from section 3.6.1 to 5.4.
6. Disallow *operator void*.

=== Project Schedule ===

Lenkov saw four parts to the discussion of the project schedule:

1. What qualities must the working paper have before we can submit it as a CD (committee draft)?
2. When can we get the document to this state?
3. Do we need to change our policies and procedures, etc., to meet 1 and 2 above ?
4. Is there enough time for public review? Are the current rules sufficient?

Lenkov observed that opinions vary on how much polishing the document needs. Plum said there's no hard and fast rule for when the document will be ready to submit as a CD. X3J11's criteria was that they would send the document out for public review when they were ready to live with it as the standard. He added that we need to look at the backlog of issues before we can project a completion schedule.

Plauger said we want to submit a CD as soon as possible so the world knows what we're working on, but not so soon that the draft is embarrassing. When you go to DIS (draft international standard) ballot, the document better be something you can live with.

Bruck didn't think we could set down written criteria for when the document is good enough to ship. He suggested just picking a date and shipping then.

Vilot proposed the following criteria for submitting the CD:
-- the set of language features is complete and stable,
-- there are no pending extensions of any consequence,
-- there are no existing language features that are dramatically unclear,
-- there are no mutually contradictory statements in the Working Paper, and
-- there are no missing cases.

Saks said it's difficult to specify the criteria for deciding when you're happy with the document. He thought there was a wide disparity among the committee members about what still needs to be done. He suggested that we need to work toward a greater shared perception. Saks praised Vilot for presenting not only what his subgroup had to do but what they needed from the other subgroups. Saks suggested that the subgroup chairs share this kind of information with each other, or that the committee poll its members individually to gain a sense of what needs to be done.

Schwarz expressed concern about the brevity of each public review period. Plum said there's not much we can do about that. Plum also said he liked most of what Vilot said, but thought we should try to do as much wordsmithing as possible before we submit the CD.

Stroustrup reminded the committee of Parkinson's Law, that work expands to fill all available time. He expected that the Extensions subgroup would finish its work, including repairs on templates and exceptions, by the last meeting of 1993. He then expected to be ready for CD in the summer of 1994.

Plauger said we will have to pick a date based on incomplete information and live with it.

Spicer suggested that the subgroup chairs should produce a list of things they need to do and the time it will take. Then they should post them and solicit comment as the basis for planning the schedule.

Carter explained that he must report the project schedule at the SC22 meeting every September. Lenkov must report the schedule at the X3 meeting in March. This does not usurp this group, but the established process requires that we give a date. It is important for us to "buy into" these dates. He said it's difficult to set a date, but it's one of the few procedures that must be followed.

Carter said we could wait until the next meeting to decide on the schedule. Lenkov said he could also wait, although it would be less convenient for him.

Vilot reiterated his criteria for submitting the CD. He added that the criteria for submitting the DIS is that all the legalese and all the rationale comments must be in the right places.

Lenkov asked if we need any other milestones beside CD registration and DIS ballot. There was no response.

Straw vote: Who thinks Vilot's criteria are reasonable? lots yes, 1 no, a few abstain.

Lenkov asked if we need to change the procedures we use at meetings. Carter suggesting freeing up more time on Mondays.

Becker said we were being too democratic. He wanted the subgroup chairs and committee chair to meet and decide on work plans and schedules. He didn't want to waste any more time in whole committee.

Koenig wanted less structured work time. Plauger agreed, but said we shouldn't spend less time in full committee; that's how we reach consensus.

Schwarz asked to see a schedule at each meeting of what the subgroups will work on during the week.

Straw vote: Who wants the X3J16 chair and subgroup chairs to take more responsibility? lots yes, 0 no, 4 abstain.

Lenkov summarized the suggestions he'd heard: The "management committee":

1. should publish a schedule of what will be discussed during the week.
2. should decide which issues are ready for a formal vote.
3. may decide to reschedule general sessions as appropriate.

Straw Vote: Who favors this plan? lots yes, 0 no.

Ward asked if it was a problem that two of the subgroup chairs were not principal members. She got no response.

Some members suggested that the issues database should track all activities, not just the core issues. Lajoie offered to log other issues and try putting them in a separate database for automated retrieval.

Plum said the purpose of the database was to tracking all outstanding core issues. He didn't want to cloud the picture with other group's issues.

The committee briefly discussed the schedule itself.

Straw vote: Who wants to defer the discussion of the schedule to the March meeting? lots yes, 0 no.

Lenkov closed the committee of the whole.

The committee recessed at 17:10 Thursday and reconvened at 08:35 Friday.

12 General session

Lenkov opened the committee of the whole.

=== Extensions ===

Stroustrup presented the subgroup's deliberations.

The subgroup considered a proposal to add *^ as an exponentiation operator. The claims were:
-- it provides a better notation,
-- it's more efficient than function calls,
-- simplifies conversion (from Fortran),
-- it's "more standard"
But, the subgroup decided
-- using ** is possible although implementation tricks are needed
-- *pow(double, double)* and *pow(double, int)* could provide the needed efficiency
-- there was no evidence of great urgency
-- ^ is a national character

The subgroup decided the reject the proposal

Action item: Bruck will explain the rejection of ^* to the proposer.

Koenig asked if the Libraries subgroup considered allowing <math.h> functions to be intrinsics. Vilot said Plauger has suggested it, but nothing has been done.

Gibbons suggested that we need *pow(int, int)* as well as *pow(double, double)* and *pow(double, int)*.

Stroustrup reported that the subgroup also considered a proposal to add restricted pointers to C++. The proposal came from NCEG (X3J11.1, the Numerical C Extensions Group) through Holly. The claims was it allows better code generation (10 to 30 times for key fragments). But the subgroup judged that
-- it was unsafe
-- alternatives had not been explored
-- it was not as important for C++ as it was for C
-- it's architecture dependent (like near and far)

The subgroup decided to reject the proposal. Holly noted the decision was not unanimous, but he went along because he had no support.

Stroustrup mentioned that the subgroup has a proposal from Hansen to allow forward declaration of nested classes, as in:

```
class X {
    class Y;  // declare X::Y;
    Y *f(Y);
    Y *p;
    // ...
};
class X::Y { ... }   // define X::Y
X::Y X::f(Y a) { ... }
int X::V = 27;
```

However, nobody had a copy of the proposal, so they'll work on it next time.

Stroustrup then discussed the ~*const* proposal. The claims were that it leads to
-- cleaner const notion
-- better documentation
-- better optimization
But the subgroup judged that
-- the optimizations were not significant
-- we still need a way to cast away *const* (we can't eliminate such casts altogether)
-- the feature is too complex

Stroustrup explained the virtue of ~*const* using this example:

```
class X {
    int cache;
    // ...
    f() const;
};

const X c(args);

X::f() const
{
    ((x *)this)->cache++; // must use cast to strip away const
}
```

If we had ~*const*, the example would become

```
class X {
    ~const int cache;   // cache is never const
    // ...
    f() const;
};

const X c(args);

X::f() const
{
    this->cache++; // no need for cast to access ~const member
}
```

Thus ~*const* reduces the need for casts.

Stroustrup said the subgroup will consider strengthening the treatment of *const* beyond the original proposal. The new rule might be: Except for members declared ~*const*, an object declared *const* may not be modified (i.e. an attempt to modify it has undefined behavior). In any event, it needs more work.

Stroustrup described the latest namespace management proposals. He gave this example.

```
// my.h:
int f(int);
class string { ... };

// your.h:
double f(double);
class string { ... };
```

Composing programs with components from different sources leads to name clashes. He said it's a real problem. The proposal let's you write

```
namespace my {
    int f(int);              // "my" f
    class string { ... };    // "my string
};

namespace your {
    double f(double);        // "your" f
    class string { ... };    // "your" string
};

my::string s = "asdf";       // uses "my" string
...
your::f(2);                  // uses "your" f
```

If you don't want to write *my::* all the time, you can write

```
using my;
string s = "asdf";       // uses "my" string
f(2);                    // uses "my" f
your::f(2);              // uses "your" f
```

Stroustrup presented further details, which he will be explain in a numbered paper in the next mailing.

Stroustrup presented the proposal for a new cast notation replacing the old cast notation *(T)v*. He said the old syntax is "invisible", and it semantics are "slippery". Using old casts is error prone and causes maintenance problems.

Stroustrup proposed adding these new casts:

```
dynamic_cast<T>(v)
static_cast<T>(v)
reinterpret_cast<T>(v)
const_cast<T>(v)
```

and deprecating the old-style casts.

He said the new cast notation is:
-- very visible,
-- documents intent,
-- allows compile time error detection,
-- respects abstraction,
-- respects const

Scian said the list of casts is not complete; he thought *volatile_cast* was missing. Stroustrup replied that it's more complete than Scian thought.

Stroustrup summarized the open issues (in no priority order):
+ RTTI (cleanup)
+ namespaces
+ casts
templates
exceptions
+ ~const
extended character sets
+ forward declaration of nested classes
operator.()
member initializers within class
where + identifies most likely candidates for votes at the next meeting.

Lenkov closed the committee of the whole.

=== Formal Motions ===

Motion (to approve the current Working Paper) by Saks/Yaker:

Move we accept 92-0091 = N0168 as the current Working Paper, provided the editor accepts the following action items:

1. fix the grammar for initializers; see March, 1992 minutes (N0118 = 92-0041).
2. incorporate definitions for function type, completely-defined object type, and incompletely-defined object type; see July, 1992 minutes (N0155 = 92-0078)
3. add lexical grammar; see July, 1992 (N0155 = 92-0078)
4. incorporate changes described in N0177 = 92-0100 regarding incompletely-defined object type.

Motion passed X3J16: lots yes, 1 no.
Motion passed WG21: 5 yes, 0 no.

Motion (to extend C++ to allow operators *new[]* and *delete[]*) by Holly/Ward:

Move we amend the Working Paper as follows:

1. add changes specified in document N0170 = 92-0093, sections 3.4 and 3.5, and
2. specify that the names of the operators *new[]* and *delete[]* may have whitespace before the [].

Becker asked if this allows whitespace between the [] in *operator new[]* and *operator delete[]*. Yaker said the proposal allows whatever *operator[]* allows.

Motion passed X3J16: lots yes, 0 no.
Motion passed WG21: 5 yes, 0 no.

Motion (to incorporate translation limits) by Tooke/Plum:

> Move we create a new section on Translation Limits containing the following text:
>
> > The implementation shall be able to translate and execute at least one program that contains one instance of every "C++ combo" limit.
> >
> > For each of the "C++ solo" limits, the implementation shall be able to translate and execute at least one program that contains an instance of that limit.
>
> and including the table of translation limits from pages 4-7 of N0178 = 92-0101.

Clamage was surprised that this came up for a vote. Bruck noted that N0178 = 92-0101 says these limits are probably too low; he thought the vote was premature.

Motion passed X3J16: 25 yes, 19 no.
Motion rejected by WG21: 2 yes, 3 no.

Motion by Saks/Charney for X3J16 to reconsider the previous motion.

Schwarz said he understood at Lund that WG21 would vote only once to approve the draft. He didn't expect WG21 to vote on each technical issue. He asked the executive committee to straighten this out.

Motion to reconsider passed X3J16: lots yes, 5 no, 5 abstain.

Tooke/Plum withdrew the motion. Lenkov asked if any WG21 member objected. None did.

Motion (to add appendix on C compatibility) by Saks/Becker:

> Move that we add N0174 = 92-0097 to the Working Paper as a non-normative appendix to replace the current section 19.2.

Motion passed X3J16: lots yes, 0 no.
Motion passed WG21: 4 yes, 0 no, 0 abstain.

Motion (to add language support library) by Allison/Budge:

> Move we accept N0161 = 92-0084, with appropriate editorial changes, for inclusion into the Working Paper.

Sloane objected that the proposed standard *operator new* throws an *xalloc* exception on allocation failure than return null. He objected on three grounds:

1. He did not think the group had reached consensus. He cited his and O'Riordan's dissent at the Toronto meeting.
2. He thought it breaks too much existing code.
3. The "camel principle" - this is the only instance of exceptions in the language support library. He didn't want to see more and more exceptions introduced into library with this use of exceptions used as the justification.

Yaker agreed that consensus had not been reached.

Vilot explained that the proposal has been available since Nashua (March, 1991). He has documented objections in his reports. The subgroup considered these objections, and still reached consensus.

Plauger said consensus is not unanimity. Rather, it's trying to achieve unanimity. We must be sure that minority voices are heard.

Motion passed X3J16: lots yes, 5 no.
Motion passed WG21: 5 yes, 0 no.

Motion (to add the i/o library) by Vilot/Allison:

> Move we accept N0179 = 92-0102, less any mention of locales and shift-states and with appropriate editorial changes, for inclusion into the Working Paper.

Koch praised Schwarz for his work on the proposal. Becker noted that iostream throw exceptions in certain places.

Motion passed X3J16: lots yes, 0 no.
Motion passed WG21: 5 yes, 0 no.

Motion (to eliminate context-sensitivity from the grammar) by Saks/Krohn:

> Move we accept the changes to the Working Paper proposed in N0173 = 92-0096.

Motion passed X3J16: lots yes, 0 no.
Motion passed WG21: 5 yes, 0 no.

Motion (to allow qualified names in elaborated-type-specifiers) by Charney/Saks:

> Move we change the Working Paper as follows:
>
> -- change the grammar for elaborated-type-specifier to:

```
elaborated-type-specifier:
    class-key identifier
    class-key qualified-class-specifier :: identifier
    enum identifier
    enum qualified-class-specifier :: identifier
```

-- add text to sections 7.1 and 9.1 to disallow declarations of the form:

```
class-key qualified-class-specifier::identifier declarator-opt;

enum qualified-class-specifier::identifier declarator-opt;
```

where the identifier has not been previously declared. For example, disallow

```
struct A::B;
enum A::B e;
```

where *B* has not been declared.

Vilot asked if 'declared' should be 'defined.' Charney affirmed it should be 'declared'.

Stroustrup said the restrictions were there to insure that the proposal did not allow forward declaration of nested classes.

Motion passed X3J16: lots yes, 0 no.
Motion passed WG21: 5 yes, 0 no.

Motion by Carter/Dovich: The committee thanks Johnson and OSF for hosting the meeting.

Motion passed WG21+X3J16: lots yes.

13 Review of the meeting

13.1 Formal motions on technical issues considered during general sessions

See Appendix B.

13.2 Review of decisions made and documents approved by the committee

Action items:

1. Saks will post the Toronto meeting minutes to the editorial reflector, and post a note to the all reflector announcing the posting.

2. Plum will produce a list of erroneous C++ constructs requiring diagnostics.

3. Plum will write a paper on layout-compatibility rules for C++.

4. Roskind will consider Koenig's suggestion to collapse the set {class-name, enum-name, typedef-name} into type-name.

5. Lajoie will track open issues.

6. Plum will help Lajoie establish a database and tracking procedures.

7. Armstrong will write a proposal for looking up *A* in the scope of *B*.

8. Pennello will write a proposal for looking up *A::z* "as if" in member of *B*.

9. Gibbons will write a proposal for looking up base classes "as if" they were members.

10. Turner will write a summary of agreements on name lookup starting with the Lund meeting.

11. Scian will summarize some specific minor issues regarding *A::B*, *enum { x = sizeof(x) }*, etc.

12. Lajoie will post to the core reflector the format for submitting core issues.

13. Shopiro will incorporate all of the changes that were proposed by the Small Issues subgroup and approved by straw votes as editorial clarifications:
    1. A declarator-id shall be a simple identifier, except when it appears in...
    2. Aggregate initializers shall only be used to initialize the aggregate's non-static data members.
    3. Add words to 3.1 and 10.2 clarifying when virtual functions must be defined.
    4. add a footnote to sentences 12.1p2.2 and 12.4p1.5 clarifying volatile semantics for constructors and destructors: "Volatile semantics might or might not be used."
    5. Copy "Any expression may be explicitly converted to type *void*." from section 3.6.1 to 5.4.
    6. Disallow *operator void*.

14. Bruck will explain the rejection of ^* to the proposer.

Lenkov also noted the committee's decision to form an executive committee to handle procedures and schedules.

14 Plans for the future

Lenkov opened the committee of the whole.

Lenkov asked again for a volunteer to edit the rationale. Koenig said he thought that ISO didn't accept rationale documents. Plum suggested the rationale might be a non-normative appendix to the standard.

Carter spoke briefly about his responses to the questionnaire on cross-language standards. He offered a copy of any pending cross-language standard to anyone who wants. Send him email if you want one.

Carter also said he asked people to review his responses to the questionnaire. He had two or three documents left to review, and he will try to get them done by December 1. After the review, he'll post the responses to the intl reflector, and post a summary of each response on the all reflector.

14.1 Next meeting

Ward announced that the next meeting will be in Portland, OR, USA on March 7-12, 1993, hosted by Tektronix. She asked members to notify her if they needed need anything for the meeting. People asked her to provide more microphones (there were only three at this meeting), and to provide computers and printers (there were none at this meeting). Saks asked her to please arrange that the ballroom will always be available.

Lenkov said the Environments subgroup informational report on extensions for static initialization will be a technical session on Monday evening at the Portland meeting.

14.2 Mailings

The post-Boston meeting will be handled by OSF.

14.3 Following meetings

Hartinger invited WG21+X3J16 to Munich, Germany in July, 1993 at the Downtown Hilton. He said the hotel rooms will be approximately $130 to $140 (US), including taxes and breakfast.

Lenkov listed the meeting dates for the next three meetings:
-- March 7-12, 1993 in Portland, OR, hosted by Tektronix, Mentor Graphics, and Sequent
-- July 11-16, 1993 in Munich, Germany, hosted by Siemens Nixdorf
-- November 7-12, 1993 in Asilomar, CA or Palo Alto, CA hosted by Apple or Taligent

Clamage announced he will resign as vice-chair of X3J16. Lenkov said that Lajoie has offered to be vice-chair, and will act as vice-chair at the next meeting.

15 Adjournment

Lenkov closed the committee of the whole.

Motion by Plauger/Rabinov: "Move we adjourn."

Motion passed WG21+X3J16: lots yes, 0 no.

The committee adjourned at 12:00.

Appendix A - Attendance

Name	Affiliation	Stat	M	Tu	W	Th	F
Coleman, Kim	Apple	A	A	A	A	A	A
Rabinov, Arkady	Apple Computer	P	V	V	V	V	V
Koenig, Andrew	AT&T	A	A	A	A	A	A
Shopiro, Jonathan	AT&T	P	V	V	V	V	V
Stroustrup, Bjarne	AT&T Bell Labs	A		A	A	A	A
Carter, Steve	Bellcore	A	A	A	A	A	A
Krohn, Eric	Bellcore	P	V	V	V	V	V
Becker, Pete	Borland International	P	V	V	V	V	V
Rumsby, Steve	British Standards Institute	P	V	V	V	V	V
Swan, Randall	C-Team	P	V	V	V	V	V
Dovich, Steven J.	Cadence Design Systems	A	V	V	V	V	V
Kendall, Sam	CenterLine Software	P	V	V	V		
Bruns, John	Chicago Research & Trading	P	V	V	V	V	V
Arbogast, Carroll	Cognex	A	V	V	V	V	V
Comeau, Greg	Comeau Computing	P			V	V	V
Dewhurst, Steve	Computer Associates	P	A			A	
Price, Philip	Computer Sciences	P	V	V		V	V
Hashemi, Azar	Control Data	A	V	V	V	V	V
Holly, Mike	Cray Research	P	V	V	V	V	V
Raeburn, Ken	Cygnus Support	A	V	V	V	V	V
Allison, Chuck	DECUS	P	V	V	V	V	V
Winder, Wayne	Digital Equipment	P	V	V	V	V	V
Adamczyk, Steve	Edison Design Group	P	V	V	V	V	V
Anderson, Mike	Edison Design Group	A	A	A	A	A	A
Spicer, John	Edison Design Group	S	A	A	A	A	A
Gidman, John	FMR	P	A				
Lenkov, Dmitry	Hewlett-Packard	P	V	V	V	V	V
Banahan, Mike	Hoskyns	O	A	A	A	A	A
Greg Colvin	I. H. S.	P	V	V	V	V	V
Knuttila, Kim	IBM	P	V	V	V		
Lajoie, Josee	IBM	A	A	A	A	V	V
Nelson, Clark	Intel	P	V	V	V	V	V
McKenna, Christine	Intergraph	O	A	A	A	A	A
Roskind, Jim	James Roskind Computing	P	V	V	V	V	V
Munch, Max	Lex Hack & Associates	O	A			A	
Turner, Prescott	Liant Software	A	V	V	V	V	V
Schwarz, Jerry	Lucid	A	V	V	V	V	V
Bruck, Dag	Lund Institute of Technology	P	V	V	V	V	V
Yaker, Laura	Mentor Graphics	P	V	V	V	V	V
Pennello, Tom	MetaWare	P	V	V	V	V	V
McLay, Michael	NIST	P	V	V	V	V	
Vilot, Mike	ObjectCraft	P	V	V	V	V	V
Johnson, Andy	OSF	P	V	V	V	V	V
Stone, Paul	Perennial	P	V	V	V	V	V
Plum, Thomas	Plum Hall	P	V	V	V	V	V
Charney, Reg	Program Conversions	P	V	V	V	V	V
Wilcox, Tom	Rational	P	A	A	A	A	A
Colligan, Terry	Rational Systems	P	A	A	A	A	A
Eckel, Bruce	Revolution2	P	V	V	V	V	V
Saks, Dan	Saks & Associates	P	V	V	V	V	V

Budge, Kent G.	Sandia National Laboratory	P	V	V	V	V	V
Koch, Gavin	SAS Institute	P	V	V	V	V	V
Tooke, Simon	SCO Canada	A	V	V	V	V	V
Kiefer, Konrad	Siemens AG	P	V	V	V	V	V
Hartinger, Roland	Siemens Nixdorf	P	V	V	V	V	V
Steinmueller, Uwe	Siemens Nixdorf	A	A	A	A	A	A
Sloane, Alan	Sun Microsystems	A		V	V	V	V
Gibbons, Bill	Taligent	P	V	V	V	V	V
Clamage, Steve	TauMetric	A	V	V	V	V	V
Ward, Cynthia	Tektronix	P	V	V	V	V	V
Dodgson, David	Unisys	P	V	V	V	V	V
Houck, Christopher	Unisys	A	A	A	A	A	
Waggoner, Susan	US West	P	V	V	V	V	V
Strickland, Henry	Versant	P	A	A	A	A	A
Scian, Anthony	Watcom	P	V	V	V	V	V
Welch, James	Watcom	A	A	A	A	A	A
Armstrong, John		O	A	A	A		
Dawes, Beman		P	V	V	V	V	V
Plauger, P. J.		P	V	V	V	V	V
Teale, Steve		O		A	A		
Young, Michael		O	A		A	A	A
Zanjani, Ramin		P	A	A	A	A	
Total Voting Members			48	49	49	49	48
Total Attendance			67	67	68	66	62

Status: P = Principal, A = Alternate, S = Second Alternate, O = Observer
Mark: V = voting, A = attending (not voting)

Appendix B - Motions Passed

1. Motion by Saks/Schwarz:

   Move we approve 92-0078 = N0155 as the minutes of the previous meeting.

   Motion passed X3J16: lots yes, 0 no.
   Motion passed WG21: 5 yes, 0 no.

2. Motion (to approve the current Working Paper) by Saks/Yaker:

   Move we accept 92-0091 = N0168 as the current Working Paper, provided the editor accepts the following action items:

   1. fix the grammar for initializers; see March, 1992 minutes (N0118 = 92-0041).
   2. incorporate definitions for function type, completely-defined object type, and incompletely-defined object type; see July, 1992 minutes (N0155 = 92-0078)
   3. add lexical grammar; see July, 1992 (N0155 = 92-0078)
   4. incorporate changes described in N0177 = 92-0100 regarding incompletely-defined object type.

   Motion passed X3J16: lots yes, 1 no.
   Motion passed WG21: 5 yes, 0 no.

3. Motion (to extend C++ to allow operators *new[]* and *delete[]*) by Holly/Ward:

   Move we amend the Working Paper as follows:

   1. add changes specified in document N0170 = 92-0093, sections 3.4 and 3.5, and
   2. specify that the names of the operators *new[]* and *delete[]* may have whitespace before the [].

   Motion passed X3J16: lots yes, 0 no.
   Motion passed WG21: 5 yes, 0 no.

4. Motion (to add appendix on C compatibility) by Saks/Becker:

   Move that we add N0174 = 92-0097 to the Working Paper as a non-normative appendix to replace the current section 19.2.

   Motion passed X3J16: lots yes, 0 no.
   Motion passed WG21: 4 yes, 0 no, 0 abstain.

5. Motion (to add language support library) by Allison/Budge:

   Move we accept N0161 = 92-0084, with appropriate editorial changes, for inclusion into the Working Paper.

Motion passed X3J16: lots yes, 5 no.
Motion passed WG21: 5 yes, 0 no.

6. Motion (to add the i/o library) by Vilot/Allison:

> Move we accept N0179 = 92-0102, less any mention of locales and shift-states and with appropriate editorial changes, for inclusion into the Working Paper.

Motion passed X3J16: lots yes, 0 no.
Motion passed WG21: 5 yes, 0 no.

7. Motion (to eliminate context-sensitivity from the grammar) by Saks/Krohn:

> Move we accept the changes to the Working Paper proposed in N0173 = 92-0096.

Motion passed X3J16: lots yes, 0 no.
Motion passed WG21: 5 yes, 0 no.

8. Motion (to allow qualified names in elaborated-type-specifiers) by Charney/Saks:

> Move we change the Working Paper as follows:
>
> -- change the grammar for elaborated-type-specifier to:
>
> ```
> elaborated-type-specifier:
>     class-key identifier
>     class-key qualified-class-specifier :: identifier
>     enum identifier
>     enum qualified-class-specifier :: identifier
> ```
>
> -- add text to sections 7.1 and 9.1 to disallow declarations of the form:
>
> ```
> class-key qualified-class-specifier::identifier declarator-opt;
>
> enum qualified-class-specifier::identifier declarator-opt;
> ```
>
> where the identifier has not been previously declared. For example, disallow
>
> ```
> struct A::B;
> enum A::B e;
> ```
>
> where *B* has not been declared.

Motion passed X3J16: lots yes, 0 no.
Motion passed WG21: 5 yes, 0 no.