

HEWLETT PACKARD * CALIFORNIA LANGUAGE LAB

19447 Pruneridge Avenue, Cupertino, CA 95014 * Telephone: (408) 725-8111

July, 1989

*Dmitry Lenkov
dmitry%hpda@hplabs.hp.com
MS: 47LE
(408) 447-5279*

**TO: X3 Secretariat, CBEMA
311 First Street, N.W., Suite 500
Washington DC 20001**

Programming Language C++

Proposal

1. Identification of Proposed Project

1.1. Title

Programming Language C++.

1.2. Proposer

Dmitry Lenkov, California Language Lab, Hewlett-Packard.

1.3. Date Submitted

April 27, 1989.

2. Justification of Proposed Standard

2.1. Needs

C++ is a C-based object-oriented language invented by Bjarne Stroustrup at AT&T Bell Laboratories, and which is widely perceived now as one of the languages best suited for software development. The attendance at the USENIX C++ conference grew from about 200 in 1987 to about 500 in 1988. The number of people interested in attending the next USENIX C++ conference is expected to grow even higher. At the same time, C++ related presentations and tutorials take an increasingly larger place at the other technical conferences and professional workshops. While the growth of C++ popularity is enthusiastically supported by the C and UNIX communities, the main reason for such growth is that many companies are finding C++ of strategic benefit for their product development.

C++ is going through a much faster public acceptance path than most other languages. C++ is already available on almost all machines where C is used. Projects choosing C++ for development vary from small to very large and ambitious ones. Thus the requirements for the portability of C++ programs, for a full specification of C++ as a language, and for a specification of C++ libraries and supporting environment have reached a level of importance similar to that of other widely used and mature languages such as C and Pascal. The delay of a response to these requirements will have the following negative impacts:

a. Multiple language dialects

The number of different C++ implementations is rapidly growing. In the absence of a full language definition, many implementations introduce their own interpretation of some C++ features. This leads to various language dialects. Multiple dialects of a language always bring serious problems to both suppliers of language products and their users, as well as future standards committees. Experience with such languages as Fortran and Pascal clearly demonstrates that.

b. Missing language features

The current implementation of C++ by AT&T and the available definition of C++ do not provide certain important features that are needed for large and medium size development projects. Among them are exception handling, certain aspects of multiple inheritance, support of generic functions and types, and garbage collection. Because of this there are already implementations that support these features, or some of them, in various ways. Such implementations are now available within certain organizations internally. However, implementations that are commercially available can be expected in the near future, as well as a significant increase of their number. This expands the danger of multiple dialects.

c. Libraries

For many languages, a standard set of library functions has become a part of the language definition. This is obviously true for C. Such a set of library functions provides an extension of language functionality without overloading the language with features. At the same time it satisfies the needs of software development in some particular areas not covered by the language features.

The functionality of C++ libraries, as well as a standard set of header files, is not completely defined. This causes almost every development project using C++ to develop their own additional libraries.

d. Supporting environment

The role of a language supporting environment is much greater for object-oriented languages than for languages like C and Pascal. Smalltalk and Ada environments clearly demonstrate this. A C++ supporting environment may be the only reasonable way to allow more independence of class interfaces from class implementations, to facilitate header file management, and solve some other language related problems. Several existing C++ supporting environments are totally different. Further development in this direction will create semantic differences in the interpretation of some language features.

e. Fear of a language with no standard definition

Many development projects find C++ to be the most appropriate language for their needs. However a large number of them do not choose C++ because of the instability of the language and absence of any effort toward its standardization. Thus they choose C, Pascal, or some other language of this level and implement their own extensions which allow the support of object-oriented programming with the chosen language. This definitely increases portability and code exchange problems.

2.2. Recommended scope of Standard

The recommended scope of the proposed standard includes three layers of the language and associated libraries and tools.

C++ has been created by Bjarne Stroustrup at AT&T. Most of the C++ implementations are based on the AT&T translator implementation of C++ and Stroustrup's book and papers as well as other documents by him. AT&T C++ Language System - Release 2.0 contains the C++ reference manual by B. Stroustrup with the latest and most complete definition of the language, the translator version implementing C++ according to this definition, and three libraries. The first layer of the standard shall encompass the features and libraries available in this version of the AT&T C++ Language System. These features and libraries have been heavily used by various development projects.

The second layer includes primarily libraries, support environment features, and certain run-time mechanisms that have been implemented and sometimes used extensively, but still involve experimentation and have problems, or are being implemented and need more experience. By the time the C++ committee prepares a draft of the C++ standard, this layer of the language will be significantly more stable.

As mentioned in the previous section, there exist C++ implementations which have added certain features to the language, provided additional libraries, and introduced environment support for the C++ language. Many of these additions are listed as wanted by a large group of C++ users. These additions constitute the third layer. The C++ committee shall decide which of them will be included into the standard.

2.3. Existing Practice in Area of Proposed Standard

C++ implementations are available on almost all machines where C is used. There already exists a significant amount of software written in C++. The number of projects choosing C++ for development is growing faster than for other languages.

2.4. Expected Stability of Proposed Standard with Respect to Current and Potential Technological Advances

Three layers of the language described above are expected to have different levels of stability. The standard for the first layer should be exceptionally stable. The standard for the second layer of the language is expected to be very stable. The third layer of the language includes features wanted by the C++ community that are currently under development or being implemented. The C++ committee shall decide which of those features will be stable enough to be included into the standard by the time when the first draft of the standard will be completed.

3. Description of Proposed Project

3.1. Definitions for Concepts and Special Terms

None needed.

3.2. Expected Relationship with Approved X3 Reference Models

There seem to be no areas of conflict.

3.3. Recommended Program of Work

It is recommended that the work be divided into the following phases:

A. Organizational phase

During this phase several tasks shall be accomplished.

- A.1. Establish the charter of the C++ committee and identify deliverables of the project (documents, recommendations, etc.).
- A.2. Identify other domestic and international organizations, groups, and committees that administer any activities in regard to C++ standardization or may become interested or involved in such activities in the future; establish working relationships with these organizations, groups, and committees and unite the efforts whenever it is possible with the purpose of eliminating potentially different C++ standards. At least two actions by the committee are deemed to be necessary:
 1. recommend to these organizations that they advise their members involved with C++ to join the C++ committee;
 2. recommend to those among them that are either standard organizations or standard endorsers to initiate corresponding C++ committees, groups, or work items; when appropriate and practical become a proposer and convener of such C++ committees and groups.
- A.3. Establish working relationship with the C committee (X3J11).
- A.4. Identify subgroups of the C++ committee that would carry out major subtasks of the project (language definition, library specification, features of C++ supporting environment, etc.)
- A.5. Determine if any validation procedure for a compliance with the standard shall be considered by the C++ committee.
- A.6. Develop procedural guidelines for the project.

- A.7. Identify individuals from the industry and academic world who may provide significant input and help to the C++ committee or any of its subgroups; invite these individuals as members or advisers.
- A.8. Identify C++ committee's members responsible for various activities and subgroups of the committee during different phases of the project.

B. Drafting phase

The first draft of the documents planned by the C++ committee shall be delivered during this phase. It is practical during this phase to have separate subgroups to draft different documents and, in some cases, different parts of one document. Also editors of documents should be identified during this phase.

C. First editing phase

During this phase all members and advisers of the C++ committee and, possibly, some other selected individuals and organizations identified by the committee will provide their input in regard to the first draft. Necessary corrections shall be selected and incorporated into the draft.

D. Public review phase

This phase will include a significant amount of work to make the drafts of C++ standard documents available to the C++ community for evaluation and comment. It is practical to have separate subgroups to make a preliminary evaluation of different groups of comments.

E. Second editing phase

The evaluation of comments submitted to the C++ committee from the C++ community, selection of necessary changes to the draft documents, and incorporation of them into the documents will primarily constitute this phase.

3.4. Resources - Individuals and Organizations Competent in Subject Matter

A. Organizations represented in X3J11.

The discussion in regard to a C++ standardization effort at the X3J11 meeting in Seattle on April 10, 1989, indicates that many organizations represented there have significant interest and competence in C++. These organizations can be contacted through their representatives in the X3J11 committee.

B. Bjarne Stroustrup

It will be very important to invite Bjarne Stroustrup, the creator and first implementor of C++, to be a member or advisor to the C++ committee.

C. C++ interest groups

These groups include people from different organizations where C++ is used as a development language and/or where C++ implementations are developed. Such organizations can be contacted through people participating in these groups. The C++ subgroup of /usr/group is one of them.

D. 1988 USENIX C++ conference

This conference brought together about 500 representatives of the C++ community. The list of participants and their organizations provides the information necessary for making these contacts.

3.5. Recommended X3 Development Technical Committee

The proposer strongly believes that one of the important factors of a wide acceptance of a C++ standard, and conformance by a large number of C++ implementations to this standard, is this standard's compatibility with ANSI C. The community of C users and C product suppliers has spent a large amount of effort to arrive at the ANSI C definition as well as the ANSI definition of C libraries. It will take a continuing effort and investment from C product suppliers to conform to ANSI C and from the developers to port existing C code to ANSI C. To minimize future investments in similar efforts, it is very important to provide compatibility of the future standard C++ with ANSI C.

C++ is not strictly compatible with either C, as defined by Kernighan and Ritchie, or ANSI C. The differences between ANSI C and C++, as defined by the AT&T 2.0 reference manual, are minimal and do not present a practical barrier for converting ANSI C programs into C++ programs. However it is important to

1. develop a standard conversion procedure that may include conversion routines and a set of documented steps to be taken;

2. establish a coordinating procedure between X3J11 and the C++ committee that will assure the same level of compatibility between ANSI C and C++ when any new extensions are considered for them.

With this in mind, it is recommended for the development of the proposed project to form:

1. New X3 committee to standardize C++;
2. Close liaison with the C++ committee within X3J11 to support the level of compatibility between C and C++ mentioned above.

3.6. Anticipated Frequency and Duration of Meetings

The full C++ committee shall meet quarterly except possibly for the drafting phase, when semiannual meetings might be sufficient.

Each subgroup of the C++ committee shall have as many meetings as practical.

The duration of meetings will be dependent on the amount of work that the committee or a subgroup has to complete.

3.7. Target Date for dpANS to X3 (Milestone 10)

The target date is 3 years from the first organizational meeting.

3.8. Estimated Useful Life of Standard

The useful life of this standard will depend on the number of C++ features, currently in use or under development, included in the standard and on the compatibility with ANSI C. If the chosen features satisfy the requirements of modern software development, and if a sufficient level of compatibility with ANSI C is provided by the standard, its useful life is expected to be 7 to 10 years.

4. Implementation Impacts

4.1. Impact on Existing User Practice and Investments

This project will stimulate a greater number of language product suppliers to produce C++ products and to support them with tools and environments of better quality and richer functionality. This will improve existing user practice and will lower user cost of software development.

4.2. Impact on Supplier Products and Support

This project will provide the base for more C++ products and better product quality. The support of these products will improve because the standard will allow suppliers to understand and better identify user problems and solutions.

4.3. Techniques and Costs for Compliance Verifications

Techniques and costs for compliance verifications are expected to be comparable to ones for existing language standards such as ANSI C and Pascal. The C++ committee shall decide if the development of any recommendations in this regard will be a part of the project.

4.4. Legal Considerations

No legal problems are anticipated.

5. Closely Related Standard Activities

5.1. Existing Standards

There are no existing standards for the C++ language.

5.2. X3 Standards Development Projects

As discussed in section 3.5, the C development project (X3J11) is closely related to the proposed project.

5.3. X3/SPARC Study Groups

None.

5.4. Other Related Domestic Standard Efforts

None known to the proposer.

5.5. ISO Standards Development Projects

The ISO C standard working group (ISO/SC22/WG14) is closely related to the proposed project for the same reasons as X3J11. At the meeting in Seattle on April 10, 1989, the ISO/SC22/WG14 group expressed a strong interest in a C++ standardization effort.

5.6. Other Related International Standards Development Projects

None known to the proposer.

5.7. Recommendations for Coordinating Liaison

1. Close liaison between the C++ committee and X3J11, WG14, and other groups involved in the definition of C and C extensions is recommended in order to provide maximum level of compatibility between C and C++ languages.
2. Close liaison between the C++ committee and all C++ committees and groups within domestic and international standard and standard endorsing organizations is recommended in order to avoid multiple definitions of the C++ language.