



# Objective Viewpoint

## A Cornucopia of C++ Resources

By [G. Bowden Wise](#)

As a "general purpose" programming language, C++ is full of features and constructs. It is impossible for C++ programmers to remember every detail of this immense language. When we begin to scratch our head, and ask "How do I do this?" or "Can this be done?" we must consult our library of C++ reference material.

There is a wealth of information available for your own C++ library. There are

- [books](#),
- [published standards](#),
- [journals and magazines](#),
- [newsletters](#),
- [Usenet newsgroups](#),
- [FAQs](#), and
- [on-line Web resources](#).

In this issue of OBJECTIVE VIEW POINT, we take a look at my C++ bookshelf.

### [Books](#)

There are a **lot** of books on C++. You probably have more than two. There are

basically three different types of C++ books: *introductory or tutorial* books which teach newcomers how to start programming in C++; *general reference* books which cover most of the language in pretty good detail; and *special-purpose or advanced* books which cover specific C++ constructs in more detail or offer advice on how to avoid traps and pitfalls.

### Introductory and Tutorial Books

When learning a new language, you need a good introductory book that gives you a firm foundation of the language and presents the concepts in a straightforward manner without bogging you down with too many details. A good book will have you writing your first program in the first chapter. One of the first books I came across was Graham's book:

- GRAHAM, N. [\*\*Learning C++\*\*](#). McGraw Hill, Inc., 1991.

Those with a C background will like Pohl's book:

- [\*\*POHL, I. C++ for C Programmers\*\*](#), 2nd ed. Benjamin/Cummings Pu. Co., 1994.

### General Reference Books

Every C++ programmer needs a couple of general reference books. Books which you can turn to when you forget syntactic details like how to initialize a static data member of a class template or which type conversions are performed implicitly by the compiler. Here are two good reference books:

- [\*\*LIPPMAN, S. B. C++ Primer\*\*](#), 2nd ed. Addison-Wesley, New York, 1991.
- [\*\*STROUSTRUP, B. The C++ Programming Language\*\*](#), 2nd ed. Addison-Wesley, New York, 1991.

My favorite is Lippman's book. However, I also frequently consult Stroustrup's for a second opinion. Having two good reference books gives you two unique explanations of the same concept which usually helps you come to understand the concept better.

### Special Purpose and Advanced Books

As you write C++ programs, you become aware of the traps and pitfalls of the

language. Things like when you should and should not use references. Or, when must you return an object and not a reference. [Scott Meyers](#) has written two excellent books full of pitfalls to avoid and other good advice. His clear explanations help you write better C++ programs that do what you intended and do it efficiently.

- MEYERS, S. [\*\*Effective C++: 50 Specific Ways to Improve Your Programs and Designs\*\*](#). Addison-Wesley, Reading, Mass., 1992.
- MEYERS, S. [\*\*More Effective C++: 35 New Ways to Improve Your Programs and Designs\*\*](#). Addison-Wesley, Reading, Mass., 1996.

Another good book for intermediate C++ programmers is Murray's book. This book includes chapters on inheritance, templates, re-usability, and exceptions.

- MURRAY, R. B. [\*\*C++ Strategies and Tactics\*\*](#). Addison-Wesley, Reading, Mass., 1993.

C++ compilers provide elaborate mechanisms to support such object-oriented constructs as inheritance, polymorphism, and run-time type identification. Lippman provides an excellent "look under the hood" and discusses an implementation of the C++ object model in detail. Understanding details such as constructor semantics, encapsulation, member functions, virtual functions, and object lifetimes will help you write better C++ programs.

- LIPPMAN, S. B. [\*\*Inside the C++ Object Model\*\*](#), Addison-Wesley, New York, 1996.

There are many different styles, or idioms, for writing programs. As a general purpose language, you can use C++ for just about any problem. However, sometimes other languages might provide a better fit for a problem.

Pure object-oriented languages, such as Smalltalk, allow the type of an object to change at run-time. Because C++ has a more static type system, the type of all objects must be known at compile time. Virtual functions do allow programs to delay some details of their type until run-time. Coplien shows how to empower C++ classes with run-time type changing more akin to Smalltalk through the use of "exemplars." This book contains discussions of other idioms with C++ examples.

- COPLIEN, J. O. [\*\*Advanced C++ Programming Styles and Idioms\*\*](#). Addison-

Wesley, Reading, Mass., 1992.

## The C++ Standard

The C++ standards effort in the US is officially carried out by a committee of the **American National Standards Institute (ANSI)**. The ANSI committee serves as a representative to the international standards committee of C++ orchestrated by the **International Standards Organization (ISO)**.

The committee is always at work on one of the ``draft working papers'' (or DWPs), and holds 3-4 meetings per year. Every once in a while, they release a DWP to the public; these drafts are called ``Committee Drafts'' or CDs.

For the nitty gritty details, take a look at the current public Committee Draft:

- KOENIG, A. E. Working paper for Draft Proposed International Standard for Information Systems - Programming Language C++. Document Number: X3J16/95-0087, WG31/N0785, April 28 1995.

Also on the Web:

**<http://www.maths.warwick.ac.uk/c++/pub/wp/>**

The *Annotated C++ Reference Manual*, better known as just ``the ARM'' documents the state of the C++ language as of December 1990 when it was first published. The ARM looks much like the C++ DWP, but with associated annotations. The ARM gives good explanations of the why and how C++ has been designed.

- ELLIS, M. A., AND STROUSTRUP, B. ***The Annotated C++ Reference Manual***. Addison-Wesley, Reading, Mass., 1990. Reprinted with corrections, 1992.

There have been several reprints of the book since it first was published in 1990. Glancing at the copyright page in my books show that mine was reprinted in 1992; a more recent printing was in 1994. The views found in earlier versions of the ARM often still hold. The original ARM contained one of the first proposals of templates.

If you are a C++ history buff, or want to read what one of C++'s renowned founders has to say about the language, then this book is for you:

- STROUSTRUP, B. [The Design and Evolution of C++](#). Addison-Wesley, New York, 1994.

## Journals and Magazines

By far, the most impressive C++ journal is the *C++ Report* published by SIGS Publications, New York.

- *The C++ Report*. SIGS Publications.  
Web: <http://www.sigs.com/publications/cppr/>

Edited by [Doug Schmidt](#), a professor at Washington University, St. Louis, the magazine is full of useful advice and information on C++. You will find regular columns by C++ experts: Jim Coplien (on C++ programming and design), [Andrew Koenig](#) (on C++ traps and pitfalls), [Scott Meyers](#) (on using C++ effectively), and John Vlissides (on design patterns). Occasionally, there is a column by Josee Lajoie on the C++ standards efforts.

The other good magazine devoted to C++ is the *C/C++ Users Journal*. Columns by P. J. Plauger, Dan Saks, and others are featured. The September 1996 issue includes several articles on STL.

- *C/C++ Users Journal*. Miller Freeman Inc.  
Web: <http://www.cuj.com/>

Their Web site contains an impressive collection of links to C++ sources.

## Newsletters

The journals above are excellent, but they cost money. Next we look at two newsletters which are available for free.

### Glenn McClusky & Associates C++ Newsletter

Recently I came across a free C++ newsletter published once or twice monthly by Glenn McClusky of Glenn McClusky & Associates, a consulting firm based in Fort Collins, Colorado specializing in Java, C++, and object-oriented technology.

Like myself, Glenn writes informative articles on C++. His newsletter contains regular discussions on new language features, performance hints, and moving from C to C++.

Most recently, he has discussed the new keywords *mutable* and *explicit*, which are just now being incorporated into today's compilers. In the back issues, there is an excellent discussion of namespaces. All of the back issues (since October 1995) are online at

- <http://rmi.net/~glenm/backcpp.html>

Glenn and his associates offer a Java newsletter as well.

### Borland C++ Newsletter

The folks at Borland provide a free C++ newsletter, as well as newsletters on Java, Paradox, Visual dBase, and Delphi. You can find back issues of these newsletters on Borland's Web site:

- <http://www.borland.com/feedback/archive/>

To subscribe, visit the Web page,

- <http://www.borland.com/feedback/listserv.html>

and fill out the form. Or if you prefer, you may simply send an e-mail message to **listserv@borland.com** containing **subscribe c++** in the body of the message.

### Usenet Newsgroups

There are really only three newsgroups within the Usenet hierarchy that are specific to C++:

- **comp.lang.c++**
- [comp.lang.c++.moderated](#)
- [comp.std.c++](#)

Other countries and organizations may have their own newsgroups devoted to C++. Check your local newsgroup hierarchies and find out.

## comp.lang.c++.moderated

Within the Usenet hierarchy, **comp.lang.\*** you can find a newsgroup devoted to most of the well-known (and some not so well-known!) languages. C++ is no exception, of which there are two newsgroups: **comp.lang.c++.moderated** and its predecessor, **comp.lang.c++**.

Those of you familiar with *large* newsgroups will know that you have to weed out almost eighty percent of the articles to get to the ``good stuff.'' There are countless cross-postings of articles that have no relevance to the newsgroup's focus.

Furthermore, so-called ``flame wars'' ensue when two parties feel strongly enough about their own opinions. All this mindless babble, although within the spirit of the free-speech centric Usenet community, makes the newsgroup less useful than it could be.

The newsgroup **comp.lang.c++** is one of those *large* groups. However, thanks to a group of enthusiastic C++ programmers, a new newsgroup was formed, **comp.lang.c++.moderated**. As the name implies, this group is moderated. This means that every article that is posted to the newsgroup must pass through the eyes of one of the several moderators before it can be presented to the newsgroup at large. You are guaranteed that every article that appears in **comp.lang.c++.moderated** will be about C++!

Kudos to all those involved with the formation and day-to-day operation of **comp.lang.c++.moderated**. The C++ community now has a forum which offers a place for C++ programmers to post questions and get excellent advice. No more weeding through mindless babble.

Many expert programmers are frequent readers of this newsgroup. This is an excellent forum and sounding board for your problems and issues you encounter while designing and programming with C++. Before posting, you should familiarize yourself with the information provided by the newsgroup moderators:

- Policy:  
<http://www.connobj.com/cpp/guide.htm>
- FAQ:  
<http://www.cerfnet.com/~mpcline/On-Line-C++-FAQs/>



Also, send a blank e-mail message to [c++-help@netlab.cs.rpi.edu](mailto:c++-help@netlab.cs.rpi.edu) or to [cpphelp@netlab.cs.rpi.edu](mailto:cpphelp@netlab.cs.rpi.edu) and you will receive a short "Help File" describing how to get further information. In particular, there are instructions on how to obtain further documents. In addition to the policy and FAQ documents above, other documents include: Nikki Locke's list of available C++ libraries; and a guide on posting articles to the newsgroup. New users should read the posting guide before submitting an article to the newsgroup. There is also a listing of other useful C++ pointers on the Web.

### comp.std.c++

The remaining Usenet newsgroup of relevance to C++ is **comp.std.c++**. This newsgroup is devoted to discussions of the C++ standards process, and the design and standardization of the C++ language and its libraries. You often hear about new language features here. If you follow the discussions, you will see many serious and expert programmers read this group. Many of them are probably on the ANSI C++ committee.

When the ANSI C++ standards committee is ready to submit a draft of the C++ standard for public review, a Committee Draft is released and announcement is posted to **comp.std.c++** and probably to **comp.lang.c++.moderated** as well. The last official release of the document was in April, 1995, as cited above.

For more information about **comp.std.c++**, follow these links:

- Charter:  
[http://reality.sgi.com/employees/austern\\_mti/std-c++/charter.html](http://reality.sgi.com/employees/austern_mti/std-c++/charter.html)
- Policy:  
[http://reality.sgi.com/employees/austern\\_mti/std-c++/policy.html](http://reality.sgi.com/employees/austern_mti/std-c++/policy.html)
- FAQ:  
[http://reality.sgi.com/employees/austern\\_mti/std-c++/faq.html](http://reality.sgi.com/employees/austern_mti/std-c++/faq.html)

### Miscellaneous Newsgroups

Since most C++ programmers practice object-oriented programming, another useful Usenet newsgroup is **comp.object**. Here you will find discussions about object-oriented design and object-oriented, object-oriented methods, and object-oriented programming. For more info, take a look at the FAQ:



- FAQ:

<http://iamwww.unibe.ch/~scg/OOinfo/FAQ/index.html>

The Usenet **comp** hierarchy is full of many newsgroups devoted to the many specialized sub-areas of computer science. There are probably other groups within the hierarchy that are relevant to your field and problem domain. Use your news reader to search for newsgroups in your domain.

## FAQs

A list of Frequently Asked Questions, or FAQ, is often put together to document the answers to the questions most often posted to a newsgroup. Newcomers to a newsgroup are encouraged to read the newsgroup's FAQ before posting questions.

A good place to look for the (ASCII text version) of the FAQ for your favorite newsgroup is to use `ftp` or point your Web browser at:

- <ftp://rtfm.mit.edu/pub/usenet-by-hierarchy/>

Below I list the FAQs most relevant to C++.

- [comp.lang.c++.moderated](#)

<http://www.cerfnet.com/~mpcline/On-Line-C++-FAQs/>

- [comp.std.c++](#)

[http://reality.sgi.com/employees/austern\\_mti/std-c++/faq.html](http://reality.sgi.com/employees/austern_mti/std-c++/faq.html)

- [comp.object](#)

<http://iamwww.unibe.ch/~scg/OOinfo/FAQ/index.html>

## Selected Web Resources

The Web, not surprisingly, is full of resources on C++. Below is my sampling of useful links.

### The C++ Virtual Library

The C++ Virtual Library at

- <http://info.desy.de/user/projects/C++.html>

contains a large collection of links devoted not only to C++ but also on such topics as object-oriented, parallel, and scientific programming. You will also find links to virtual C++ courses, coding style, conferences, and a host of other subjects. If you have not seen this impressive collection of C++ links, it is worth a look.

## STL

The Standard Template Library (STL) is the latest craving among C++ developers these days. Coming soon (if not already!) to a compiler near you! For a good introduction to the STL, see my column which appeared in February: ``An Introduction to the Standard Template Library," **Crossroads 2.3**, February, 1996 at

- <http://www.acm.org/crossroads/xrds2-3/ovp.html>

There is also a lot of information about the STL on the Web. Some good starting points are:

- <http://www.cs.rpi.edu/~musser/stl.html>
- <http://www.cs.rpi.edu/~wiseb/stl-notes.html>

## C++ - Beyond the ARM

Back in 1990, the definitive reference on C++ was ``the ARM" by Ellis and Stroustrup (see the full citation under the C++ Standard section above).

1990 is a long time ago, and the C++ standard has had several revisions since the ARM was published. The best place to find out details about the current standard is to read the standard itself as described in the April 1995 Committee Draft (again, see the full citation in the C++ Standard section above).

The standard is a little dense and not always explained in easily understandable terms. (At least for a non-compiler writers like most of us.) If you agree with me, you may like to take a look at Sean Corfield's excellent discussion of some of the more recent, ``post-ARM" language features at

- <http://www.corfield.org/index.php?fuseaction=cplusplus.main>

You will find descriptions of topics like namespaces, exceptions, run-time type identification (RTTI), bool, explicit, and mutable.

### Jon Morris Smith's C++ Resources Directory

By now you have noticed there is a large mass of information on C++ on the Web. But how do you find it? You either hope that someone has cataloged a list of useful links on one of their own Web pages from which you can make a bookmark, or you are stuck with one of the search engines such as Yahoo or Lycos.

A good example of the former category is Jon Morris Smith's extensive collection of C++ links at

- <http://www.cs.bham.ac.uk/~jdm/cpp.html>

Jon's Web page includes links to C++ courses, tutorials, books, magazines, publishers, documentation, compilers, libraries, extensions, and other information. Since Jon is in the UK, this page is embellished with links to many UK and European mirror sites for these C++ links when such mirrors exist. Those of you not in North America will appreciate the quick response these mirrors provide!

### Yahoo

If you still have not found the C++ link you are looking for, or are just interested in finding new C++ links to add to your own collection of links, try one of the Web search engines such as Yahoo. Yahoo provides a page on C++

- [http://www.yahoo.com/Computers\\_and\\_Internet/Programming\\_Languages/C\\_C\\_/](http://www.yahoo.com/Computers_and_Internet/Programming_Languages/C_C_/)

### Start Your Own Collection!

I urge you all to catalog your own C++ library of books, journals, electronic resources, and FAQs. Put up a Web page so that you can get at your references quickly and others can find it. If you find some area of C++ which needs further clarification, why not write it up and put it on the Web for all of us to read.

### An Apology

---

In a previous issue of OBJECTIVE VIEW POINT, [` ` Statics: Schizophrenia for C++ Programmers," Crossroads 2.4](#), April, 1996, I offended some of you with the choice of my title. Statics take on multiple meanings in C++. I made the allusion to [schizophrenia](#) because of my (incorrect) understanding that those that suffer from this disease take on multiple personalities.

Several of you objected to this stance, pointing out that my allusion was not only incorrect but also offensive and discriminating to those who may suffer from this disorder. Some of you even referred me to more accurate information on the Web! For an informative and educational description of this disorder, take a look at this page provided by the National Alliance for the Mentally Ill:

- <http://www.nami.org/disorder/disord8.htm#schiz>

I am truly sorry and had no intention of offending anyone. I was simply trying to come up with a catchy title for the column.

---