Preface

In the five years since the first edition of *STL Tutorial and Reference Guide* appeared, the C++ language standard has been finalized and officially accepted, C++ compiler vendors have made great progress in bringing their compilers into compliance with the standard, and dozens of other books and magazine articles have appeared that describe and explain the standardized language and libraries. Many of these books and articles have highlighted the Standard Template Library (STL) as the most significant addition to the standard. Some hailed it, as we did in this book's first edition, as having the potential to revolutionize the way a large number of people program. The past five years have already seen much of that potential realized, with the first edition of this book playing a key role for tens of thousands of programmers. We wrote in the preface of the first edition that there are five reasons why the STL components could become some of the most widely used software in existence:

- C++ is becoming one of the most widely used programming languages (in large part due to the support it provides for building and using component libraries).

- Since STL has been incorporated into the ANSI/ISO standard for C++ and its libraries, compiler vendors are making it part of their standard distributions.

- All components in STL are generic, meaning that they are adaptable (by language-supported compile-time techniques) to many different uses.

- The generality of STL components has been achieved without sacrificing efficiency.

- The design of STL components as fine-grained, interchangeable building blocks makes them a suitable basis for further development of components for specialized areas such as databases, user interfaces, and so forth.

We have enjoyed seeing these statements borne out by the developments of the past five years.

Changes in the Second Edition

In this new edition we have added substantially more tutorial material including expanded chapters in Part I on function objects and container, iterator, and function adaptors, and two entirely new chapters in Part II containing substantial new examples. We have also gone through all example code and surrounding discussion, including the reference material in Part III, to bring them up to date with the final standard. (Although some ambiguities in the standard have been discovered since it was finalized, we believe that in most cases the remaining uncertainties about the meaning of STL component specifications have no important consequences for the practicing programmer. In the few cases where they might, we point them out.) We also added a new chapter in Part III describing utility components such as the pair and comparison classes, and a new appendix describing the STL-related features of the standard string class.

In this edition we have also adopted the "literate programming" style for presenting example programs and code fragments. For readers unfamiliar with this approach to simultaneous programming and documenting, a brief explanation is given in Chapter 2 and more details are presented in Chapter 12. One benefit of the literate programming approach is that coding details can be presented once and then referred to (by name and page number) many times, so readers do not have to read through the same details repeatedly. Another major benefit is that we have been able check even more thoroughly than before that all code is syntactically and logically correct, since literate programming tools make it easy to extract the code directly from the manuscript and compile and test it. A list of the compilers the code has been compiled and tested with is given in Appendix D.

Some History, from the Preface to the First Edition

Virtually all C++ programmers know that this language was originated by one person, Bjarne Stroustrup, who began thinking of how to extend the C language to support definition of classes and objects as early as 1979. So too, the architecture of STL is largely the creation of one person, Alexander Stepanov.

It is interesting that it was also in 1979, at about the same time as Stroustrup's initial research, that Alex began working out his initial ideas of generic programming and exploring their potential for revolutionizing software development. Although Dave Musser had developed and advocated some aspects of generic programming as early as 1971, it was limited to a rather specialized area of software development (computer algebra). Alex recognized the full potential for generic programming and persuaded his then-colleagues at General Electric Research and Development (including, primarily, Dave Musser and Deepak Kapur) that generic programming should be pursued as a comprehensive basis for software development. But at that time there was no real support in any programming language for generic programming. The first major language to provide such support was Ada, with its generic units feature, and by 1987 Dave and Alex had developed and published an Ada library for list processing that embodied the results of much of their research on generic programming. However, Ada had not achieved much acceptance outside the defense industry, and C++ seemed more likely to become widely used and provide good support for generic programming, even though the language was relatively immature (it did not even have templates, added only later). Another reason for turning to C++, which Alex recognized early on, was that the C/C++ model of computation, which allows very flexible access to storage (via pointers), is crucial to achieving generality without losing efficiency.

Still, much research and experimentation were needed, not just to develop individual components, but more important to develop an overall architecture for a component library based on generic programming. First at AT&T Bell Laboratories and later at Hewlett-Packard Research Labs, Alex experimented with many architectural and algorithm formulations, first in C and later in C++. Dave Musser collaborated in this research, and in 1992 Meng Lee joined Alex's project at HP and became a major contributor.

This work undoubtedly would have continued for some time as just a research project or at best would have resulted in an HP proprietary library, if Andrew Koenig of Bell Labs had not become aware of the work and

asked Alex to present the main ideas at a November 1993 meeting of the ANSI/ISO committee for C++ standardization. The committee's response was overwhelmingly favorable and led to a request from Andy for a formal proposal in time for the March 1994 meeting. Despite the tremendous time pressure, Alex and Meng were able to produce a draft proposal that received preliminary approval at that meeting.

The committee had several requests for changes and extensions (some of them major), and a small group of committee members met with Alex and Meng to help work out the details. The requirements for the most significant extension (associative containers) had to be shown to be consistent by fully implementing them, a task Alex delegated to Dave Musser. It would have been quite easy for the whole enterprise to spin out of control at this point, but again Alex and Meng met the challenge and produced a proposal that received final approval at the July 1994 ANSI/ISO committee meeting. (Additional details of this history can be found in an interview Alex gave in the March 1995 issue of *Dr. Dobb's Journal*.)

Spreading the Word

Subsequently, the Stepanov and Lee document [**?**] was incorporated into the ANSI/ISO C++ draft standard ([**?**], parts of clauses 17 through 27). It also influenced other parts of the C++ Standard Library, such as the string facilities, and some of the previously adopted standards in those areas were revised accordingly.

In spite of STL's success with the committee, there remained the question of how STL would make its way into actual availability and use. With the STL requirements part of the publicly available draft standard, compiler vendors and independent software library vendors could of course develop their own implementations and market them as separate products or as selling points for their other wares. One of the first edition's authors, Atul Saini, was among the first to recognize the commercial potential and began exploring it as a line of business for his company, Modena Software Incorporated, even before STL had been fully accepted by the committee.

The prospects for early widespread dissemination of STL were considerably improved with Hewlett-Packard's decision to make its implementation freely available on the Internet in August 1994. This implementation, developed by Stepanov, Lee, and Musser during the standardization process, became the basis of all implementations offered by compiler and library vendors today.

Also in 1994, Dave Musser and Atul Saini developed the *STL++ Manual*, the first comprehensive user-level documentation of STL, but they soon recognized that an even more comprehensive treatment of STL was needed, one that would have better and more complete coverage of all aspects of the library. In an attempt to meet this goal, and with much encouragement and assistance from their editor, Mike Hendrickson, they wrote the first edition of this book.

In the second edition, the two original authors are joined by Gillmer J. Derge, President and CEO of the consulting firm Toltec Software Services, Inc. He has been developing applications with C++ for more than a decade, including seven years with General Electric Corporate R&D, where he received a Whitney Award for technical achievement.

Acknowledgments for the First Edition

We gratefully acknowledge the encouragement and assistance of many people. First and foremost, Alex Stepanov and Meng Lee offered continuous encouragement and were always available to help straighten out any misconceptions we had about the design of the library. Invaluable assistance with code development and testing was provided by several Modena staff members, including Atul Gupta, Kolachala Kalyan, and Narasimhan Rampalli. Several reviewers of earlier drafts gave us much valuable feedback and helped us find ways to present the most crucial ideas more clearly. They include Mike Ballantyne, Tom Cargill, Edgar Chrisostomo, Brian Kernighan, Scott Meyers, Larry Podmolik, Kathy Stark, Steve Vinoski, and John Vlissides. Others who also made valuable suggestions include Dan Benanav, Bob Cook, Bob Ingalls, Nathan Schimke, Kedar Tupil, and Rick Wilhelm. Finally, we thank the team at Addison-Wesley for their expert editorial and production assistance: Kim Dawley, Katie Duffy, Rosa Gonzalez, Mike Hendrickson, Simone Payment, Avanda Peters, John Wait, and Pamela Yee.

Acknowledgments for the Second Edition

For assistance with this edition, we wish first of all to thank the reviewers for pointing out errors in the discussion and examples and suggesting many other improvements in the presentation. The extensive comments of Max A. Lebow, Lawrence Rauchwerger, and Jan Christiaan van Winkel were especially helpful. We also thank Deborah Lafferty, our editor, and Julie De-

Baggis, who served as editor during the early planning of the second edition. Several other members of the production and marketing teams at Addison Wesley helped in many ways, including Jacquelyn Doucette, Chanda Leary-Coutu, Curt Johnson, Jennifer Lawinski, and Marty Rabinowitz.

D.R.M.
Loudonville, NY

G.J.D.
Cohoes, NY

A.S.
Los Gatos, CA

October 2000