Preface

When you wish to instruct,
be brief.
– Cicero

C++ feels like a new language. That is, I can express my ideas more clearly, more simply, and more directly today than I could in C++98 or C++11. Furthermore, the resulting programs are better checked by the compiler and run faster.

This book gives an overview of C++ as defined by C++20, the current ISO C++ standard, and implemented by the major C++ suppliers. In addition, it mentions a couple library components in current use, but not scheduled for inclusion into the standard until C++23.

Like other modern languages, C++ is large and there are a large number of libraries needed for effective use. This thin book aims to give an experienced programmer an idea of what constitutes modern C++. It covers most major language features and the major standard-library components. This book can be read in just a day or two but, obviously, there is much more to writing good C++ than can be learned in that amount of time. However, the aim here is not mastery, but to give an overview, to give key examples, and to help a programmer get started.

The assumption is that you have programmed before. If not, please consider reading a textbook, such as *Programming: Principles and Practice Using C++ (Second edition)* [Stroustrup,2014], before continuing here. Even if you have programmed before, the language you used or the applications you wrote may be very different from the style of C++ presented here.

Think of a sightseeing tour of a city, such as Copenhagen or New York. In just a few hours, you are given a quick peek at the major attractions, told a few background stories, and given some suggestions about what to do next. You do *not* know the city after such a tour. You do *not* understand all you have seen and heard; some stories may sound strange or even implausible. You do *not* know how to navigate the formal and informal rules that govern life in the city. To really know a city, you have to live in it, often for years. However, with a bit of luck, you will have gained a bit of an overview, a notion of what is special about the city, and ideas of what might be of interest to you. After the tour, the real exploration can begin.

This tour presents the major C++ language features as they support programming styles, such as object-oriented and generic programming. It does not attempt to provide a detailed, reference-manual, feature-by-feature view of the language. In the best textbook tradition, I try to explain a feature before I use it, but that is not always possible and not everybody reads the text strictly sequentially. I assume some technical maturity from my readers. So, the reader is encouraged to use the cross references and the index.

Similarly, this tour presents the standard libraries in terms of examples, rather than exhaustively. The reader is encouraged to search out additional and supporting material as needed. There is far more to the C++ ecosystem than just the facilities offered by ISO standard (e.g., libraries, build systems, analysis tools, and development environments). There is an enormous amount of material (of varying quality) available on the Web. Most readers will find useful tutorial and overview videos from conferences such as CppCon and Meeting C++. For technical details of the language and library offered by the ISO C++ standard, I recommend [Cppreference]. For example, when I mention a standard-library function or class, its definition can easily be looked up, and by examining its documentation, many related facilities can be found.

This tour presents C++ as an integrated whole, rather than as a layer cake. Consequently, I rarely identify language features as present in C, C++98, or later ISO standards. Such information can be found in Chapter 19 (History and Compatibility). I focus on fundamentals and try to be brief, but I have not completely resisted the temptation to overrepresent novel features, such as modules (§3.2.2), concepts (§8.2), and coroutines (§18.6). Slightly favoring recent developments also seems to satisfy the curiosity of many readers who already know some older version of C++.

A programming language reference manual or standard simply states what can be done, but programmers are often more interested in learning how to use the language well. This aspect is partly addressed in the selection of topics covered, partly in the text, and specifically in the advice sections. More advice about what constitutes good modern C++ can be found in the C++ Core Guidelines [Stroustrup,2015]. The Core Guidelines can be a good source for further exploration of the ideas presented in this book. You may note a remarkable similarity of the advice formulation and even the numbering of advice between the Core Guidelines and this book. One reason is that the first edition of *A Tour of C++* was a major source of the initial Core Guidelines.

Acknowledgments

Thanks to all who helped complete and correct the earlier editions of *A Tour of C++*, especially to the students in my ‘‘Design Using C++’’ course at Columbia University. Thanks to Morgan Stanley for giving me time to write this third edition. Thanks to Chuck Allison, Guy Davidson, Stephen Dewhurst, Kate Gregory, Danny Kalev, Gor Nishanov, and J.C. van Winkel for reviewing the book and suggesting many improvements.

This book was set using troff by the author using macros originating from Brian Kernighan.

Manhattan, New York *Bjarne Stroustrup*