1.3. Style and Structure of the Book

The C++ standard library provides components that are somewhat, but not totally, independent of one another, so there is no easy way to describe each part without mentioning others. I considered various approaches for presenting the contents of this book. One was on the order of the C++ standard. However, this is not the best way to explain the components of the C++ standard library from scratch. Another approach was to start with an overview of all components, followed by chapters that provided more details. Alternatively, I could have sorted the components, trying to find an order that had a minimum of cross-references to other sections. My solution was to use a mixture of all three approaches. I start with a brief introduction of the general concepts and the utilities that the library uses. Then, I describe all the components, each in one or more chapters. The first component is the standard template library (STL). There is no doubt that the STL is the most powerful, most complex, and most exciting part of the library. Its design influences other components heavily. Then, I describe the more self-explanatory components, such as special containers, strings, and regular expressions. The next component discussed is one you probably know and use already: the IOStream library. That component is followed by a discussion of internationalization, which had some influence on the IOStream library. Finally, I describe the library parts dealing with numerics, concurrency, and allocators.

Each component description begins with the component's purpose, design, and some examples. Next, a detailed description begins with various ways to use the component, as well as any traps and pitfalls associated with it. The description usually ends with a reference section, in which you can find the exact signature and definition of a component's classes and its functions.

List of Contents

The first five chapters introduce this book and the C++ standard library in general:

- **Chapter 1**: **About This Book** introduces the book's subject and describes its contents.
- **Chapter 2**: **Introduction to C++ and the Standard Library** provides a brief overview of the history of the C++ standard library and the context of its standardization and introduces the concept of complexity.
- **Chapter 3**: **New Language Features** provides an overview of the new language features you should know to read this book and to use the C++ standard library.
- **Chapter 4**: **General Concepts** describes the fundamental library concepts that you need to understand to work with all the components. In particular, the chapter introduces the namespace `std`, the format of header files, and the general support of error and exception handling.
- **Chapter 5**: **Utilities** describes several small utilities provided for the user of the library and for the library itself. In particular, the chapter describes classes `pair<>` and `tuple<>`, smart pointers, numeric limits, type traits and type utilities, auxiliary functions, class `ratio<>`, clocks and timers, and available C functions.

Chapters 6 through 11 describe all aspects of the STL:

- **Chapter 6**: **The Standard Template Library** presents a detailed introduction to the concept of the STL, which provides container classes and algorithms that are used to process collections of data. The chapter explains step-by-step the concept, the problems, and the special programming techniques of the STL, as well as the roles of its parts.
- **Chapter 7**: **STL Containers** explains the concepts and describes the abilities of the STL's container classes. The chapter describes arrays, vectors, deques, lists, forward lists, sets, maps, and unordered containers with their common abilities, differences, specific benefits, and drawbacks and provides typical examples.
- **Chapter 8**: **STL Container Members in Detail** lists and describes all container members (types and operations) in the form of a handy reference.
- **Chapter 9**: **STL Iterators** explains the various iterator categories, the auxiliary functions for iterators, and the iterator adapters, such as stream iterators, reverse iterators, insert iterators, and move iterators.
- **Chapter 10**: **STL Function Objects and Using Lambdas** details the STL's function object classes, including lambdas, and how to use them to define the behavior of containers and algorithms.
- **Chapter 11**: **STL Algorithms** lists and describes the STL's algorithms. After a brief introduction and comparison of the algorithms, each algorithm is described in detail, followed by one or more example programs.

Chapters 12 through 14 describe "simple" individual standard classes of the C++ standard library:

- **Chapter 12**: **Special Containers** describes the container adapters for queues and stacks, as well as the class `bitset`, which manages a bitfield with an arbitrary number of bits or flags.
- **Chapter 13**: **Strings** describes the string types of the C++ standard library (yes, there are more than one). The standard provides strings as "kind of" fundamental data types with the ability to use different types of characters.
- **Chapter 14**: **Regular Expressions** describes the interface to deal with regular expressions, which can be used to search and replace characters and substrings.

Chapters 15 and 16 deal with the two closely related subjects of I/O and internationalization:

- **Chapter 15**: **Input/Output Using Stream Classes** covers the standardized form of the commonly known IOStream library. The chapter also describes details that are typically not so well known but that may be important to programmers, such as the correct way to define and integrate special I/O channels.
- **Chapter 16**: **Internationalization** covers the concepts and classes for the internationalization of programs, such as the handling of different character sets and the use of different formats for floating-point numbers and dates.

The remaining chapters cover numerics, concurrency, and allocators:

- **Chapter 17**: **Numerics** describes the numeric components of the C++ standard library: in particular, classes for random numbers and distributions, types for complex numbers, and some numeric C functions.

- **Chapter 18**: **Concurrency** describes the features provided by the C++ standard library to enable and support concurrency and multithreading.
- **Chapter 19**: **Allocators** describes the concept of different memory models in the C++ standard library.

The book concludes with a **bibliography** and an **index**.

Due to the size of this book I had to move material that is not so relevant for the average application programmer but should be covered to a **supplementary chapter** provided on the Web site of this book: http://www.cppstdlib.com. That material includes:

- Details of bitsets (introduced in Section 12.5)
- Class `valarray<>` (very briefly introduced in Section 17.4)
- Details of allocators (introduced in Chapter 19)