Part 2: The C++ Language

Before beginning this part of the book you should have a fundamental grasp of computers, how they work, how to use yours, how to use your operating system's command console, and how compilers work. Lastly and most importantly, you should have obtained and installed some C++ software (text editor and compiler) that you know how to use with your computer's operating system. If you are not prepared by these standards, please read part 1 of this book which covers all these things.

There are three primary pieces, in my eyes, of C++: the language, the standard library, and the standard template library (STL). Your C++ software undoubtedly supports the language and contains its standard library, but the STL may be provided by a third party unrelated to your compiler.

The language of C++ itself is the key to the whole of C++. You can write poetry or legal documents when you're familiar with a written language, just as you can build any software you can imagine when you're familiar with a programming language. Thus, this part of the book focuses exclusively on the language itself and attempts to cover it in its entirety.

Small parts of the standard library are touched briefly for certain concepts because it's illustrative and sometimes necessary for easily doing some more complex things (e.g. reading and writing files). However, the STL is not covered at all in this part or even this book. The reason is that the STL is an *extension* of what I think of as *core* C++, or the language and the standard library.

Author's Opinion: Once you've learned C++ and you're comfortable with it then you won't need a book to *learn* the STL, you'll want a reference to use it effectively.