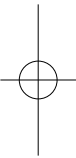
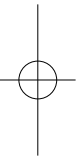
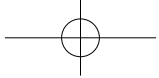


핵심 C++ 표준 라이브러리
The C++ Standard Library





The C++ Standard Library

by Rainer Grimm

Korean-language edition copyright © 2017 Insight Press

Copyright of the English edition:

© 2016 Rainer Grimm

The Korean edition was published by arrangement with Rainer Grimm through Agency-One, Seoul.

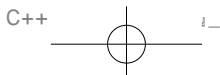
이 책의 한국어판 저작권은 에이전시 원을 통해 저작권자와의 독점 계약으로 인사이트에 있습니다.

저작권법에 의해 한국 내에서 보호를 받는 저작물이므로 무단전재와 무단복제를 금합니다.

핵심 C++ 표준 라이브러리: C++11, C++14 표준 반영

초판 1쇄 발행 2017년 6월 15일 **지은이** 라이너 그림 **옮긴이** 류광 **펴낸이** 한기성 **펴낸곳** 인사이트 **편집** 이지연 **제작·관리** 박미경 **표지출력** 소다미디어 **용지** 월드페이퍼 **인쇄** 현문인쇄 **후가공** 이지앤비 **제본** 자연재책 **등록번호** 제10-2313호 **등록일자** 2002년 2월 19일 **주소** 서울시 마포구 잔다리로 119 석우빌딩 3층 **전화** 02-322-5143 **팩스** 02-3143-5579 **블로그** <http://www.insightbook.co.kr> **이메일** insight@insightbook.co.kr **ISBN** 978-89-6626-309-7 책값은 뒤표지에 있습니다. 잘못 만들어진 책은 바꾸어 드립니다. 이 책의 정오표는 <http://www.insightbook.co.kr>에서 확인하실 수 있습니다. 이 도서의 국립중앙도서관 출판예정도서목록(CIP)은 서지정보유통지원시스템 홈페이지(<http://seoji.nl.go.kr>)와 국가자료공동목록시스템(<http://www.nl.go.kr/kolisnet>)에서 이용하실 수 있습니다.(CIP제어번호: CIP2017012356)

부록 Copyright © 2017 류광, 인사이트



인사이트

핵심 C++

표준 라이브러리
C++11, C++14 표준 반영

인사이트
리니어 그래픽 | 응용 | 플랫폼 | 운영
시스템

인사이트
insight

차례

C++11·C++14 일람	viii
옮긴이의 글	xii
서문	xv
1장 C++ 표준 라이브러리 소개	1
역사	1
개요	3
라이브러리 사용 방법	9
2장 편의 수단 라이브러리	15
유용한 함수들	15
함수 적응자	19
쌍	21
튜플	23
참조 래퍼	25
똑똑한 포인터	26
형식 특질	36
시간 라이브러리	42
3장 모든 컨테이너에 공통인 인터페이스	47
컨테이너의 생성과 삭제	48
컨테이너의 크기	49
컨테이너 요소 접근	50
배정과 교환	52
컨테이너 비교	53

4장	순차 컨테이너	55
	배열	56
	벡터	58
	데크	61
	목록	62
	전진 목록	63
5장	연관 컨테이너	67
	개요	67
	순서 있는 연관 컨테이너	70
	순서 없는 연관 컨테이너	75
6장	컨테이너 적응자	81
	스택	82
	대기열	83
	우선순위 대기열	84
7장	반복자	87
	범주	88
	반복자 생성	89
	유용한 함수들	90
	반복자 적응자	92
8장	호출 가능 단위	97
	함수	98
	함수 객체	98
	람다 함수	100
9장	알고리즘	101
	관례	102
	점착제로서의 반복자	104
	for_each 알고리즘	104

요소를 수정하지 않은 알고리즘	105
요소를 수정하는 알고리즘	111
분할	122
정렬	124
이진 검색	126
병합 연산	128
합	131
최대 최소	133
순열	134
수치	135
10장 수치 라이브러리	139
난수	139
C에서 물려받은 수치 함수들	143
11장 문자열 라이브러리	145
생성과 파괴	146
C++ 문자열과 C 문자열 사이의 변환	148
크기 대 용량	149
문자열 비교	150
문자열 연결	151
요소 접근	151
입력과 출력	152
검색	153
문자열 수정	156
문자열과 수치 사이의 변환	158
12장 정규 표현식 라이브러리	161
텍스트 문자 형식	162
정규 표현식 객체	163
부합 결과를 담은 match_results 객체	164
부합	168

검색	169
대체	170
서식화	171
반복 검색	172
13장 입출력 스트림 라이브러리	177
클래스 계통구조	178
입출력 함수	179
스트림	186
사용자 정의 자료 형식의 스트림 입출력	194
14장 스레드 지원 라이브러리	197
메모리 모형	197
원자적 자료 형식	198
스레드 적용	199
공유 변수	204
스레드 지역 자료	214
조건 변수	215
과제	218
부록 C++17 표준 라이브러리 소개	227
문자열에 대한 비소유 참조, <code>std::string_view</code>	228
없을 수도 있는 값을 나타내는 <code>std::optional</code>	230
아무 형식이나 담을 수 있는 <code>std::any</code>	232
형식에 안전한 공용체, <code>std::variant</code>	234
파일 시스템 라이브러리	236
알고리즘의 병렬화	238
새로 추가된 알고리즘	241
기타 변경 사항	243
참고자료	245
찾아보기	247

C++11·C++14 일람

C++11

TR1과 C++11	2
std::function과 std::bind 소개	3
std::pair와 std::tuple 소개	3
std::ref와 std::cref 소개	3
std::unique_ptr와 std::shared_ptr 소개	4
std::forward_list 소개	5
순서 없는 연관 컨테이너 소개	5
난수 라이브러리 소개	7
정규 표현식 소개	7
스레드 지원 라이브러리 소개	8
std::minmax	16
초기치 목록을 받는 std::min	16
초기치 목록을 받는 std::max	17
std::move	17
std::forward	18
std::bind	20
std::function	20
std::pair::swap	22
튜플	23
참조 래퍼	25
std::unique_ptr	28
std::shared_ptr	31
std::weak_ptr	33
형식 특질	36

시간 라이브러리	42
컨테이너 이동 생성	48
std::array의 생성	49
컨테이너 이동 배정	52
std::array의 분류	55
std::forward_list	55
순차 컨테이너의 메모리 해제와 shrink_to_fit	55
std::array 설명	57
순차 컨테이너 메모리 관리 메서드 shrink_to_fit	59
초기치 목록을 벡터에 배정	60
벡터의 emplace 메서드	60
벡터의 emplace_back 메서드	60
초기치 목록을 벡터에 삽입	60
std::forward_list	63
std::unordered_set	68
std::unordered_map	68
std::unordered_multiset	68
std::unordered_multimap	68
이동 가능(키의 조건)	71
이동 가능(값의 조건)	71
순서 없는 연관 컨테이너	75
std::unordered_map	89
std::map의 crbegin	89
std::map의 crend	89
반복자 정의 시 auto 활용	90
std::begin	91
std::end	91
std::prev	91
std::next	91
반복자 적응자 std::move_iterator	92
람다 함수	100
std::make_move_iterator	103

호출 가능 단위의 이동 반환	104
<code>std::find_if_not</code>	106
<code>std::find_first_of</code> 의 입력 반복자 매개변수	106
<code>std::all_of</code>	108
<code>std::any_of</code>	108
<code>std::none_of</code>	108
<code>std::copy_n</code>	111
<code>std::copy_if</code>	111
범위 이동 알고리즘들	116
<code>std::random_shuffle</code> 과 <code>std::shuffle</code>	120
<code>std::is_partitioned</code> , <code>std::partition_copy</code> , <code>std::partition_point</code>	123
<code>std::is_sorted</code>	125
<code>std::is_sorted_until</code>	126
<code>std::is_heap</code> 과 <code>std::is_heap_until</code>	132
<code>std::is_permutation</code>	135
<code>std::iota</code>	136
난수	139
<code>std::u16string</code> 과 <code>std::u32string</code>	146
문자열 이동 생성	147
초기치 목록으로 문자열 생성	147
<code>std::string::data</code> 의 널 문자 자동 추가	148
문자열 메모리 관리 메서드 <code>shrink_to_fit</code>	149
<code>std::string</code> 의 <code>front</code> 와 <code>back</code>	151
<code>std::string</code> 의 <code>pop_backv</code>	156
문자열과 수치 사이의 변환	158
<code>std::to_string</code> 과 <code>std::to_wstring</code>	159
<code>std::stoi</code> , <code>std::stol</code> , <code>std::stoll</code> , <code>std::stoul</code> , <code>std::stoull</code> , <code>std::stof</code> , <code>std::stod</code> , <code>std::stold</code>	159
정규 표현식	161
입출력 스트림 라이브러리 주요 변경 사항	177
<code>std::boolalpha</code> 와 <code>std::noboolalpha</code>	184
스레드 지원 라이브러리	197

C++14

C++14 소개	2
std::make_unique	30
::type 생략	42
기간(time duration) 내장 리터럴	45
std::make_reverse_iterator	91
std::rbegin과 std::rend	91
std::cbegin과 std::cend	91
std::crbegin과 std::crend	91
std::bit_xor와 std::bit_not	100
환경 갈무리 시 이동 의미론 적용	100
C++ 고유의 문자열 리터럴	147
std::shared_timed_mutex의 분류	207
std::shared_timed_mutex	212

옮긴이의 글

제목과 두께에서 짐작하듯이, 이 책은 이를테면 조슈티스^{Jossutis}의 *The C++ Standard Library: A Tutorial and Reference* 같은 두껍고 자세한 표준 라이브러리 책과는 성격이 다릅니다. 저자 서문에 나오듯이, 이 번역서의 원서는 “C++ 표준 라이브러리의 주요 구성요소를 간결하게 훑어보고 참조할 수 있게 하는” 것을 목표로 합니다. C++을 배운 지 얼마 되지 않은 초보자이든, C++의 문법과 관용구에는 능숙하지만 표준 라이브러리는 string, vector, list 정도만 사용해 본 숙련자이든, 방대하고 복잡한 C++ 표준 라이브러리의 세계에서 길을 잃지 않고 자신에게 필요한 것을 파악하는 데 이 책이 큰 도움이 될 것입니다.

번역서는 원서의 목표를 좀 더 확장해서, 기존 C++ 표준 라이브러리에 익숙한 독자가 C++11과 C++14에서 변한 것만 빠르게 찾아볼 수 있게 하는 데에도 노력을 들였습니다. 원서가 C++11과 C++14에서 표준에 새로 추가된 주요 요소들을 다루긴 하지만, 아쉽게도 그런 요소들이 C++11이나 C++14에서 새로이 추가되었다는 사실 자체는 언급하지 않는 경우가 종종 있습니다. 이를 보완하기 위해 번역서에는 적절한 위치에 [C++11] 아이콘과 [C++14] 아이콘을 배치했고, 그런 아이콘이 있는 페이지들의 목록을 차례 다음에 추가했습니다. 참고로, 버전 표시 아이콘들은 ‘중첩(cascading)’ 방식으로 배치되어 있습니다. 예를 들어 한 장(chapter)의 제목에 [C++11]이 붙어 있으면 기본적으로 그 장 전체가 C++11에서 변한 내용을 다루는 것입니다. 그러나 그 장의 특정 절(section)이나 문단 등에 [C++14]가 있으면 그 부분만큼은 C++14에 해당합니다. 말로 설명하자니 좀 장황하지만, 그냥 C++의 중괄호 범위(scope) 규칙을 연상하면 바로 이해가 될 것입니다.

원서에 비한 번역서의 또 다른 특징 하나는 C++17 표준 라이브러리를 소개하는 부록입니다. 원서는 ‘현재’ 최신 표준 라이브러리를 다루다 보니 조만간 C++ 공식 표준으로 채택될 C++17의 변경 사항을 반영하지 않았는데, 번역하면서 내내 이 점이 아쉬웠습니다. 그래서 제가 없는 지식이나마 끌어모아서 부록을 썼습니다. 아무쪼록 독자 여러분이 부록을 통해서 C++17 표준 라이브러리를 주마간산 격으로라도 파악할 수 있길 바랄 뿐입니다.

본문이나 부록이나 간결함을 중요시하다 보니 좀 더 설명이 필요하다 싶은 부분이 있을 것입니다. 또한, 항상 그렇듯이 오역이나 내용상의 오류도 있을 수 있습니다. 혹시 본문이나 부록에서 좀 더 논의하고 싶거나 뭔가 이상한 점을 발견했다면, 제 홈페이지 *occam's Razor* (<http://occamsrazr.net/>)를 활용해 주세요. 홈페이지의 ‘번역서 정보’ 페이지를 통해 접근할 수 있는 이 책 페이지에 의견을 나눌 수 있는 공간이 있고, 정오표와 추가적인 참고 자료도 등록될 것입니다.

잘 알려지지 않은 원서의 번역서 출판을 과감하게 결심하고 제게 번역을 맡겨 주신 도서출판 인사이트 한기성 사장님, 저와 출판사 사이의 의사소통을 원활하게 진행해 주신 이지연 편집자님, 그리고 조판과 교정 등 이 책을 지금과 같은 모습으로 만드는 데 노력하신 모든 분께 감사 인사 전합니다. 또한, 원서 내용 확인과 부록 작성에 영문 위키백과와 *cppreference.com*이 큰 도움이 되었다는 말도 빼먹을 수 없겠습니다. 끝으로, 번역 원고를 꼼꼼히 살피면서 *splice*를 *slice*로 착각하는 황당한 실수를 비롯한 여러 오역과 오타를 잡아 준 아내 오현숙에게 사랑과 감사의 마음을 전합니다.

옴긴이 류광

서문

본서 『핵심 C++ 표준 라이브러리』는 C++의 현재 표준인 C++14 ISO/IEC 14882:2014¹의 표준 라이브러리를 요약한 참고서이다. C++14는 C++11을 소폭 수정, 확장했다. 1,300페이지가 넘는 C++11 명세서는 2011년에 나왔는데, 이는 최초의 C++ 표준인 C++98이 나온 지 13년이 지난 때이다. 물론 그사이에 C++03이 2003년에 발표되었지만, C++03은 그냥 버그 교정 릴리스로 간주된다.

이 요약 참고서의 목표는 독자가 C++ 표준 라이브러리의 주요 구성요소를 간결하게 훑어보고 참조할 수 있게 하는 것이다. 이 책은 독자가 C++ 자체에 익숙하다고 가정한다. 그런 독자라면 이 책의 장점을 최대한 누릴 수 있을 것이다. C++에 익숙하지 않은 독자라면 먼저 C++ 언어의 핵심 내용에 관한 책을 읽기 바란다. 핵심 언어[†]에 관한 책을 충분히 익힌 후라면, 이 책을 통해서 한층 더 높은 수준으로 발돋움할 수 있을 것이다. 독자의 학습 편의를 위해 이 책은 이론과 실재를 연결해 주는 짧은 코드 예제들을 많이 제공한다.

¹ http://www.iso.org/iso/catalogue_detail.htm?csnumber=64029

[†] (옮긴이) 핵심 언어(core language)는 C++ 표준 관련 문헌들에서 자주 등장하는 용어로, 특정 라이브러리에 의존하지 않는 C++ 언어 자체의 구성요소와 기능을 말한다. 아주 간단히 말해서, 어떠한 헤더 파일도 포함시키지 않고 사용할 수 있는 것이면 핵심 언어에 속한다.

조판 관례

이 책에는 다음과 같이 그리 많지 않은 조판 관례가 쓰였다.

특별한 글꼴

돋움 또는 *italic*

중요한 용어나 개념은 돋움 또는 *italic* 글꼴로 표시한다.

고정폭

코드, 명령어, 키워드, 그리고 변수, 함수, 클래스 이름은 고정폭 글꼴로 표시한다.

특별한 정보

특별한 정보나 팁, 주의 사항은 다음과 같이 표시한다.

✓ 정보 제목
정보 내용.

💡 팁 제목
팁 내용.

! 주의 사항 제목
주의 사항 내용.

예제 코드

원래 필자는 `using` 지시문이나 `using` 선언을 좋아하지 않는다. 그런 것을 사용하면 코드에 쓰인 함수나 클래스, 객체의 기원(소속된 이름공간)이 숨겨지기 때문이다. 그러나 지면 제한 때문에 이 책에서는 `using` 지시문/선언을 사용할 수밖에 없었다. 대신 가능한 한 최소한으로만 사

용했다. `using` 지시문(이름테면 `using namespace std;`)과 `using` 선언(이름테면 `using std::cout;`)에서 기원을 즉시 알아낼 수 있도록 직접적인 형태로만 사용했으므로, 예제 코드를 읽는 데 큰 어려움이 없을 것이다.

각 예제 코드에는 해당 부분이 다루는 기능성과 관련된 헤더들만 나온다. 예제 코드의 출력을 표시할 때 부울 값은 `true` 또는 `false`로 표시했지만, 간결함을 위해 `std::boolalpha(p.184)`를 적용하는 코드는 생략했다.†

값과 객체

C에서 물려받은 C++의 기본 자료 형식의 인스턴스는 값(value)이라고 부른다. 좀 더 복잡한 형식(기본 형식들로 이루어진 경우가 많다)의 인스턴스는 객체(objects)라고 부른다. 객체들은 대체로 사용자 정의 형식(user defined type)이나 컨테이너의 인스턴스이다.

C++과 C++11

C++11의 특징을 가장 잘 말해줄 수 있는 사람은 C++을 만든 비야네 스트롭스트롭^{Bjarne Stroustrup} 자신일 것이다. 그는 이렇게 말했다.

놀랍게도 C++11은 새로운 언어처럼 느껴진다. 언어의 구성요소들이 예전보다 더 잘 맞물리며, 고수준 프로그래밍 스타일이 전보다 자연스럽게 전혀 없이 효율적이다.

— 비야네 스트롭스트롭, <http://www.stroustrup.com/C++11FAQ.html>

비야네 스트롭스트롭이 전적으로 옳다. C++11은 새로운 언어처럼 느껴지는데, 이는 고전적인 C++에 비해 수많은 것이 추가되었기 때문이다. 핵심 언어에 대해 옳은 것은 물론이고, 개선되고 확장된 표준 라이브러리에 대해서는 더욱 옳다. 텍스트 조작을 위한 정규 표현식 라이

† (옮긴이) 따라서 해당 코드를 실행하면 실제로는 `true` 또는 `false`가 아니라 1 또는 0이 출력된다. 실제로 `true` 또는 `false`가 출력되게 하려면 `std::cout << std::boolalpha << some_bool_val;` 형태의 코드를 사용해야 한다.

브리리, 형식의 조회·비교·조작을 위한 형식 특질 라이브러리, 새로운
난수 라이브러리와 시간 관련 라이브러리는 모두 C++11에 새로이 추가
된 것들이다. 그러나 그것이 전부는 아니다. 자동 메모리 관리를 위한
똑똑한 포인터들이 추가되었고 `std::array`라는 컨테이너와 `std::tuple`
이라는 편의용 템플릿도 추가되었는데, 이들은 C++14에서 더욱 개선되
었다. 그리고 C++11은 C++ 역사 최초로 다중 스레드를 인식하고 스레
드 지원 라이브러리를 제공한다.

감사의 글

우선 독어판 C++ *Standardbibliothek*²의 감수(proofreading)를 진행한
O'Reilly의 알렉산드라 폴레니우스^{Alexandra Follenius}에게 감사한다. 짐작했
겠지만 이 책(원서)은 그 독어판을 기초로 한 것이다. 알렉산드라 외에
도 카르슈텐 아네르트^{Karsten Ahnert}, 군트람 베르티^{Guntram Berti}, 트미트리
가뉴우스^{Dmitry Ganyushin}, 스벤 요한젠^{Sven Johannsen}, 토르슈텐 로비츠키
^{Torsten Robitzki}, 바르트 판데보이스튀네^{Bart Vandewoestyne}, 펠릭스 빈터^{Felix Winter}
가 독어판 C++ *Standardbibliothek*의 아주 훌륭한 감수자로 일해 주었다.
그들 모두 아주 고맙다.

이 책을 영어로 옮기면서 나는 내 영어 블로그 www.ModernesCpp.com
에서 원고의 감수를 요청했는데³, 예상보다 훨씬 많은 분이 참여해
주었다. 첫 감수자인 내 아들 마리우스를 포함해서 그들 모두에게 특별
한 감사의 뜻을 표한다.

다음은 감수자들이다(알파벳순): 마헤시 아타르데^{Mahesh Attarde}, 릭
오뎃^{Rick Audet}, 피트 배로^{Pete Barrow}, 마이클 벤-데이비드^{Michael Ben-David}, 데

† (옮긴이) 물론 C++ 표준 위원회가 그때까지 스레드의 존재를 모르고 있었다거나 지원 요구를 무
시했던 것은 아니다. 스레드를 비롯한 병렬성/동시성 관련 기능은 아키텍처와 운영체제에 의존
적인 측면이 많은데, C++이 쓰이는 다양한 아키텍처와 운영체제를 널리 아우르는 적절한 다중
스레드 지원 접근방식이 무르익는 데 시간이 걸렸다고 보는 것이 타당할 것이다.

2 <http://shop.oreilly.com/product/9783955619688.do>

3 <http://www.modernescpp.com/index.php/do-you-wan-t-to-proofread-a-book>

이브 번스^{Dave Burns}, 알바로 페르난데스^{Alvaro Fernandez}, 조지 하케^{George Haake}, 클레어 매크레이^{Clare Macrae}, 아르네 메르츠^{Arne Mertz}, 제이슨 터너^{Jason Turner}, 이반 베르길리예프^{Ivan Vergiliev}.

서평

아마도 C++ 표준 라이브러리의 가장 중요한 부분을 가장 빠르고도 쉬운 방식으로 익히는 수단이 바로 이 책일 것이다. 모든 현대적 C++ 프로그래머에게 추천한다.

— 릭 오렛, Dolby Laboratories의 선임 기술자

1장

T h e C + + S t a n d a r d L i b r a r y

C++ 표준 라이브러리 소개

C++ 표준 라이브러리는 다수의 구성요소로 이루어져 있다. 이번 장의 목적은 두 가지이다. 첫째는 표준 라이브러리의 주요 구성요소를 개괄하는 것이고, 또 하나는 그것들의 기본적인 용도와 용법을 제시하는 것이다.

역사

C++은 역사가 긴 언어이고, 따라서 C++ 표준 라이브러리의 역사도 길다. C++은 지난 세기 후반, 좀 더 구체적으로는 1980년대에 처음 등장했으며, 가장 최근의 표준은 2014년에 나왔다. 소프트웨어 개발을 하는 사람이라면 소프트웨어의 영역이 얼마나 빠르게 발전하는지 잘 알고 있을 것이며, 따라서 30년이 얼마나 긴 시간인지도 알 것이다. 그런 만큼, 초창기 C++의 구성요소들, 이를테면 I/O 스트림 같은 것들의 설계에 깔린 사고방식이 현대적인 표준 템플릿 라이브러리(Standard Template Library, STL)의 설계에 깔린 사고방식과 완전히 다른 것도 놀랄 일은 아니다. 지난 30년간 소프트웨어 개발 분야에서 일어난 진화와 발전은 C++ 표준 라이브러리에서도 볼 수 있다. 사실, C++ 표준 라이브러리는 소프트웨어 문제들의 해결 방식에서 발생한 진화의 중요한 사례이기도 하다. 객체 지향적 언어로 시작한 C++은 이후 STL과 함께 일반적 프

그래밍(generic programming)을 도입했으며, 이제는 함수형 프로그래밍의 여러 개념도 받아들인 상태이다.



그림 1.1 C++ 연표

1998년에 나온 첫 번째 C++ 표준 라이브러리의 구성요소는 세 가 지였다. 하나는 앞에서 언급한 I/O 스트림(주로 파일 처리를 위한)이고 또 하나는 문자열 라이브러리, 나머지 하나는 표준 템플릿 라이브러리(STL)이다. 표준 템플릿 라이브러리를 이용하면 알고리즘을 컨테이너에 투명하게 적용하기가 쉬워진다.

시간이 흘러, 2005년에 TR1(Technical Report 1; 기술 보고서 1)이 나왔다. C++ 표준 라이브러리의 확장을 담은 TR1(구체적인 명칭은 ISO/IEC TR 19768, C++ Library Extensions)은 공식적인 표준은 아니었 C++11지만, 구성요소들의 거의 대부분이 C++11의 일부가 되었다. TR1에는 예를 들어 정규 표현식, 똑똑한 포인터(smart pointer), 해시 테이블, 난수, 시간을 위한 라이브러리들이 포함되었는데, 이들은 해당 Boost 라이브러리(<http://www.boost.org/>)에 기초한 것들이었다.

TR1을 공식적으로 표준화한 것 외에, C++11의 표준 라이브러리에 는 완전히 새로운 구성요소도 하나 포함되었다. 바로 다중 스레드 적용(multithreading)을 위한 스레드 지원 라이브러리이다.

C++14 C++14는 C++11 표준의 작은 개정판일 뿐이었다. 그래서 C++14의 표준 라이브러리는 기존의 똑똑한 포인터, 튜플, 형식 특질(type trait), 스레드 지원 라이브러리를 조금 개선한 정도이다.

2 1장 C++ 표준 라이브러리 소개