



# C++ Concurrency IN ACTION

Practical Multithreading

Anthony Williams

 MANNING

<b>About This Book.....</b>	<b>1</b>
Roadmap.....	1
Who should read this book.....	2
How to use this book.....	2
Code conventions and downloads.....	2
Software requirements.....	3
Author Online.....	3

## *about this book*

---

This book is an in-depth guide to the concurrency and multithreading facilities from the new C++ Standard, from the basic usage of `std::thread`, `std::mutex`, and `std::async`, to the complexities of atomic operations and the memory model.

### **Roadmap**

The first four chapters introduce the various library facilities provided by the library and show how they can be used.

Chapter 5 covers the low-level nitty-gritty of the memory model and atomic operations, including how atomic operations can be used to impose ordering constraints on other code, and marks the end of the introductory chapters.

Chapters 6 and 7 start the coverage of higher-level topics, with some examples of how to use the basic facilities to build more complex data structures—lock-based data structures in chapter 6, and lock-free data structures in chapter 7.

Chapter 8 continues the higher-level topics, with guidelines for designing multithreaded code, coverage of the issues that affect performance, and example implementations of various parallel algorithms.

Chapter 9 covers thread management—thread pools, work queues, and interrupting operations.

Chapter 10 covers testing and debugging—types of bugs, techniques for locating them, how to test for them, and so forth.

The appendixes include a brief description of some of the new language facilities introduced with the new standard that are relevant to multithreading, the

implementation details of the message-passing library mentioned in chapter 4, and a complete reference to the C++11 Thread Library.

### **Who should read this book**

If you're writing multithreaded code in C++, you should read this book. If you're using the new multithreading facilities from the C++ Standard Library, this book is an essential guide. If you're using alternative thread libraries, the guidelines and techniques from the later chapters should still prove useful.

A good working knowledge of C++ is assumed, though familiarity with the new language features is not—these are covered in appendix A. Prior knowledge or experience of multithreaded programming is not assumed, though it may be useful.

### **How to use this book**

If you've never written multithreaded code before, I suggest reading this book sequentially from beginning to end, though possibly skipping the more detailed parts of chapter 5. Chapter 7 relies heavily on the material in chapter 5, so if you skipped chapter 5, you should save chapter 7 until you've read it.

If you've not used the new C++11 language facilities before, it might be worth skimming appendix A before you start to ensure that you're up to speed with the examples in the book. The uses of the new language facilities are highlighted in the text, though, and you can always flip to the appendix if you encounter something you've not seen before.

If you have extensive experience with writing multithreaded code in other environments, the beginning chapters are probably still worth skimming so you can see how the facilities you know map onto the new standard C++ ones. If you're going to be doing any low-level work with atomic variables, chapter 5 is a must. Chapter 8 is worth reviewing to ensure that you're familiar with things like exception safety in multithreaded C++. If you have a particular task in mind, the index and table of contents should help you find a relevant section quickly.

Once you're up to speed on the use of the C++ Thread Library, appendix D should continue to be useful, such as for looking up the exact details of each class and function call. You may also like to dip back into the main chapters from time to time to refresh your use of a particular construct or look at the sample code.

### **Code conventions and downloads**

All source code in listings or in text is in a fixed-width font like this to separate it from ordinary text. Code annotations accompany many of the listings, highlighting important concepts. In some cases, numbered bullets link to explanations that follow the listing.

Source code for all working examples in this book is available for download from the publisher's website at [www.manning.com/CPlusPlusConcurrencyinAction](http://www.manning.com/CPlusPlusConcurrencyinAction).

## Software requirements

To use the code from this book unchanged, you'll need a recent C++ compiler that supports the new C++11 language features used in the examples (see appendix A), and you'll need a copy of the C++ Standard Thread Library.

At the time of writing, g++ is the only compiler I'm aware of that ships with an implementation of the Standard Thread Library, although the Microsoft Visual Studio 2011 preview also includes an implementation. The g++ implementation of the Thread Library was first introduced in a basic form in g++ 4.3 and extended in subsequent releases. g++ 4.3 also introduced the first support for some of the new C++11 language features; more of the new language features are supported in each subsequent release. See the g++ C++11 status page for details.<sup>1</sup>

Microsoft Visual Studio 2010 provides some of the new C++11 language features, such as rvalue references and lambda functions, but doesn't ship with an implementation of the Thread Library.

My company, Just Software Solutions Ltd, sells a complete implementation of the C++11 Standard Thread Library for Microsoft Visual Studio 2005, Microsoft Visual Studio 2008, Microsoft Visual Studio 2010, and various versions of g++.<sup>2</sup> This implementation has been used for testing the examples in this book.

The Boost Thread Library<sup>3</sup> provides an API that's based on the C++11 Standard Thread Library proposals and is portable to many platforms. Most of the examples from the book can be modified to work with the Boost Thread Library by judicious replacement of `std::` with `boost::` and use of the appropriate `#include` directives. There are a few facilities that are either not supported (such as `std::async`) or have different names (such as `boost::unique_future`) in the Boost Thread Library.

## Author Online

Purchase of *C++ Concurrency in Action* includes free access to a private web forum run by Manning Publications where you can make comments about the book, ask technical questions, and receive help from the author and from other users. To access the forum and subscribe to it, point your web browser to [www.manning.com/CPlusPlusConcurrencyinAction](http://www.manning.com/CPlusPlusConcurrencyinAction). This page provides information on how to get on the forum once you're registered, what kind of help is available, and the rules of conduct on the forum.

Manning's commitment to our readers is to provide a venue where a meaningful dialogue between individual readers and between readers and the author can take place. It's not a commitment to any specific amount of participation on the part of the author, whose contribution to the book's forum remains voluntary (and unpaid). We suggest you try asking the author some challenging questions, lest his interest stray!

The Author Online forum and the archives of previous discussions will be accessible from the publisher's website as long as the book is in print.

<sup>1</sup> GNU Compiler Collection C++0x/C++11 status page, <http://gcc.gnu.org/projects/cxx0x.html>.

<sup>2</sup> The `just::thread` implementation of the C++ Standard Thread Library, <http://www.stdthread.co.uk>.

<sup>3</sup> The Boost C++ library collection, <http://www.boost.org>.