Introduction to the Boost C++ Libraries

Volume III – Algorithms, Concurrency, Generic Programming and Higher-order Functions, Numerics

Publisher: Datasim Press

Author: Daniel J. Duffy

January 2018

Goals of Book

This book is a follow-up to two popular books on Boost C++ libraries (Demming and Duffy 2010, Demming and Duffy 2012). We introduce forty libraries that were not discussed in the two previous volumes. Of particular interest is that approximately ten of these libraries are also supported in the new C++ standard that we have described in detail in Duffy 2018.

The approach taken in this hands-on book is based on a standard format for each chapter that describes a specific library as described below. Regarding content, the goal is to get the reader up to speed as it were so that it becomes possible to compile and run simple examples without errors as quickly as possible. Once we understand how each library works we continue by discussing more extended and complex functionality in the library. Finally, we discuss how the library can be used with modern software design patterns and applications.

The book has an associated web site where readers and author discuss the book and exchange ideas on using the libraries in applications.

If you are interested in takin this course (in original or customised form) please contact us at info@datasim.nl

Subjects Covered

- 40 Boost C++ libraries classified into seven major categories.
- Data containers and algorithms.
- Concurrent and parallel programming.
- Libraries for higher-order functions and the functional programming model.
- Generic programming and Template Metaprogramming.
- Integration with modern software design patterns and Domain Architectures.

Benefits

- Learn Boost C++ libraries in an easy to follow, step-by-step manner.
- Each library can be independently studied; minimal inter-library dependence.
- Each library is documented at basic, intermediate and advanced levels.
- Chapters are documented according to a software design patterns style (similar to that in GOF 1995).
- Access to full source code in the book.
- Exercises and mini-projects to consolidate the skills learned.

For whom is this Book?

Software developers who wish to use industrial-strength Boost C++ libraries in their applications. Instead of writing your own home-grown code it is possible to deploy robust and tested library code. We also discuss approximately 10 libraries that have found their way into the C++11 standard and this is useful for developers whose applications do not yet support C++11/C++14/C++17 because they will be ready for the upgrades when their applications move to a new version of the language.

Publication Date: Q1 2010.

Contents

Structure of Chapters

Since we discuss approximately forty Boost libraries and each library offers substantial functionality we found it necessary to document these libraries in a standardised fashion similar to how software design patterns are described. In this way, we hope to achieve the following goals:

- *Understandability*: the effort needed to understand the intent and applicability of the library. This could entail the ability to create a simple example in approximately 20 minutes.
- *Learnability*: the effort that is needed in order to learn how to find and use the main functionality in the library.
- *Operability*: the effort needed to customise the library and use it with other libraries and applications.

We see that knowledge of a library proceeds in increments, from basic to expert levels. The chapters are structured accordingly to ease the learning process:

1. Intent of Library
2. Motivating Examples
3. General Applicability and Context
4. The Core Process: From Input to Output
5. Variants and Exception Handling
6. Examples
7. Alternative Solutions
8. Relationships with other Libraries
9. Integration with Software Design Patterns
10. Application Areas and Support in other Languages

11. Exercises and Projects

Each chapter can be read independently of the other chapters in principle. This means that the amount of forward and backward referencing Is reduced to a minimum. Furthermore, the code in each chapter is graded in such a way that we begin with simple examples before moving to more complex examples, software design patterns and mini-applications.

References

Demming, R. and Duffy, D. 2010 Introduction to the Boost C++ Libraries, Volume I Datasim Press.
Demming, R. and Duffy, D. 2012 Introduction to the Boost C++ Libraries Volume II Datasim Press.
Duffy, D. J. 2004 Domain Architectures John Wiley and Sons Chichester.
Duffy, D.J. 2018 Financial Instrument Pricing using C++ Second Edition John Wiley and Sons Chichester.
GOF 1995 Gamma, E., Helm, R., Johnson, R. and Vlissides, J. Design Patterns, Addison-Wesley.

Author

Daniel J. Duffy started the company Datasim in 1987 to promote C++ as a new object-oriented language for developing applications in the roles of developer, architect and requirements analyst to help clients design and analyse software systems for Computer Aided Design (CAD), process control and hardware-software systems, logistics, holography (optical technology) and computational finance. He used a combination of top-down functional decomposition and bottom-up object-oriented programming techniques to create stable and extendible applications (for a discussion, see Duffy 2004 where we have grouped applications into domain categories). Previous to Datasim he worked on engineering applications in oil and gas and semiconductor industries using a range of numerical methods (for example, the finite element method (FEM)) on mainframe and mini-computers.

Daniel Duffy has BA, MSc and PhD degrees in pure and applied mathematics and has been active in promoting partial differential equation (PDE) and finite difference methods (FDM) for applications in computational finance. He was responsible for the introduction of the Fractional Step (Soviet Splitting) method and the Alternating Direction Explicit (ADE) method in computational finance. He is also the inventor of exponential fitting for time-dependent partial differential equations.

He is also the originator of two very popular C++ online courses (both C++98 and C++11/14) on www.quantnet.com in cooperation with Quantnet LLC and Baruch College (CUNY), NYC. He also trains developers and designers around the world. He can be contacted dduffy@datasim.nl