1. Record Nr. UNINA9910511471303321

Autore	Mukherjee Arindam
Titolo	Learning Boost C++ libraries : solve practical programming problems using powerful, portable, and expressive libraries from Boost / / Arindam Mukherjee
Pubbl/distr/stampa	Birmingham, [England] ; ; Mumbai, [India] : , : Packt Publishing, , 2015 ©2015
ISBN	1-78355-122-4
Edizione	[1st edition]
Descrizione fisica	1 online resource (558 p.)
Collana	Community Experience Distilled
Disciplina	005.13/3 005.133
Soggetti	C++ (Computer program language) Computer software - Development Electronic books.
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Description based upon print version of record.
Nota di bibliografia	Includes bibliographical references at the end of each chapters and index.
Nota di contenuto	Cover; Copyright; Credits; About the Author; About the Reviewers; www.PacktPub.com; Table of Contents; Preface; Chapter 1: Introducing Boost; How it all started; What is Boost?; Getting started with Boost libraries; Necessary software; Linux toolchain; Windows toolchain; Obtaining and building Boost libraries; Planning your Boost sandbox; Library naming conventions; Library name components; Library name layouts; Installing a Boost binary distribution; Building and installing the Boost libraries from source; Using Boost libraries in your projects; Linking against Boost libraries on Linux Linking against Boost libraries on WindowsBuilding the code listings in this book; CMake; Code examples; Self-test questions; Summary; Chapter 2: The First Brush with Boost's Utilities; Simple data structures; Boost.Optional; Accessing values stored in boost::optional; get_value_or; Boost.Optional versus pointers; Boost.Tuple; Creating tuples; Accessing tuple elements; Comparing tuples; Writing generic code using tuples; Working with heterogeneous values; Boost.Variant; Accessing values in a variant; Defining recursive variants; Boost.Any; Boost.Conversion; lexical_cast

Handling command-line argumentsDesigning command-line options; The diff command - a case study; Using Boost.Program_Options; Parsing positional parameters; Multiple option values; Other utilities and compile-time checks; BOOST_CURRENT_FUNCTION; Boost.Swap; Compile-time asserts; Diagnostics using preprocessor macros; Self-test questions; Summary; References; Chapter 3: Memory Management and Exception Safety; Dynamic memory allocation and exception safety; Exception safety and RAII; Smart pointers; Unique ownership semantics; boost::scoped_ptr; boost::scoped_array; std::unique_ptr
Shared ownership semanticsboost::shared_ptr and std::shared_ptr; Intrusive smart pointers - boost::intrusive_ptr; shared_array; Managing non-memory resources using smart pointers; Self-test questions; Summary; References; Chapter 4: Working with Strings; Text processing with Boost String Algorithms library; Using Boost String Algorithms; Find algorithms; Case-conversion and trimming algorithms; The replace and erase algorithms; The split and join algorithms; Splitting text using the Boost Tokenizer library; Tokenizing based on separators Tokenizing records with fields containing metacharactersTokenizing records with fixed-length fields; Writing your own tokenizer functions; Regular expressions using Boost.Regex; Regular expression syntax; Atoms; Quantifiers; Character classes; Anchors; Sub-expressions; Disjunctions; Using Boost.Regex to parse regular expressions; Matching text; Searching text; Tokenizing text using regex; Replacing text; Self-test questions; Summary; Chapter 5: Effective Data Structures beyond STL; Boost Container library; Move-awareness and in-place construction; Nonstandard containers
Flat associative containers

Sommario/riassunto

If you are a C++ programmer who has never used Boost libraries before, this book will get you up-to-speed with using them. Whether you are developing new C++ software or maintaining existing code written using Boost libraries, this hands-on introduction will help you decide on the right library and techniques to solve your practical programming problems.