1. Record Nr. UNINA9910300639903321

Autore	Horton Ivor
Titolo	Using the C++ Standard Template Libraries / / by Ivor Horton
Pubbl/distr/stampa	Berkeley, CA : , : Apress : , : Imprint : Apress, , 2015
ISBN	9781484200049 1484200047
Edizione	[1st ed. 2015.]
Descrizione fisica	1 online resource (504 p.)
Collana	Expert's Voice in C++
Disciplina	004
Soggetti	Programming languages (Electronic computers) Software engineering Programming Languages, Compilers, Interpreters Software Engineering/Programming and Operating Systems
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Contents at a Glance; Contents; About the Author; About the Technical Reviewer; Acknowledgments; Introduction; Chapter 1: Introducing the Standard Template Library; Basic Ideas; Templates; The Containers ; Iterators ; Obtaining Iterators; Iterator Categories; Stream Iterators; Iterator Adaptors ; Reverse Iterators; Insert Iterators; Move Iterators; Operations on Iterators ; Smart Pointers ; Using unique_ptr Pointers ; Resetting unique_ptr Objects; Comparing and Checking unique_ptr Objects; Using shared_ptr Pointers ; Resetting shared_ptr Objects Comparing and Checking shared_ptr Objects weak_ptr Pointers; Algorithms ; Passing a Function as an Argument; Function Objects; Lambda Expressions ; Naming a Lambda Expression; Passing a Lambda Expression to a Function ; The Capture Clause ; Summary; Chapter 2: Using Sequence Containers; The Sequence Containers; Function Members That Are Common Between Containers; Using array Containers; Accessing Elements ; Using Iterators with array Containers ; Comparing array Containers; Using vector Containers; Creating vector Containers; The Capacity and Size of a Vector Accessing Elements Using Iterators with a vector Container; Adding New Elements to a vector Container; Appending Elements ; Inserting Elements ; Deleting Elements ; vector Containers ; Using deque

Containers; Creating deque Containers; Accessing Elements ; Adding and Removing Elements ; Replacing the Contents of a deque Container; Using a list Container; Creating list Containers; Adding Elements ; Removing Elements ; Sorting and Merging Elements ; Accessing Elements ; Using forward_list Containers; Defining Your Own Iterators; STL Iterator Requirements
A Problem with Using STL Iterators The STL Approach ; Using the Iterator Template; STL Iterator Member Function Requirements ; Summary; Chapter 3: Container Adapters; What Are Container Adapters?; Creating and Using a stack Container Adapter; Stack Operations ; Creating and Using a queue Container Adapter; Queue Operations ; A Practical Use of a Queue Container; Using a priority_queue Container Adapter; Creating a Priority Queue; Operations for a Priority Queue; Heaps; Creating a Heap; Heap Operations ; Storing Pointers in a Container
Storing Pointers in Sequence Containers Storing Pointers in a Priority Queue ; Heaps of Pointers; Containers of Base Class Pointers ; Applying Algorithms to a Range of Pointers; Summary; Chapter 4: Map Containers; Introducing Map Containers; Using a map Container; Creating a map Container; Inserting Elements in a map; Constructing map Elements in Place; Accessing Elements in a map; Deleting Elements; Using pair and tuple Objects; Operations with a pair; Operations with a tuple; tuples and pairs in Action; Using a multimap Container; Changing the Comparison Function
Using a greater Object

Sommario/riassunto

Using the C++ Standard Template Libraries is a contemporary treatment that teaches the generic programming capabilities that the C++ 14 Standard Library provides. In this book, author Ivor Horton explains what the class and function templates available with C++ 14 do, and how to use them in a practical context. You'll learn how to create containers, and how iterators are used with them to access, modify, and extend the data elements they contain. You'll also learn about stream iterators that can transfer data between containers and streams, including file streams. The function templates that define algorithms are explained in detail, and you'll learn how to pass function objects or lambda expressions to them to customize their behavior. Many working examples are included to demonstrate how to apply the algorithms with different types of containers. After reading this book, you will understand the scope and power of the templates that the C++ 14 Standard Library includes and how these can greatly reduce the coding and development time for many applications. You'll be able to combine the class and function templates to great effect in dealing with real-world problems. The templates in the Standard Library provide you as a C++ programmer with a comprehensive set of efficiently implemented generic programming tools that you can use for most types of application. How to use Standard Library templates with your C++ applications. Understand the different types of containers that are available and what they are used for. How to define your own class types to meet the requirements of use with containers. What iterators are, the characteristics of the various types of iterators, and how they allow algorithms to be applied to the data in different types of container. How you can define your own iterator types. What the templates that define algorithms do, and how you apply them to data stored in containers and arrays. How to access hardware clocks and use them for timing execution. How to use the templates available for compute-intensive numerical data processing. How to create and use pseudo-random number generators with distribution objects.