

Username: Pralay Patoria **Book:** The C++ Standard Library: A Tutorial and Reference, Second Edition. No part of any chapter or book may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher of the book or chapter. Redistribution or other use that violates the fair use privilege under U.S. copyright laws (see 17 USC107) or that otherwise violates these Terms of Service is strictly prohibited. Violators will be prosecuted to the full extent of U.S. Federal and Massachusetts laws.

Chapter 6. The Standard Template Library

The heart of the C++ standard library — the part that influenced its overall architecture — is the *standard template library* (STL). The STL is a generic library that provides solutions to managing collections of data with modern and efficient algorithms. It allows programmers to benefit from innovations in the area of data structures and algorithms without needing to learn how they work.

From the programmer's point of view, the STL provides a bunch of collection classes that meet various needs, together with several algorithms that operate on them. All components of the STL are templates, so they can be used for arbitrary element types. But the STL does even more: It provides a framework for supplying other collection classes or algorithms for which existing collection classes and algorithms work. All in all, the STL gives C++ a new level of abstraction. Forget programming dynamic arrays, linked lists, binary trees, or hash tables; forget programming different search algorithms. To use the appropriate kind of collection, you simply define the appropriate container and call the corresponding member functions and algorithms to process the data.

The STL's flexibility, however, has a price, chief of which is that it is not self-explanatory. Therefore, the subject of the STL fills several chapters in this book. This chapter introduces the general concept of the STL and explains the programming techniques needed to use it. The first examples show how to use the STL and what to consider while doing so. [Chapters 7](#) through [11](#) discuss the components of the STL (containers, iterators, function objects, and algorithms) in detail and present several more examples.