

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 15. Input/Output Using Stream Classes

The classes for I/O form an important part of the C++ standard library; a program without I/O is not of much use. The I/O classes from the C++ standard library are not restricted to files or to screen and keyboard but instead form an extensible framework for the formatting of arbitrary data and access to arbitrary "external representations."

The *IOStream library*, as the classes for I/O are called, is the only part of the C++ standard library that was widely used prior to the standardization of C++98. Early distributions of C++ systems came with a set of classes, developed at AT&T, that established a de facto standard for doing I/O. Although these classes have undergone several changes to fit consistently into the C++ standard library and to suit new needs, the basic principles of the IOStream library remain unchanged.

This chapter first presents a general overview of the most important components and techniques, and then demonstrates in detail how the IOStream library can be used in practice. Its use ranges from simple formatting to the integration of new external representations, a topic that is often addressed improperly.

This chapter does not attempt to discuss all aspects of the IOStream library in detail; to do that would take an entire book by itself. For details not found here, please consult one of the books that focus on the I/O stream library or the reference manual of the C++ standard library.

Many thanks to Dietmar Kühl, an expert on I/O and internationalization in the C++ standard library, who gave valuable feedback and wrote initial parts of this chapter.

Recent Changes with C++11

C++98 specified almost all features of the IOStream library. Here is a list of the most important features added with C++11:

- A few new manipulators were introduced: `hexfloat` and `defaultfloat` ([see Section 15.7.6, page 788](#)), as well as `get_time()` and `put_time()` ([see Section 16.4.3, page 890](#)) and `get_money()` and `put_money()` ([see Section 16.4.2, page 882](#)).
- In order to provide more information about an exception, the class for exceptions is now derived from `std::system_error` rather than directly from `std::exception` ([see Section 15.4.4, page 762](#)).
- String stream and file stream classes now support rvalue and move semantics, so you can move construct, move assign, and swap a string stream or a file stream. This also provides the ability to use temporary string or file streams for I/O. [See Section 15.9.2, page 795](#), and [Section 15.10.2, page 806](#).
- File streams now also allow you to pass a `std::string` for the filename rather than only a `const char*` ([see Section 15.9.1, page 794](#)).
- The output and input operators `<<` and `>>` are now also overloaded for `long long` and `unsigned long long`.
- I/O streams now partially support concurrency ([see Section 15.2.2, page 752](#)).
- Character traits are now also provided for types `char16_t` and `char32_t` ([see Section 16.1.4, page 853](#)).
- With the help of the new class `wbuffer_convert`, you can let streams read and write different character sets, such as UTF-8 ([see Section 16.4.4, page 903](#)).