C++
Information
Tutorials
Reference
Articles
Forum

Reference
C library:
Containers:
Input/Output:
<fstream>
<iomanip>
<ios>
<iosfwd>
<iostream>
<istream>
<ostream>
<sstream>
<streambuf>
Multi-threading:
Other:

<ostream>
class templates:
basic_ostream
classes:
ostream
wostream
manipulators:
endl
ends
flush

ostream
ostream::ostream
ostream::~ostream
member classes:
ostream::sentry
member functions:
ostream::flush
ostream::operator<<
ostream::put
ostream::seekp
ostream::tellp
ostream::write
non-member overloads:
operator<< (ostream)
protected members:
ostream::operator= C++11
ostream::swap C++11

public member function

std::ostream::ostream

<ostream> <iostream>

C++98 C++11

```
initialization (1)  explicit ostream (streambuf* sb);
```

Construct object

Constructs an `ostream` object.

(1) inititalization constructor
Assigns initial values to the components of its base classes by calling the inherited member `ios::init` with *sb* as argument.

(2) copy constructor (deleted)
Deleted: no copy constructor.

(3) move constructor (protected)
Acquires the contents of *x*, except its associated *stream buffer*: It calls `ios::move` to transfer *x*'s internal components inherited from `ios`. *x* is left with its associated *stream buffer* unchanged and not *tied* (all other components of *x* are in an unspecified but valid state after the call).

Parameters

sb
Pointer to a `streambuf` object.

x
Another `ostream` object.

Example

```
1  // ostream constructor
2  #include <iostream>     // std::cout, std::ostream, std::ios
3  #include <fstream>      // std::filebuf
4
5  int main () {
6    std::filebuf fb;
7    fb.open ("test.txt",std::ios::out);
8    std::ostream os(&fb);
9    os << "Test sentence\n";
10   fb.close();
11   return 0;
12 }
```

This code uses a `filebuf` object (derived from `streambuf`) to open the file `test.txt`. The buffer is then passed as parameter to the `ostream` constructor, associating it to the stream.

Objects of class `ostream` are seldom constructed directly. Generally some derived class is used (like the standard `ofstream` and `ostringstream`).

Data races

The object pointed by *sb* may be accessed and/or modified.

Exception safety

If an exception is thrown, the only side effects may come from accessing/modifying *sb*.

See also

ios::init	Initialize object (protected member function)