

Accredited Standards Committee*
X3, INFORMATION PROCESSING SYSTEMS

Doc No: X3J16/95-0088
WG21/N0688
Date: 28 April 1995
Project: Programming Language C++
Reply to: Andrew Koenig
AT&T Bell Laboratories
PO Box 636
600 Mountain Avenue
Room 2C-306
Murray Hill, NJ 07974 USA
ark@research.att.com

Concordance for July 1995 (Monterey) C++ Working Paper

This document provides a cross-reference between absolute section numbers and the corresponding symbolic names used for cross-references within the Working Paper. Its purpose is to make it easier to replace absolute section numbers by their corresponding names, which in turn will make it easier to renumber sections in future versions of the Working Paper.

* *Operating under the procedures of the American National Standards Institute (ANSI)*
Standards Secretariat: CBEMA, 1250 Eye Street NW, Suite 200, Washington DC 20005

Listing by clause number

1	intro	General
1.1	intro.scope	Scope
1.2	intro.refs	Normative references
1.3	intro.defs	Definitions
1.4	syntax	Syntax notation
1.5	intro.memory	The C++ memory model
1.6	intro.object	The C++ object model
1.7	intro.compliance	Processor compliance
1.8	intro.execution	Program execution
2	lex	Lexical conventions
2.1	lex.phases	Phases of translation
2.2	lex.trigraph	Trigraph sequences
2.3	lex.pptoken	Preprocessing tokens
2.4	lex.digraph	Alternative tokens
2.5	lex.token	Tokens
2.6	lex.comment	Comments
2.7	lex.name	Identifiers
2.8	lex.key	Keywords
2.9	lex.literal	Literals
2.9.1	lex.icon	Integer literals
2.9.2	lex.ccon	Character literals
2.9.3	lex.fcon	Floating literals
2.9.4	lex.string	String literals
2.9.5	lex.bool	Boolean literals
3	basic	Basic concepts
3.1	basic.def	Declarations and definitions
3.2	basic.def.odr	One definition rule
3.3	basic.scope	Declarative regions and scopes
3.3.1	basic.scope.local	Local scope
3.3.2	basic.scope.proto	Function prototype scope
3.3.3		Function scope
3.3.4	basic.scope.namespace	Namespace scope
3.3.5	basic.scope.class	Class scope
3.3.6	basic.scope.hiding	Name hiding
3.3.7	basic.scope.exqual	Explicit qualification
3.3.8	basic.scope.elab	Elaborated type specifier
3.3.9	basic.scope.pdecl	Point of declaration
3.4	class.scope	Name look up
3.5	basic.link	Program and linkage
3.6	basic.start	Start and termination
3.6.1	basic.start.main	Main function
3.6.2	basic.start.init	Initialization of non-local objects
3.6.3	basic.start.term	Termination
3.7	basic.stc	Storage duration
3.7.1	basic.stc.static	Static storage duration
3.7.2	basic.stc.auto	Automatic storage duration
3.7.3	basic.stc.dynamic	Dynamic storage duration
3.7.3.1	basic.stc.dynamic.allocation	Allocation functions
3.7.3.2	basic.stc.dynamic.deallocation	Deallocation functions
3.7.4	basic.stc.inherit	Duration of sub-objects
3.8	basic.life	Object Lifetime
3.9	basic.types	Types

3.9.1	basic.fundamental	Fundamental types
3.9.2	basic.compound	Compound types
3.9.3	basic.type.qualifier	CV-qualifiers
3.9.4	basic.type.name	Type names
3.10	basic.lval	Lvalues and rvalues
4	conv	Standard conversions
4.1	conv.lval	Lvalue-to-rvalue conversion
4.2	conv.array	Array-to-pointer conversion
4.3	conv.func	Function-to-pointer conversion
4.4	conv.qual	Qualification conversions
4.5	conv.prom	Integral promotions
4.6	conv.fpprom	Floating point promotion
4.7	conv.integral	Integral conversions
4.8	conv.double	Floating point conversions
4.9	conv.fpint	Floating-integral conversions
4.10	conv.ptr	Pointer conversions
4.11	conv.mem	Pointer to member conversions
4.12	conv.class	Base class conversion
4.13	conv.bool	Boolean conversions
5	expr	Expressions
5.1	expr.prim	Primary expressions
5.2	expr.post	Postfix expressions
5.2.1	expr.sub	Subscripting
5.2.2	expr.call	Function call
5.2.3	expr.type.conv	Explicit type conversion (functional notation)
5.2.4	expr.ref	Class member access
5.2.5	expr.post.incr	Increment and decrement
5.2.6	expr.dynamic.cast	Dynamic cast
5.2.7	expr typeid	Type identification
5.2.8	expr.static.cast	Static cast
5.2.9	expr.reinterpret.cast	Reinterpret cast
5.2.10	expr.const.cast	Const cast
5.3	expr.unary	Unary expressions
5.3.1	expr.unary.op	Unary operators
5.3.2	expr.pre.incr	Increment and decrement
5.3.3	expr.sizeof	Sizeof
5.3.4	expr.new	New
5.3.5	expr.delete	Delete
5.4	expr.cast	Explicit type conversion (cast notation)
5.5	expr.mptr.oper	Pointer-to-member operators
5.6	expr.mul	Multiplicative operators
5.7	expr.add	Additive operators
5.8	expr.shift	Shift operators
5.9	expr.rel	Relational operators
5.10	expr.eq	Equality operators
5.11	expr.bit.and	Bitwise AND operator
5.12	expr.xor	Bitwise exclusive OR operator
5.13	expr.or	Bitwise inclusive OR operator
5.14	expr.log.and	Logical AND operator
5.15	expr.log.or	Logical OR operator
5.16	expr.cond	Conditional operator
5.17	expr.ass	Assignment operators
5.18	expr.comma	Comma operator
5.19	expr.const	Constant expressions
6	stmt.stmt	Statements

6.1	stmt.label	Labeled statement
6.2	stmt.expr	Expression statement
6.3	stmt.block	Compound statement or block
6.4	stmt.select	Selection statements
6.4.1	stmt.if	The <code>if</code> statement
6.4.2	stmt.switch	The <code>switch</code> statement
6.5	stmt.iter	Iteration statements
6.5.1	stmt.while	The <code>while</code> statement
6.5.2	stmt.do	The <code>do</code> statement
6.5.3	stmt.for	The <code>for</code> statement
6.6	stmt.jump	Jump statements
6.6.1	stmt.break	The <code>break</code> statement
6.6.2	stmt.cont	The <code>continue</code> statement
6.6.3	stmt.return	The <code>return</code> statement
6.6.4	stmt.goto	The <code>goto</code> statement
6.7	stmt.dcl	Declaration statement
6.8	stmt.ambig	Ambiguity resolution
7	dcl.dcl	Declarations
7.1	dcl.spec	Specifiers
7.1.1	dcl.stc	Storage class specifiers
7.1.2	dcl.fct.spec	Function specifiers
7.1.3	dcl.typedef	The <code>typedef</code> specifier
7.1.4	dcl.friend	The <code>friend</code> specifier
7.1.5	dcl.type	Type specifiers
7.1.5.1	dcl.type.cv	The <i>cv-qualifiers</i>
7.1.5.2	dcl.type.simple	Simple type specifiers
7.1.5.3	dcl.type.elab	Elaborated type specifiers
7.2	dcl.enum	Enumeration declarations
7.3	basic.namespace	Namespaces
7.3.1	namespace.def	Namespace definition
7.3.1.1	namespace.qual	Explicit qualification
7.3.1.2	namespace.unnamed	Unnamed namespaces
7.3.1.3	namespace.scope	Namespace scope
7.3.1.4	namespace.memdef	Namespace member definitions
7.3.2	namespace.alias	Namespace or class alias
7.3.3	namespace.udecl	The <code>using</code> declaration
7.3.4	namespace.udir	Using directive
7.4	dcl.asm	The <code>asm</code> declaration
7.5	dcl.link	Linkage specifications
8	dcl.decl	Declarators
8.1	dcl.name	Type names
8.2	dcl.ambig.res	Ambiguity resolution
8.3	dcl.meaning	Meaning of declarators
8.3.1	dcl.ptr	Pointers
8.3.2	dcl.ref	References
8.3.3	dcl.mptr	Pointers to members
8.3.4	dcl.array	Arrays
8.3.5	dcl.fct	Functions
8.3.6	dcl.fct.default	Default arguments
8.4	dcl.fct.def	Function definitions
8.5	dcl.init	Initializers
8.5.1	dcl.init.aggr	Aggregates
8.5.2	dcl.init.string	Character arrays
8.5.3	dcl.init.ref	References
9	class	Classes

9.1	class.name	Class names
9.2	class.mem	Class members
9.3	class.scope0	Scope rules for classes
9.4	class.mfct	Member functions
9.4.1	class.mfct.nonstatic	Nonstatic member functions
9.4.2	class.this	The <code>this</code> pointer
9.5	class.static	Static members
9.5.1	class.static.mfct	Static member functions
9.5.2	class.static.data	Static data members
9.6	class.union	Unions
9.7	class.bit	Bit-fields
9.8	class.nest	Nested class declarations
9.9	class.local	Local class declarations
9.10	class.nested.type	Nested type names
10	class.derived	Derived classes
10.1	class.mi	Multiple base classes
10.2	class.member.lookup	Member name lookup
10.3	class.virtual	Virtual functions
10.4	class.abstract	Abstract classes
11	class.access	Member access control
11.1	class.access.spec	Access specifiers
11.2	class.access.base	Access specifiers for base classes
11.3	class.access.dcl	Access declarations
11.4	class.friend	Friends
11.5	class.protected	Protected member access
11.6	class.access.virt	Access to virtual functions
11.7	class.paths	Multiple access
12	special	Special member functions
12.1	class.ctor	Constructors
12.2	class temporary	Temporary objects
12.3	class.conv	Conversions
12.3.1	class.conv.ctor	Conversion by constructor
12.3.2	class.conv.fct	Conversion functions
12.4	class.dtor	Destructors
12.5	class.free	Free store
12.6	class.init	Initialization
12.6.1	class.expl.init	Explicit initialization
12.6.2	class.base.init	Initializing bases and members
12.7	class.ctor	Construction and destruction
12.8	class.copy	Copying class objects
13	over	Overloading
13.1	over.load	Overloadable declarations
13.2	over.dcl	Declaration matching
13.3	over.match	Overload resolution
13.3.1	over.match.funcs	Candidate functions and argument lists
13.3.1.1	over.match.call	Function call syntax
13.3.1.1.1	over.call.func	Call to named function
13.3.1.1.2	over.call.object	Call to object of class type
13.3.1.2	over.match.oper	Operators in expressions
13.3.1.3	over.match.user	Initialization by user-defined conversions
13.3.1.4	over.match.ctor	Initialization by constructor
13.3.2	over.match.viable	Viable functions
13.3.3	over.match.best	Best Viable Function
13.3.3.1	over.best.ics	Implicit conversion sequences
13.3.3.1.1	over.ics.scs	Standard conversion sequences

13.3.3.1.2	over.ics.user	User-defined conversion sequences
13.3.3.1.3	over.ics.ellipsis	Ellipsis conversion sequences
13.3.3.1.4	over.ics.ref	Reference binding
13.3.3.2	over.ics.rank	Ranking implicit conversion sequences
13.4	over.over	Address of overloaded function
13.5	over.oper	Overloaded operators
13.5.1	over.unary	Unary operators
13.5.2	over.binary	Binary operators
13.5.3	over.ass	Assignment
13.5.4	over.call	Function call
13.5.5	over.sub	Subscripting
13.5.6	over.ref	Class member access
13.5.7	over.inc	Increment and decrement
13.6	over.built	Built-in operators
14	temp	Templates
14.1	temp.names	Template names
14.2	temp.res	Name resolution
14.2.1	temp.local	Locally declared names
14.2.2	temp.encl	Names from the template's enclosing scope
14.2.3	temp.dep	Dependent names
14.2.4	temp.inject	Non-local names declared within a template
14.3	temp.inst	Template instantiation
14.3.1	temp.linkage	Template linkage
14.3.2	temp.point	Point of instantiation
14.3.3	temp.opref	Instantiation of operator->
14.4	temp.explicit	Explicit instantiation
14.5	temp.spec	Template specialization
14.6	temp.class.spec	Class template specializations
14.6.1	temp.class.spec.match	Matching of class template specializations
14.6.2	temp.class.order	Partial ordering of class template specializations
14.7	temp.param	Template parameters
14.8	temp.arg	Template arguments
14.9	temp.type	Type equivalence
14.10	temp.fct	Function templates
14.10.1	temp.arg.explicit	Explicit template argument specification
14.10.2	temp.deduct	Template argument deduction
14.10.3	temp.over	Overload resolution
14.10.4	temp.over.link	Overloading and linkage
14.10.5	temp.over.spec	Overloading and specialization
14.10.6	temp.func.order	Partial ordering of function templates
14.11	temp.mem.func	Member function templates
14.12	temp.friend	Friends
14.13	temp.static	Static members and variables
15	except	Exception handling
15.1	except.throw	Throwing an exception
15.2	except.ctor	Constructors and destructors
15.3	except.handle	Handling an exception
15.4	except.spec	Exception specifications
15.5	except.special	Special functions
15.5.1	except.terminate	The terminate() function
15.5.2	except.unexpected	The unexpected() function
15.6	except.access	Exceptions and access
16	cpp	Preprocessing directives
16.1	cpp.cond	Conditional inclusion
16.2	cpp.include	Source file inclusion

16.3	cpp.replace	Macro replacement
16.3.1	cpp.subst	Argument substitution
16.3.2	cpp.stringize	The # operator
16.3.3	cpp.concat	The ## operator
16.3.4	cpp.rescan	Rescanning and further replacement
16.3.5	cpp.scope	Scope of macro definitions
16.4	cpp.line	Line control
16.5	cpp.error	Error directive
16.6	cpp.pragma	Pragma directive
16.7	cpp.null	Null directive
16.8	cpp.predefined	Predefined macro names
17	lib.library	Library introduction
17.1	lib.definitions	Definitions
17.2	lib.description	Method of description (Informative)
17.2.1	lib.structure	Structure of each subclause
17.2.1.1	lib.structure.summary	Summary
17.2.1.2	lib.structure.requirements	Requirements
17.2.1.3	lib.structure.specifications	Specifications
17.2.1.4	lib.structure.see.also	C Library
17.2.2	lib.conventions	Other conventions
17.2.2.1	lib.type.descriptions	Type descriptions
17.2.2.1.1	lib.enumerated.types	Enumerated types
17.2.2.1.2	lib.bitmask.types	Bitmask types
17.2.2.1.3	lib.character.seq	Character sequences
17.2.2.1.3.1	lib.byte.strings	Byte strings
17.2.2.1.3.2	lib.multibyte.strings	Multibyte strings
17.2.2.1.3.3	lib.wide.characters	Wide-character sequences
17.2.2.2	lib.functions.within.classes	Functions within classes
17.2.2.3	lib.objects.within.classes	Private members
17.3	lib.requirements	Library-wide requirements
17.3.1	lib.organization	Library contents and organization
17.3.1.1	lib.contents	Library contents
17.3.1.2	lib.headers	Headers
17.3.1.3	lib.compliance	Freestanding implementations
17.3.2	lib.using	Using the library
17.3.2.1	lib.using.headers	Headers
17.3.2.2	lib.using.linkage	Linkage
17.3.3	lib.constraints	Constraints on programs
17.3.3.1	lib.reserved.names	Reserved names
17.3.3.1.1	lib.macro.names	Macro names
17.3.3.1.2	lib.global.names	Global names
17.3.3.1.3	lib.extern.names	External linkage
17.3.3.2	lib.alt.headers	Headers
17.3.3.3	lib.derived.classes	Derived classes
17.3.3.4	lib.replacement.functions	Replacement functions
17.3.3.5	lib.handler.functions	Handler functions
17.3.3.6	lib.res.on.functions	Other functions
17.3.3.7	lib.res.on.arguments	Function arguments
17.3.4	lib.conforming	Conforming implementations
17.3.4.1	lib.res.on.headers	Headers
17.3.4.2	lib.res.on.macro.definitions	Restrictions on macro definitions
17.3.4.3	lib.global.functions	Global functions
17.3.4.4	lib.member.functions	Member functions
17.3.4.5	lib.reentrancy	Reentrancy
17.3.4.6	lib.protection.within.classes	Protection within classes

17.3.4.7	lib.derivation	Derived classes
17.3.4.8	lib.res.on.exception.handling	Restrictions on exception handling
18	lib.language.support	Language support library
18.1	lib.support.types	Types
18.2	lib.support.limits	Implementation properties
18.2.1	lib.limits	Numeric limits
18.2.1.1	lib.numeric.limits	Template class <code>numeric_limits</code>
18.2.1.2	lib.numeric.limits.members	<code>numeric_limits</code> members
18.2.1.3	lib.round.style	Type <code>float_round_style</code>
18.2.1.4	lib.numeric.special	<code>numeric_limits</code> specializations
18.2.2	lib.c.limits	C Library
18.3	lib.support.start.term	Start and termination
18.4	lib.support.dynamic	Dynamic memory management
18.4.1	lib.new.delete	Storage allocation and deallocation
18.4.1.1	lib.new.delete.single	Single-object forms
18.4.1.2	lib.new.delete.array	Array forms
18.4.1.3	lib.new.delete.placement	Placement forms
18.4.2	lib.alloc.errors	Storage allocation errors
18.4.2.1	lib.bad.alloc	Class <code>bad_alloc</code>
18.4.2.2	lib.new.handler	Type <code>new_handler</code>
18.4.2.3	lib.set.new.handler	<code>set_new_handler</code>
18.5	lib.support.rtti	Type identification
18.5.1	lib.type.info	Class <code>type_info</code>
18.5.2	lib.bad.cast	Class <code>bad_cast</code>
18.5.3	lib.bad.typeid	Class <code>bad_typeid</code>
18.6	lib.support.exception	Exception handling
18.6.1	lib.exception.unexpected	Violating <i>exception-specifications</i>
18.6.1.1	lib.bad.exception	Class <code>bad_exception</code>
18.6.1.2	lib.unexpected.handler	Type <code>unexpected_handler</code>
18.6.1.3	lib.set.unexpected	<code>set_unexpected</code>
18.6.1.4	lib.unexpected	<code>unexpected</code>
18.6.2	lib.exception.terminate	Abnormal termination
18.6.2.1	lib.terminate.handler	Type <code>terminate_handler</code>
18.6.2.2	lib.set.terminate	<code>set_terminate</code>
18.6.2.3	lib.terminate	<code>terminate</code>
18.7	lib.support.runtime	Other runtime support
19	lib.diagnostics	Diagnostics library
19.1	lib.std.exceptions	Exception classes
19.1.1	lib.exception	Class <code>exception</code>
19.1.2	lib.logic.error	Class <code>logic_error</code>
19.1.3	lib.domain.error	Class <code>domain_error</code>
19.1.4	lib.invalid.argument	Class <code>invalid_argument</code>
19.1.5	lib.length.error	Class <code>length_error</code>
19.1.6	lib.out.of.range	Class <code>out_of_range</code>
19.1.7	lib.runtime.error	Class <code>runtime_error</code>
19.1.8	lib.range.error	Class <code>range_error</code>
19.1.9	lib.overflow.error	Class <code>overflow_error</code>
19.2	lib.assertions	Assertions
19.3	lib.errno	Error numbers
20	lib.utilities	General utilities library
20.1	lib allocator.requirements	Allocator requirements
20.2	lib.utility	Utility components
20.2.1	lib.operators	Operators
20.2.2	lib.pairs	Pairs
20.3	lib.function.objects	Function objects

20.3.1	lib.base	Base
20.3.2	lib.arithmetic.operations	Arithmetic operations
20.3.3	lib.comparisons	Comparisons
20.3.4	lib.logical.operations	Logical operations
20.3.5	lib.negators	Negators
20.3.6	lib.binders	Binders
20.3.6.1	lib.binder.1st	Template class binder1st
20.3.6.2	lib.bind.1st	bind1st
20.3.6.3	lib.binder.2nd	Template class binder2nd
20.3.6.4	lib.bind.2nd	bind2nd
20.3.7	lib.function.pointer.adaptors	Adaptors for pointers to functions
20.4	lib.memory	Memory
20.4.1	lib.default allocator	The default allocator
20.4.1.1	lib.allocator.members	allocator members
20.4.1.2	lib.allocator.placement	allocator placement new
20.4.1.3	lib.allocator.example	Example allocator
20.4.2	lib.storage.iterator	Raw storage iterator
20.4.3	lib.memory.primitives	Memory handling primitives
20.4.3.1	lib.allocate	allocate
20.4.3.2	lib.deallocate	deallocate
20.4.3.3	lib.construct	construct
20.4.3.4	lib.destroy	destroy
20.4.3.5	lib.temporary.buffer	Temporary buffers
20.4.4	lib.specialized.algorithms	Specialized algorithms
20.4.4.1	lib.uninitialized.copy	uninitialized_copy
20.4.4.2	lib.uninitialized.fill	uninitialized_fill
20.4.4.3	lib.uninitialized.fill.n	uninitialized_fill
20.4.5	lib.auto_ptr	Template class auto_ptr
20.4.5.1	lib.auto_ptr.cons	auto_ptr constructors
20.4.5.2	lib.auto_ptr.members	auto_ptr members
20.4.6	lib.c.malloc	C Library
20.5	lib.date.time	Date and time
21	lib.strings	Strings library
21.1	lib.string.classes	String classes
21.1.1	lib.template.string	Template class basic_string
21.1.1.1	lib.string.char.traits	Template class string_char_traits
21.1.1.2	lib.string.char.traits.members	string_char_traits members
21.1.1.3	lib.basic.string	Template class basic_string
21.1.1.4	lib.string.cons	basic_string constructors
21.1.1.5	lib.string.iterators	basic_string iterator support
21.1.1.6	lib.string.capacity	basic_string capacity
21.1.1.7	lib.string.access	basic_string element access
21.1.1.8	lib.string.modifiers	basic_string modifiers
21.1.1.8.1	lib.string::op+=	basic_string::operator+=
21.1.1.8.2	lib.string::append	basic_string::append
21.1.1.8.3	lib.string::assign	basic_string::assign
21.1.1.8.4	lib.string::insert	basic_string::insert
21.1.1.8.5	lib.string::remove	basic_string::remove
21.1.1.8.6	lib.string::replace	basic_string::replace
21.1.1.8.7	lib.string::copy	basic_string::copy
21.1.1.8.8	lib.string::swap	basic_string::swap
21.1.1.9	lib.string.ops	basic_string string operations
21.1.1.9.1	lib.string::find	basic_string::find
21.1.1.9.2	lib.string::rfind	basic_string::rfind

21.1.1.9.3	lib.string::find.first.of	basic_string::find_first_of
21.1.1.9.4	lib.string::find.last.of	basic_string::find_last_of
21.1.1.9.5	lib.string::find.first.not.of	basic_string::find_first_not_of
21.1.1.9.6	lib.string::find.last.not.of	basic_string::find_last_not_of
21.1.1.9.7	lib.string::substr	basic_string::substr
21.1.1.9.8	lib.string::compare	basic_string::compare
21.1.1.10	lib.string.nonmembers	basic_string non-member functions
21.1.1.10.1	lib.string::op+	operator+
21.1.1.10.2	lib.string::operator==	operator==
21.1.1.10.3	lib.string::op!=	operator!=
21.1.1.10.4	lib.string::op<	operator<
21.1.1.10.5	lib.string::op>	operator>
21.1.1.10.6	lib.string::op<=	operator<=
21.1.1.10.7	lib.string::op>=	operator>=
21.1.1.10.8		Inserters and extractors
21.1.2	lib.string	Class string
21.1.3	lib.string.traits.members	string_char_traits<char> members
21.1.4	lib.wstring	Class wstring
21.1.5	lib.wstring.members	string_char_traits<wchar_t> members
21.2	lib.c.strings	Null-terminated sequence utilities
22	lib.localization	Localization library
22.1	lib.locales	Locales
22.1.1	lib.locale	Class locale
22.1.1.1	lib.locale.types	locale types
22.1.1.1.1	lib.locale.category	Type locale::category
22.1.1.1.2	lib.locale.facet	Class locale::facet
22.1.1.1.3	lib.locale.id	Class locale::id
22.1.1.2	lib.locale.cons	locale constructors and destructor
22.1.1.3	lib.locale.members	locale members
22.1.1.4	lib.locale.operators	locale operators
22.1.1.5	lib.locale.statics	locale static members
22.1.2	lib.locale.convenience	Convenience interfaces
22.1.2.1	lib.classification	Character classification
22.1.2.2	lib.conversions	Character conversions
22.2	lib.locale.categories	Standard locale categories
22.2.1	lib.category ctype	The ctype category
22.2.1.1	lib.locale.ctype	Template class ctype
22.2.1.1.1	lib.locale.ctype.members	ctype members
22.2.1.1.2	lib.locale.ctype.virtuals	ctype virtual functions
22.2.1.2	lib.locale.ctype.byname	Template class ctype_byname
22.2.1.3	lib.facet.ctype.special	ctype specializations
22.2.1.3.1	lib.facet.ctype.char.dtor	ctype<char> destructor
22.2.1.3.2	lib.facet.ctype.char.members	ctype<char> members
22.2.1.3.3	lib.facet.ctype.char.virtuals	ctype<char> overridden virtual functions
22.2.1.4	lib.locale.codecvt	Template class codecvt
22.2.1.4.1	lib.locale.codecvt.members	codecvt members
22.2.1.4.2	lib.locale.codecvt.virtuals	codecvt virtual functions
22.2.1.5	lib.locale.codecvt.byname	Template class codecvt_byname
22.2.2	lib.category.numeric	The numeric category
22.2.2.1	lib.locale.num.get	Template class num_get
22.2.2.1.1	lib.facet.num.get.members	num_get members
22.2.2.1.2	lib.facet.num.get.virtuals	num_get virtual functions
22.2.2.2	lib.locale.num.put	Template class num_put
22.2.2.2.1	lib.facet.num.put.members	num_put members
22.2.2.2.2	lib.facet.num.put.virtuals	num_put virtual functions

22.2.3	lib.facet.numpunct	The numeric punctuation facet
22.2.3.1	lib.locale.numpunct	Template class numpunct
22.2.3.1.1	lib.facet.numpunct.members	numpunct members
22.2.3.1.2	lib.facet.numpunct.virtuals	numpunct virtual functions
22.2.3.2	lib.locale.numpunct.byname	Template class numpunct_byname
22.2.4	lib.category.collate	The collate category
22.2.4.1	lib.locale.collate	Template class collate
22.2.4.1.1	lib.locale.collate.members	collate members
22.2.4.1.2	lib.locale.collate.virtuals	collate virtual functions
22.2.4.2	lib.locale.collate.byname	Template class collate_byname
22.2.5	lib.category.time	The time category
22.2.5.1	lib.locale.time.get	Template class time_get
22.2.5.1.1	lib.locale.time.get.members	time_get members
22.2.5.1.2	lib.locale.time.get.virtuals	time_get virtual functions
22.2.5.2	lib.locale.time.get.byname	Template class time_get_byname
22.2.5.3	lib.locale.time.put	Template class time_put
22.2.5.3.1	lib.locale.time.put.members	time_put members
22.2.5.3.2	lib.locale.time.put.virtuals	time_put virtual functions
22.2.5.4	lib.locale.time.put.byname	Template class time_put_byname
22.2.6	lib.category.monetary	The monetary category
22.2.6.1	lib.locale.money.get	Template class money_get
22.2.6.1.1	lib.locale.money.get.members	money_get members
22.2.6.1.2	lib.locale.money.get.virtuals	money_get virtual functions
22.2.6.2	lib.locale.money.put	Template class money_put
22.2.6.2.1	lib.locale.money.put.members	money_put members
22.2.6.2.2	lib.locale.money.put.virtuals	money_put virtual functions
22.2.6.3	lib.locale.money.punct	Template class money_punct
22.2.6.3.1	lib.locale.money.punct.members	money_punct members
22.2.6.3.2	lib.locale.money.punct.virtuals	money_punct virtual functions
22.2.6.4	lib.locale.money.punct.byname	Template class money_punct_byname
22.2.7	lib.category.messages	The message retrieval category
22.2.7.1	lib.locale.messages	Template class messages
22.2.7.1.1	lib.locale.messages.members	messages members
22.2.7.1.2	lib.locale.messages.virtuals	messages virtual functions
22.2.7.2	lib.locale.messages.byname	Template class messages_byname
22.2.8	lib.facets.examples	Program-defined facets
22.3	lib.c.locales	C Library Locales
23	lib.containers	Containers library
23.1	lib.container.requirements	Container requirements
23.1.1	lib.sequence.reqmts	Sequences
23.1.2	lib.associative.reqmts	Associative containers
23.2	lib.sequences	Sequences
23.2.1	lib.template.bitset	Template class bitset
23.2.1.1	lib.bitset.cons	bitset constructors
23.2.1.2	lib.bitset.members	bitset members
23.2.1.3	lib.bitset.operators	bitset operators
23.2.2	lib.deque	Template class deque
23.2.2.1	lib.deque.types	deque types
23.2.2.2	lib.deque.cons	deque constructors, copy, and assignment
23.2.2.3	lib.deque.iterators	deque iterator support

23.2.2.4	lib.deque.capacity	deque capacity
23.2.2.5	lib.deque.access	deque element access
23.2.2.6	lib.deque.modifiers	deque modifiers
23.2.3	lib.list	Template class <code>list</code>
23.2.3.1	lib.list.types	<code>list</code> types
23.2.3.2	lib.list.cons	<code>list</code> constructors, copy, and assignment
23.2.3.3	lib.list.iterators	<code>list</code> iterator support
23.2.3.4	lib.list.capacity	<code>list</code> capacity
23.2.3.5	lib.list.access	<code>list</code> element access
23.2.3.6	lib.list.modifiers	<code>list</code> modifiers
23.2.3.7	lib.list.ops	<code>list</code> operations
23.2.4	lib.container.adapters	Container adapters
23.2.4.1	lib.queue	Template class <code>queue</code>
23.2.4.2	lib.priority.queue	Template class <code>priority_queue</code>
23.2.4.2.1	lib.priqueue.cons	<code>priority_queue</code> constructors
23.2.4.2.2	lib.priqueue.members	<code>priority_queue</code> members
23.2.4.3	lib.stack	Template class <code>stack</code>
23.2.5	lib.vector	Template class <code>vector</code>
23.2.5.1	lib.vector.types	<code>vector</code> types
23.2.5.2	lib.vector.cons	<code>vector</code> constructors, copy, and assignment
23.2.5.3	lib.vector.iterators	<code>vector</code> iterator support
23.2.5.4	lib.vector.capacity	<code>vector</code> capacity
23.2.5.5	lib.vector.access	<code>vector</code> element access
23.2.5.6	lib.vector.modifiers	<code>vector</code> modifiers
23.2.6	lib.vector.bool	Class <code>vector<bool></code>
23.3	lib.associative	Associative containers
23.3.1	lib.map	Template class <code>map</code>
23.3.1.1	lib.map.types	<code>map</code> types
23.3.1.2	lib.map.cons	<code>map</code> constructors, copy, and assignment
23.3.1.3	lib.map.iterators	<code>map</code> iterator support
23.3.1.4	lib.map.capacity	<code>map</code> capacity
23.3.1.5	lib.map.access	<code>map</code> element access
23.3.1.6	lib.map.modifiers	<code>map</code> modifiers
23.3.1.7	lib.map.observers	<code>map</code> observers
23.3.1.8	lib.map.ops	<code>map</code> operations
23.3.2	lib.multimap	Template class <code>multimap</code>
23.3.3	lib.set	Template class <code>set</code>
23.3.3.1	lib.set.types	<code>set</code> types
23.3.3.2	lib.set.cons	<code>set</code> constructors, copy, and assignment
23.3.3.3	lib.set.iterators	<code>set</code> iterator support
23.3.3.4	lib.set.capacity	<code>set</code> capacity
23.3.3.5	lib.set.modifiers	<code>set</code> modifiers
23.3.3.6	lib.set.observers	<code>set</code> observers
23.3.3.7	lib.set.ops	<code>set</code> operations
23.3.4	lib.multiset	Template class <code>multiset</code>
24	lib.iterators	Iterators library
24.1	lib.iterator.requirements	Iterator requirements
24.1.1	lib.input.iterators	Input iterators
24.1.2	lib.output.iterators	Output iterators
24.1.3	lib.forward.iterators	Forward iterators
24.1.4	lib.bidirectional.iterators	Bidirectional iterators
24.1.5	lib.random.access.iterators	Random access iterators
24.1.6	lib.iterator.tags	Iterator tags
24.2	lib.iterator.primitives	Iterator primitives
24.2.1	lib.std.iterator.tags	Standard iterator tags

24.2.2	lib.basic.iterators	Basic iterators	
24.2.3	lib.iterator.category	iterator_category	
24.2.4	lib.value.type	value_type	
24.2.5	lib.distance.type	distance_type	
24.2.6	lib.iterator.operations	Iterator operations	
24.3	lib.predef.iterators	Predefined iterators	
24.3.1	lib.reverse.iterators	Reverse iterators	
24.3.1.1	lib.reverse.bidir.iter	Template	class
		reverse_bidirectional_iterator	
24.3.1.2	lib.reverse.bidir.iter.ops	reverse_bidirectional_iterator operations	
24.3.1.2.1	lib.reverse.bidir.iter.cons	reverse_bidirectional_iterator constructor	
24.3.1.2.2	lib.reverse.bidir.iter.conv	Conversion	
24.3.1.2.3	lib.reverse.bidir.iter.op.star	operator*	
24.3.1.2.4	lib.reverse.bidir.iter.op.++	operator++	
24.3.1.2.5	lib.reverse.bidir.iter.op.--	operator--	
24.3.1.2.6	lib.reverse.bidir.iter.op.==	operator==	
24.3.1.3	lib.reverse.iterator	Template class reverse_iterator	
24.3.1.4	lib.reverse.iter.ops	reverse_iterator operations	
24.3.1.4.1	lib.reverse.iter.cons	reverse_iterator constructor	
24.3.1.4.2	lib.reverse.iter.conv	Conversion	
24.3.1.4.3	lib.reverse.iter.op.star	operator*	
24.3.1.4.4	lib.reverse.iter.op.++	operator++	
24.3.1.4.5	lib.reverse.iter.op.--	operator--	
24.3.1.4.6	lib.reverse.iter.op.==	operator==	
24.3.2	lib.insert.iterators	Insert iterators	
24.3.2.1	lib.back.insert.iterator	Template class back_insert_iterator	
24.3.2.2	lib.back.insert.iter.ops	back_insert_iterator operations	
24.3.2.2.1	lib.back.insert.iter.cons	back_insert_iterator constructor	
24.3.2.2.2	lib.back.insert.iter.op.=	back_insert_iterator::operator=	
24.3.2.2.3	lib.back.insert.iter.op.*	back_insert_iterator::operator*	
24.3.2.2.4	lib.back.insert.iter.op.++	back_insert_iterator::operator++	
24.3.2.2.5	lib.back.inserter	back_inserter	
24.3.2.3	lib.front.insert.iterator	Template class front_insert_iterator	
24.3.2.4	lib.front.insert.iter.ops	front_insert_iterator operations	
24.3.2.4.1	lib.front.insert.iter.cons	front_insert_iterator constructor	
24.3.2.4.2	lib.front.insert.iter.op.=	front_insert_iterator::operator=	
24.3.2.4.3	lib.front.insert.iter.op.*	front_insert_iterator::operator*	
24.3.2.4.4	lib.front.insert.iter.op.++	front_insert_iterator::operator++	
24.3.2.4.5	lib.front.inserter	front_inserter	
24.3.2.5	lib.insert.iterator	Template class insert_iterator	
24.3.2.6	lib.insert.iter.ops	insert_iterator operations	
24.3.2.6.1	lib.insert.iter.cons	insert_iterator constructor	
24.3.2.6.2	lib.insert.iter.op.=	insert_iterator::operator=	
24.3.2.6.3	lib.insert.iter.op.*	insert_iterator::operator*	
24.3.2.6.4	lib.insert.iter.op.++	insert_iterator::operator++	
24.3.2.6.5	lib.inserter	inserter	
24.4	lib.stream.iterators	Stream iterators	
24.4.1	lib.istream.iterator	Template class istream_iterator	
24.4.2	lib ostream.iterator	Template class ostream_iterator	
24.4.3	lib.istreambuf.iterator	Template class istreambuf_iterator	
24.4.3.1	lib.istreambuf.iterator::proxy	Template	class
		istreambuf_iterator::proxy	
24.4.3.2	lib.istreambuf.iterator.cons	istreambuf_iterator constructors	

24.4.3.3	lib.istreambuf.iterator::op*	istreambuf_iterator::operator*
24.4.3.4	lib.istreambuf.iterator::op++	istreambuf_iterator::operator++
24.4.3.5	lib.istreambuf.iterator::equal	istreambuf_iterator::equal
24.4.3.6	lib.iterator.category.i	iterator_category
24.4.3.7	lib.istreambuf.iterator::op==	operator==
24.4.3.8	lib.istreambuf.iterator::op!=	operator!=
24.4.4	lib ostreambuf.iterator	Template class ostreambuf_iterator
24.4.4.1	lib ostreambuf.iter.cons	ostreambuf_iterator constructors
24.4.4.2	lib ostreambuf.iter.ops	ostreambuf_iterator operations
24.4.4.3	lib ostreambuf.iterator.nonmembers	ostreambuf_iterator non-member operations
25	lib.algorithms	Algorithms library
25.1	lib.alg.nonmodifying	Non-modifying sequence operations
25.1.1	lib.alg.foreach	For each
25.1.2	lib.alg.find	Find
25.1.3	lib.alg.find.end	Find End
25.1.4	lib.alg.find.first.of	Find First
25.1.5	lib.alg.adjacent.find	Adjacent find
25.1.6	lib.alg.count	Count
25.1.7	lib.mismatch	Mismatch
25.1.8	lib.alg.equal	Equal
25.1.9	lib.alg.search	Search
25.2	lib.alg.modifying.operations	Mutating sequence operations
25.2.1	lib.alg.copy	Copy
25.2.2	lib.alg.swap	Swap
25.2.3	lib.alg.transform	Transform
25.2.4	lib.alg.replace	Replace
25.2.5	lib.alg.fill	Fill
25.2.6	lib.alg.generate	Generate
25.2.7	lib.alg.remove	Remove
25.2.8	lib.alg.unique	Unique
25.2.9	lib.alg.reverse	Reverse
25.2.10	lib.alg.rotate	Rotate
25.2.11	lib.alg.random.shuffle	Random shuffle
25.2.12	lib.alg.partitions	Partitions
25.3	lib.alg.sorting	Sorting and related operations
25.3.1	lib.alg.sort	Sorting
25.3.1.1	lib.sort	sort
25.3.1.2	lib.stable.sort	stable_sort
25.3.1.3	lib.partial.sort	partial_sort
25.3.1.4	lib.partial.sort.copy	partial_sort_copy
25.3.2	lib.alg.nth.element	Nth element
25.3.3	lib.alg.binary.search	Binary search
25.3.3.1	lib.lower.bound	lower_bound
25.3.3.2	lib.upper.bound	upper_bound
25.3.3.3	lib.equal.range	equal_range
25.3.3.4	lib.binary.search	binary_search
25.3.4	lib.alg.merge	Merge
25.3.5	lib.alg.set.operations	Set operations on sorted structures
25.3.5.1	lib.includes	includes
25.3.5.2	lib.set.union	set_union
25.3.5.3	lib.set.intersection	set_intersection
25.3.5.4	lib.set.difference	set_difference
25.3.5.5	lib.set.symmetric.difference	set_symmetric_difference

25.3.6	lib.alg.heap.operations	Heap operations
25.3.6.1	lib.push.heap	push_heap
25.3.6.2	lib.pop.heap	pop_heap
25.3.6.3	lib.make.heap	make_heap
25.3.6.4	lib.sort.heap	sort_heap
25.3.7	lib.alg.min.max	Minimum and maximum
25.3.8	lib.alg.lex.comparison	Lexicographical comparison
25.3.9	lib.alg.permutation.generators	Permutation generators
25.4	lib.alg.c.library	C library algorithms
26	lib.numerics	Numerics library
26.1	lib.numeric.requirements	Numeric type requirements
26.2	lib.complex.numbers	Complex numbers
26.2.1	lib.complex	Template class <code>complex</code>
26.2.2	lib.complex.special	<code>complex</code> specializations
26.2.3	lib.complex.members	<code>complex</code> member functions
26.2.4	lib.complex.member.ops	<code>complex</code> member operators
26.2.5	lib.complex.ops	<code>complex</code> non-member operations
26.2.6	lib.complex.value.ops	<code>complex</code> value operations
26.2.7	lib.complex.transcendentals	<code>complex</code> transcendentals
26.3	lib.numarray	Numeric arrays
26.3.1	lib.template.valarray	Template class <code>valarray</code>
26.3.1.1	lib.valarray.cons	<code>valarray</code> constructors
26.3.1.2	lib.valarray.assign	<code>valarray</code> assignment
26.3.1.3	lib.valarray.access	<code>valarray</code> element access
26.3.1.4	lib.valarray.sub	<code>valarray</code> subset operations
26.3.1.5	lib.valarray.unary	<code>valarray</code> unary operators
26.3.1.6	lib.valarray.cassign	<code>valarray</code> computed assignment
26.3.1.7	lib.valarray.members	<code>valarray</code> member functions
26.3.2	lib.valarray.nonmembers	<code>valarray</code> non-member operations
26.3.2.1	lib.valarray.binary	<code>valarray</code> binary operators
26.3.2.2	lib.valarray.comparison	<code>valarray</code> comparison operators
26.3.2.3	lib.valarray.min.max	<code>valarray</code> min and max functions
26.3.2.4	lib.valarray.transcend	<code>valarray</code> transcendentals
26.3.3	lib.class.slice	Class <code>slice</code>
26.3.3.1	lib.cons.slice	<code>slice</code> constructors
26.3.3.2	lib.slice.access	<code>slice</code> access functions
26.3.4	lib.template.slice.array	Template class <code>slice_array</code>
26.3.4.1	lib.cons.slice.arr	<code>slice_array</code> constructors
26.3.4.2	lib.slice.arr.assign	<code>slice_array</code> assignment
26.3.4.3	lib.slice.arr.comp.assign	<code>slice_array</code> computed assignment
26.3.4.4	lib.slice.arr.fill	<code>slice_array</code> fill function
26.3.5	lib.class.gslic	The <code>gslic</code> class
26.3.5.1	lib.gslic.cons	<code>gslic</code> constructors
26.3.5.2	lib.gslic.access	<code>gslic</code> access functions
26.3.6	lib.template.gslic.array	Template class <code>gslic_array</code>
26.3.6.1	lib.gslic.array.cons	<code>gslic_array</code> constructors
26.3.6.2	lib.gslic.array.assign	<code>gslic_array</code> assignment
26.3.6.3	lib.gslic.array.comp.assign	<code>gslic_array</code> computed assignment
26.3.6.4	lib.gslic.array.fill	<code>gslic_array</code> fill function
26.3.7	lib.template.mask.array	Template class <code>mask_array</code>
26.3.7.1	lib.mask.array.cons	<code>mask_array</code> constructors
26.3.7.2	lib.mask.array.assign	<code>mask_array</code> assignment
26.3.7.3	lib.mask.array.comp.assign	<code>mask_array</code> computed assignment
26.3.7.4	lib.mask.array.fill	<code>mask_array</code> fill function

26.3.8	lib.template.indirect.array	Template class <code>indirect_array</code>
26.3.8.1	lib.indirect.array.cons	<code>indirect_array</code> constructors
26.3.8.2	lib.indirect.array.assign	<code>indirect_array</code> assignment
26.3.8.3	lib.indirect.array.comp.assign	<code>indirect_array</code> computed assignment
26.3.8.4	lib.indirect.array.fill	<code>indirect_array</code> fill function
26.4	lib.numeric.ops	Generalized numeric operations
26.4.1	lib.accumulate	Accumulate
26.4.2	lib.inner.product	Inner product
26.4.3	lib.partial.sum	Partial sum
26.4.4	lib.adjacent.difference	Adjacent difference
26.5	lib.c.math	C Library
27	lib.input.output	Input/output library
27.1	lib.iostreams.requirements	Iostreams requirements
27.1.1	lib.iostreams.definitions	Definitions
27.1.2	lib.iostreams.type.reqmts	Type requirements
27.1.2.1	lib.iostreams.char.t	Type <code>CHAR_T</code>
27.1.2.2	lib.iostreams.int.t	Type <code>INT_T</code>
27.1.2.3	lib.iostreams.off.t	Type <code>OFF_T</code>
27.1.2.4	lib.iostreams.pos.t	Type <code>POS_T</code>
27.2	lib.iostream.forward	Forward declarations
27.3	lib.iostream.objects	Standard iostream objects
27.3.1	lib.narrow.stream.objects	Narrow stream objects
27.3.2	lib.wide.stream.objects	Wide stream objects
27.4	lib.iostreams.base	Iostreams base classes
27.4.1	lib.stream.types	Types
27.4.2	lib.ios.traits	Template struct <code>ios_traits</code>
27.4.2.1	lib.ios.traits.types	<code>ios_traits</code> types
27.4.2.2	lib.ios.traits.values	<code>ios_traits</code> value functions
27.4.2.3	lib.ios.traits.tests	<code>ios_traits</code> test functions
27.4.2.4	lib.ios.traits.convert	<code>ios_traits</code> conversion functions
27.4.3	lib.ios.base	Class <code>ios_base</code>
27.4.3.1	lib.ios.types	Types
27.4.3.1.1	lib.ios::failure	Class <code>ios_base::failure</code>
27.4.3.1.2	lib.ios::fmtflags	Type <code>ios_base::fmtflags</code>
27.4.3.1.3	lib.ios::iostate	Type <code>ios_base::iostate</code>
27.4.3.1.4	lib.ios::openmode	Type <code>ios_base::openmode</code>
27.4.3.1.5	lib.ios::seekdir	Type <code>ios_base::seekdir</code>
27.4.3.1.6	lib.ios::Init	Class <code>ios_base::Init</code>
27.4.3.2	lib.fmtflags.state	<code>ios_base</code> <code>fmtflags</code> state functions
27.4.3.3	lib.ios.base.locales	<code>ios_base</code> locale functions
27.4.3.4	lib.ios.base.storage	<code>ios_base</code> storage functions
27.4.3.5	lib.ios.base.cons	<code>ios_base</code> constructors
27.4.4	lib.ios	Template class <code>basic_ios</code>
27.4.4.1	lib.basic.ios.cons	<code>basic_ios</code> constructors
27.4.4.2	lib.basic.ios.members	Member functions
27.4.4.3	lib.iostate.flags	<code>basic_ios</code> <code>iostate</code> flags functions
27.4.5	lib.std.ios.manip	<code>ios_base</code> manipulators
27.4.5.1	lib.fmtflags.manip	<code>fmtflags</code> manipulators
27.4.5.2	lib.adjustfield.manip	<code>adjustfield</code> manipulators
27.4.5.3	lib.basefield.manip	<code>basefield</code> manipulators
27.4.5.4	lib.floatfield.manip	<code>floatfield</code> manipulators
27.5	lib.stream.buffer	Stream buffers
27.5.1	lib.streambuf.reqts	Stream buffer requirements
27.5.2	lib.streambuf	Template class

		<code>basic_streambuf<charT, traits></code>
27.5.2.1	<code>lib.streambuf.cons</code>	<code>basic_streambuf</code> constructors
27.5.2.2	<code>lib.streambuf.members</code>	<code>basic_streambuf</code> public member functions
27.5.2.2.1	<code>lib.streambuf.locales</code>	Locales
27.5.2.2.2	<code>lib.streambuf.buffer</code>	Buffer management and positioning
27.5.2.2.3	<code>lib.streambuf.pub.get</code>	Get area
27.5.2.2.4	<code>lib.streambuf.pub.pback</code>	Putback
27.5.2.2.5	<code>lib.streambuf.pub.put</code>	Put area
27.5.2.3	<code>lib.streambuf.protected</code>	<code>basic_streambuf</code> protected member functions
27.5.2.3.1	<code>lib.streambuf.get.area</code>	Get area access
27.5.2.3.2	<code>lib.streambuf.put.area</code>	Put area access
27.5.2.4	<code>lib.streambuf.virtuals</code>	<code>basic_streambuf</code> virtual functions
27.5.2.4.1	<code>lib.streambuf.virt.locales</code>	Locales
27.5.2.4.2	<code>lib.streambuf.virt.buffer</code>	Buffer management and positioning
27.5.2.4.3	<code>lib.streambuf.virt.get</code>	Get area
27.5.2.4.4	<code>lib.streambuf.virt.pback</code>	Putback
27.5.2.4.5	<code>lib.streambuf.virt.put</code>	Put area
27.6	<code>lib.iostream.format</code>	Formatting and manipulators
27.6.1	<code>lib.input.streams</code>	Input streams
27.6.1.1	<code>lib.istream</code>	Template class <code>basic_istream</code>
27.6.1.1.1	<code>lib.basic.istream.cons</code>	<code>basic_istream</code> constructors
27.6.1.1.2	<code>lib.istream.prefix</code>	<code>basic_istream</code> prefix and suffix
27.6.1.2	<code>lib.istream.formatted</code>	Formatted input functions
27.6.1.2.1	<code>lib.istream.formatted.reqmts</code>	Common requirements
27.6.1.2.2	<code>lib.istream::extractors</code>	<code>basic_istream::operator>></code>
27.6.1.3	<code>lib.istream.unformatted</code>	Unformatted input functions
27.6.1.4	<code>lib.istream.manip</code>	Standard <code>basic_istream</code> manipulators
27.6.2	<code>lib.output.streams</code>	Output streams
27.6.2.1	<code>lib ostream</code>	Template class <code>basic_ostream</code>
27.6.2.2	<code>lib ostream.cons</code>	<code>basic_ostream</code> constructors
27.6.2.3	<code>lib ostream.prefix</code>	<code>basic_ostream</code> prefix and suffix functions
27.6.2.4	<code>lib ostream.formatted</code>	Formatted output functions
27.6.2.4.1	<code>lib ostream.formatted.reqmts</code>	Common requirements
27.6.2.4.2	<code>lib ostream.inserters</code>	<code>basic_ostream::operator<<</code>
27.6.2.5	<code>lib ostream.unformatted</code>	Unformatted output functions
27.6.2.6	<code>lib ostream.manip</code>	Standard <code>basic_ostream</code> manipulators
27.6.3	<code>lib.std.manip</code>	Standard manipulators
27.7	<code>lib.string.streams</code>	String-based streams
27.7.1	<code>lib.stringbuf</code>	Template class <code>basic_stringbuf</code>
27.7.1.1	<code>lib.stringbuf.cons</code>	<code>basic_stringbuf</code> constructors
27.7.1.2	<code>lib.stringbuf.members</code>	Member functions
27.7.1.3	<code>lib.stringbuf.virtuals</code>	Overridden virtual functions
27.7.2	<code>lib.istringstream</code>	Template class <code>basic_istringstream</code>
27.7.2.1	<code>lib.istringstream.cons</code>	<code>basic_istringstream</code> constructors
27.7.2.2	<code>lib.istringstream.members</code>	Member functions
27.7.2.3	<code>lib.ostringstream</code>	Class <code>basic_ostringstream</code>
27.7.2.4	<code>lib.ostringstream.cons</code>	<code>basic_ostringstream</code> constructors
27.7.2.5	<code>lib.ostringstream.members</code>	Member functions
27.8	<code>lib.file.streams</code>	File-based streams
27.8.1	<code>lib.fstreams</code>	File streams
27.8.1.1	<code>lib.filebuf</code>	Template class <code>basic_filebuf</code>
27.8.1.2	<code>lib.filebuf.cons</code>	<code>basic_filebuf</code> constructors
27.8.1.3	<code>lib.filebuf.members</code>	Member functions
27.8.1.4	<code>lib.filebuf.virtuals</code>	Overridden virtual functions

27.8.1.5	lib ifstream	Template class <code>basic_ifstream</code>
27.8.1.6	lib ifstream.cons	<code>basic_ifstream</code> constructors
27.8.1.7	lib ifstream.members	Member functions
27.8.1.8	lib ofstream	Template class <code>basic_ofstream</code>
27.8.1.9	lib ofstream.cons	<code>basic_ofstream</code> constructors
27.8.1.10	lib ofstream.members	Member functions
27.8.2	lib.c.files	C Library files
A	gram	Grammar summary
A.1	gram.key	Keywords
A.2	gram.lex	Lexical conventions
A.3	gram.basic	Basic concepts
A.4	gram.expr	Expressions
A.5	gram.stmt.stmt	Statements
A.6	gram.dcl.dcl	Declarations
A.7	gram.dcl.decl	Declarators
A.8	gram.class	Classes
A.9	gram.class.derived	Derived classes
A.10	gram.special	Special member functions
A.11	gram.over	Overloading
A.12	gram.temp	Templates
A.13	gram.except	Exception handling
B	limits	Implementation quantities
C	diff	Compatibility
C.1	diff.c	Extensions
C.1.1	diff.early	C++ features available in 1985
C.1.2	diff.c++	C++ features added since 1985
C.2	diff.iso	C++ and ISO C
C.2.1	diff.lex	Clause <code>_lex_</code> : lexical conventions
C.2.2	diff.basic	Clause <code>_basic_</code> : basic concepts
C.2.3	diff.expr	Clause <code>_expr_</code> : expressions
C.2.4	diff.stat	Clause <code>_stmt.stmt_</code> : statements
C.2.5	diff.dcl	Clause <code>_dcl.dcl_</code> : declarations
C.2.6	diff.decl	Clause <code>_dcl.decl_</code> : declarators
C.2.7	diff.class	Clause <code>_class_</code> : classes
C.2.8	diff.special	Clause <code>_special_</code> : special member functions
C.2.9	diff.cpp	Clause <code>_cpp_</code> : preprocessing directives
C.3	diff.anac	Anachronisms
C.3.1	diff.fct.def	Old style function definitions
C.3.2	diff.base.init	Old style base class initializer
C.3.3	diff.this	Assignment to <code>this</code>
C.3.4	diff.bound	Cast of bound pointer
C.3.5	diff.class.nonnested	Nonnested classes
C.4	diff.library	Standard C library
C.4.1	diff.mods.to.headers	Modifications to headers
C.4.2	diff.mods.to.definitions	Modifications to definitions
C.4.2.1	diff.wchar.t	Type <code>wchar_t</code>
C.4.2.2	diff.header.iso646.h	Header <code><iso646.h></code>
C.4.2.3	diff.null	Macro <code>NULL</code>
C.4.3	diff.mods.to.declarations	Modifications to declarations
C.4.4	diff.mods.to.behavior	Modifications to behavior
C.4.4.1	diff.offsetof	Macro <code>offsetof(type, member-</code>
C.4.4.2	diff.malloc	<code>designator)</code> .CW <code><stddef.h></code> Memory allocation functions
D	depr	Compatibility features
D.1	depr.c.headers	Standard C library headers

D.2	depr.ios.members	Old iostreams members
D.3	depr.str.strstreams	char* streams
D.3.1	depr.strstreambuf	Class <code>strstreambuf</code>
D.3.1.1	depr.strstreambuf.cons	<code>strstreambuf</code> constructors
D.3.1.2	depr.strstreambuf.members	Member functions
D.3.1.3	depr.strstreambuf.virtuals	<code>strstreambuf</code> overridden virtual functions
D.3.2	depr.istream	Template class <code>istream</code>
D.3.2.1	depr.istream.cons	<code>istream</code> constructors
D.3.2.2	depr.istream.members	Member functions
D.3.3	depr.ostringstream	Template class <code>ostringstream</code>
D.3.3.1	depr.ostringstream.cons	<code>ostringstream</code> constructors
D.3.3.2	depr.ostringstream.members	Member functions

Listing by symbolic name

basic	3	Basic concepts
basic.compound	3.9.2	Compound types
basic.def	3.1	Declarations and definitions
basic.def.odr	3.2	One definition rule
basic.fundamental	3.9.1	Fundamental types
basic.life	3.8	Object Lifetime
basic.link	3.5	Program and linkage
basic.lval	3.10	Lvalues and rvalues
basic.namespace	7.3	Namespaces
basic.scope	3.3	Declarative regions and scopes
basic.scope.class	3.3.5	Class scope
basic.scope.elab	3.3.8	Elaborated type specifier
basic.scope.exqual	3.3.7	Explicit qualification
basic.scope.hiding	3.3.6	Name hiding
basic.scope.local	3.3.1	Local scope
basic.scope.namespace	3.3.4	Namespace scope
basic.scope.pdecl	3.3.9	Point of declaration
basic.scope.proto	3.3.2	Function prototype scope
basic.start	3.6	Start and termination
basic.start.init	3.6.2	Initialization of non-local objects
basic.start.main	3.6.1	Main function
basic.start.term	3.6.3	Termination
basic.stc	3.7	Storage duration
basic.stc.auto	3.7.2	Automatic storage duration
basic.stc.dynamic	3.7.3	Dynamic storage duration
basic.stc.dynamic.allocation	3.7.3.1	Allocation functions
basic.stc.dynamic.deallocation	3.7.3.2	Deallocation functions
basic.stc.inherit	3.7.4	Duration of sub-objects
basic.stc.static	3.7.1	Static storage duration
basic.type.name	3.9.4	Type names
basic.type.qualifier	3.9.3	CV-qualifiers
basic.types	3.9	Types
class	9	Classes
class.abstract	10.4	Abstract classes
class.access	11	Member access control
class.access.base	11.2	Access specifiers for base classes
class.access.dcl	11.3	Access declarations
class.access.spec	11.1	Access specifiers
class.access.virt	11.6	Access to virtual functions
class.base.init	12.6.2	Initializing bases and members
class.bit	9.7	Bit-fields
class.ctor	12.7	Construction and destruction
class.conv	12.3	Conversions
class.conv.ctor	12.3.1	Conversion by constructor
class.conv.fct	12.3.2	Conversion functions
class.copy	12.8	Copying class objects
class.ctor	12.1	Constructors
class.derived	10	Derived classes
class.dtor	12.4	Destructors
class.expl.init	12.6.1	Explicit initialization
class.free	12.5	Free store
class.friend	11.4	Friends

class.init	12.6	Initialization
class.local	9.9	Local class declarations
class.mem	9.2	Class members
class.member.lookup	10.2	Member name lookup
class.mfct	9.4	Member functions
class.mfct.nonstatic	9.4.1	Nonstatic member functions
class.mi	10.1	Multiple base classes
class.name	9.1	Class names
class.nest	9.8	Nested class declarations
class.nested.type	9.10	Nested type names
class.paths	11.7	Multiple access
class.protected	11.5	Protected member access
class.scope	3.4	Name look up
class.scope0	9.3	Scope rules for classes
class.static	9.5	Static members
class.static.data	9.5.2	Static data members
class.static.mfct	9.5.1	Static member functions
class.temporary	12.2	Temporary objects
class.this	9.4.2	The <code>this</code> pointer
class.union	9.6	Unions
class.virtual	10.3	Virtual functions
conv	4	Standard conversions
conv.array	4.2	Array-to-pointer conversion
conv.bool	4.13	Boolean conversions
conv.class	4.12	Base class conversion
conv.double	4.8	Floating point conversions
conv.fpint	4.9	Floating-integral conversions
conv.fpprom	4.6	Floating point promotion
conv.func	4.3	Function-to-pointer conversion
conv.integral	4.7	Integral conversions
conv.lval	4.1	Lvalue-to-rvalue conversion
conv.mem	4.11	Pointer to member conversions
conv.prom	4.5	Integral promotions
conv.ptr	4.10	Pointer conversions
conv.qual	4.4	Qualification conversions
cpp	16	Preprocessing directives
cpp.concat	16.3.3	The <code>##</code> operator
cpp.cond	16.1	Conditional inclusion
cpp.error	16.5	Error directive
cpp.include	16.2	Source file inclusion
cpp.line	16.4	Line control
cpp.null	16.7	Null directive
cpp.pragma	16.6	Pragma directive
cpp.predefined	16.8	Predefined macro names
cpp.replace	16.3	Macro replacement
cpp.rescan	16.3.4	Rescanning and further replacement
cpp.scope	16.3.5	Scope of macro definitions
cpp.stringize	16.3.2	The <code>#</code> operator
cpp.subst	16.3.1	Argument substitution
dcl.ambig.res	8.2	Ambiguity resolution
dcl.array	8.3.4	Arrays
dcl.asm	7.4	The <code>asm</code> declaration
dcl.dcl	7	Declarations
dcl.decl	8	Declarators
dcl.enum	7.2	Enumeration declarations

dcl.fct	8.3.5	Functions
dcl.fct.def	8.4	Function definitions
dcl.fct.default	8.3.6	Default arguments
dcl.fct.spec	7.1.2	Function specifiers
dcl.friend	7.1.4	The <code>friend</code> specifier
dcl.init	8.5	Initializers
dcl.init.aggr	8.5.1	Aggregates
dcl.init.ref	8.5.3	References
dcl.init.string	8.5.2	Character arrays
dcl.link	7.5	Linkage specifications
dcl.meaning	8.3	Meaning of declarators
dcl.mptr	8.3.3	Pointers to members
dcl.name	8.1	Type names
dcl.ptr	8.3.1	Pointers
dcl.ref	8.3.2	References
dcl.spec	7.1	Specifiers
dcl.stc	7.1.1	Storage class specifiers
dcl.type	7.1.5	Type specifiers
dcl.type.cv	7.1.5.1	The <i>cv-qualifiers</i>
dcl.type.elab	7.1.5.3	Elaborated type specifiers
dcl.type.simple	7.1.5.2	Simple type specifiers
dcl.typedef	7.1.3	The <code>typedef</code> specifier
depr	D	Compatibility features
depr.c.headers	D.1	Standard C library headers
depr.ios.members	D.2	Old iostreams members
depr.istream	D.3.2	Template class <code>istream</code>
depr.istream.cons	D.3.2.1	<code>istream</code> constructors
depr.istream.members	D.3.2.2	Member functions
depr.ostringstream	D.3.3	Template class <code>ostringstream</code>
depr.ostringstream.cons	D.3.3.1	<code>ostringstream</code> constructors
depr.ostringstream.members	D.3.3.2	Member functions
depr.str.strstreams	D.3	<code>char*</code> streams
depr.strstreambuf	D.3.1	Class <code>strstreambuf</code>
depr.strstreambuf.cons	D.3.1.1	<code>strstreambuf</code> constructors
depr.strstreambuf.members	D.3.1.2	Member functions
depr.strstreambuf.virtuals	D.3.1.3	<code>strstreambuf</code> overridden virtual functions
diff	C	Compatibility
diff.anac	C.3	Anachronisms
diff.base.init	C.3.2	Old style base class initializer
diff.basic	C.2.2	Clause <code>_basic_</code> : basic concepts
diff.bound	C.3.4	Cast of bound pointer
diff.c	C.1	Extensions
diff.c++	C.1.2	C++ features added since 1985
diff.class	C.2.7	Clause <code>_class_</code> : classes
diff.class.nonnested	C.3.5	Nonnested classes
diff.cpp	C.2.9	Clause <code>_cpp_</code> : preprocessing directives
diff.dcl	C.2.5	Clause <code>_dcl.dcl_</code> : declarations
diff.decl	C.2.6	Clause <code>_dcl.decl_</code> : declarators
diff.early	C.1.1	C++ features available in 1985
diff.expr	C.2.3	Clause <code>_expr_</code> : expressions
diff.fct.def	C.3.1	Old style function definitions
diff.header.iso646.h	C.4.2.2	Header <code><iso646.h></code>
diff.iso	C.2	C++ and ISO C
diff.lex	C.2.1	Clause <code>_lex_</code> : lexical conventions
diff.library	C.4	Standard C library

diff.malloc	C.4.4.2	Memory allocation functions
diff.mods.to.behavior	C.4.4	Modifications to behavior
diff.mods.to.declarations	C.4.3	Modifications to declarations
diff.mods.to.definitions	C.4.2	Modifications to definitions
diff.mods.to.headers	C.4.1	Modifications to headers
diff.null	C.4.2.3	Macro <code>NULL</code>
diff.offsetof	C.4.4.1	Macro <code>offsetof(type, member-designator)</code> .CW <stddef.h>
diff.special	C.2.8	Clause <code>_special_</code> : special member functions
diff.stat	C.2.4	Clause <code>_stmt.stmt_</code> : statements
diff.this	C.3.3	Assignment to <code>this</code>
diff.wchar.t	C.4.2.1	Type <code>wchar_t</code>
except	15	Exception handling
except.access	15.6	Exceptions and access
except.ctor	15.2	Constructors and destructors
except.handle	15.3	Handling an exception
except.spec	15.4	Exception specifications
except.special	15.5	Special functions
except.terminate	15.5.1	The <code>terminate()</code> function
except.throw	15.1	Throwing an exception
except.unexpected	15.5.2	The <code>unexpected()</code> function
expr	5	Expressions
expr.add	5.7	Additive operators
expr.ass	5.17	Assignment operators
expr.bit.and	5.11	Bitwise AND operator
expr.call	5.2.2	Function call
expr.cast	5.4	Explicit type conversion (cast notation)
expr.comma	5.18	Comma operator
expr.cond	5.16	Conditional operator
expr.const	5.19	Constant expressions
expr.const.cast	5.2.10	Const cast
expr.delete	5.3.5	Delete
expr.dynamic.cast	5.2.6	Dynamic cast
expr.eq	5.10	Equality operators
expr.log.and	5.14	Logical AND operator
expr.log.or	5.15	Logical OR operator
expr.mptr.oper	5.5	Pointer-to-member operators
expr.mul	5.6	Multiplicative operators
expr.new	5.3.4	New
expr.or	5.13	Bitwise inclusive OR operator
expr.post	5.2	Postfix expressions
expr.post.incr	5.2.5	Increment and decrement
expr.pre.incr	5.3.2	Increment and decrement
expr.prim	5.1	Primary expressions
expr.ref	5.2.4	Class member access
expr.reinterpret.cast	5.2.9	Reinterpret cast
expr.rel	5.9	Relational operators
expr.shift	5.8	Shift operators
expr.sizeof	5.3.3	Sizeof
expr.static.cast	5.2.8	Static cast
expr.sub	5.2.1	Subscripting
expr.type.conv	5.2.3	Explicit type conversion (functional notation)
expr.typeid	5.2.7	Type identification
expr.unary	5.3	Unary expressions
expr.unary.op	5.3.1	Unary operators

expr.xor	5.12	Bitwise exclusive OR operator
gram	A	Grammar summary
gram.basic	A.3	Basic concepts
gram.class	A.8	Classes
gram.class.derived	A.9	Derived classes
gram.dcl.dcl	A.6	Declarations
gram.dcl.decl	A.7	Declarators
gram.except	A.13	Exception handling
gram.expr	A.4	Expressions
gram.key	A.1	Keywords
gram.lex	A.2	Lexical conventions
gram.over	A.11	Overloading
gram.special	A.10	Special member functions
gram.stmt.stmt	A.5	Statements
gram.temp	A.12	Templates
intro	1	General
intro.compliance	1.7	Processor compliance
intro.defs	1.3	Definitions
intro.execution	1.8	Program execution
intro.memory	1.5	The C++ memory model
intro.object	1.6	The C++ object model
intro.refs	1.2	Normative references
intro.scope	1.1	Scope
lex	2	Lexical conventions
lex.bool	2.9.5	Boolean literals
lex.ccon	2.9.2	Character literals
lex.comment	2.6	Comments
lex.digraph	2.4	Alternative tokens
lex.fcon	2.9.3	Floating literals
lex.icon	2.9.1	Integer literals
lex.key	2.8	Keywords
lex.literal	2.9	Literals
lex.name	2.7	Identifiers
lex.phases	2.1	Phases of translation
lex.pptoken	2.3	Preprocessing tokens
lex.string	2.9.4	String literals
lex.token	2.5	Tokens
lex.trigraph	2.2	Trigraph sequences
lib.accumulate	26.4.1	Accumulate
lib.adjacent.difference	26.4.4	Adjacent difference
lib.adjustfield.manip	27.4.5.2	adjustfield manipulators
lib.alg.adjacent.find	25.1.5	Adjacent find
lib.alg.binary.search	25.3.3	Binary search
lib.alg.c.library	25.4	C library algorithms
lib.alg.copy	25.2.1	Copy
lib.alg.count	25.1.6	Count
lib.alg.equal	25.1.8	Equal
lib.alg.fill	25.2.5	Fill
lib.alg.find	25.1.2	Find
lib.alg.find.end	25.1.3	Find End
lib.alg.find.first.of	25.1.4	Find First
lib.alg.foreach	25.1.1	For each
lib.alg.generate	25.2.6	Generate
lib.alg.heap.operations	25.3.6	Heap operations
lib.alg.lex.comparison	25.3.8	Lexicographical comparison

lib.alg.merge	25.3.4	Merge
lib.alg.min.max	25.3.7	Minimum and maximum
lib.alg.modifying.operations	25.2	Mutating sequence operations
lib.alg.nonmodifying	25.1	Non-modifying sequence operations
lib.alg.nth.element	25.3.2	Nth element
lib.alg.partitions	25.2.12	Partitions
lib.alg.permutation.generators	25.3.9	Permutation generators
lib.alg.random.shuffle	25.2.11	Random shuffle
lib.alg.remove	25.2.7	Remove
lib.alg.replace	25.2.4	Replace
lib.alg.reverse	25.2.9	Reverse
lib.alg.rotate	25.2.10	Rotate
lib.alg.search	25.1.9	Search
lib.alg.set.operations	25.3.5	Set operations on sorted structures
lib.alg.sort	25.3.1	Sorting
lib.alg.sorting	25.3	Sorting and related operations
lib.alg.swap	25.2.2	Swap
lib.alg.transform	25.2.3	Transform
lib.alg.unique	25.2.8	Unique
lib.algorithms	25	Algorithms library
lib.alloc.errors	18.4.2	Storage allocation errors
lib.allocate	20.4.3.1	allocate
lib.allocator.example	20.4.1.3	Example allocator
lib.allocator.members	20.4.1.1	allocator members
lib.allocator.placement	20.4.1.2	allocator placement new
lib.allocator.requirements	20.1	Allocator requirements
lib.alt.headers	17.3.3.2	Headers
lib.arithmetic.operations	20.3.2	Arithmetic operations
lib.assertions	19.2	Assertions
lib.associative	23.3	Associative containers
lib.associative.reqmts	23.1.2	Associative containers
lib.auto.ptr	20.4.5	Template class auto_ptr
lib.auto.ptr.cons	20.4.5.1	auto_ptr constructors
lib.auto.ptr.members	20.4.5.2	auto_ptr members
lib.back.insert.iter.cons	24.3.2.2.1	back_insert_iterator constructor
lib.back.insert.iter.op*	24.3.2.2.3	back_insert_iterator::operator*
lib.back.insert.iter.op++	24.3.2.2.4	back_insert_iterator::operator++
lib.back.insert.iter.op=	24.3.2.2.2	back_insert_iterator::operator=
lib.back.insert.iter.ops	24.3.2.2	back_insert_iterator operations
lib.back.insert.iterator	24.3.2.1	Template class back_insert_iterator
lib.back.inserter	24.3.2.2.5	back_inserter
lib.bad.alloc	18.4.2.1	Class bad_alloc
lib.bad.cast	18.5.2	Class bad_cast
lib.bad.exception	18.6.1.1	Class bad_exception
lib.bad.typeid	18.5.3	Class bad_typeid
lib.base	20.3.1	Base
lib.basefield.manip	27.4.5.3	basefield manipulators
lib.basic.ios.cons	27.4.4.1	basic_ios constructors
lib.basic.ios.members	27.4.4.2	Member functions
lib.basic.istream.cons	27.6.1.1.1	basic_istream constructors
lib.basic.iterators	24.2.2	Basic iterators
lib.basic.string	21.1.1.3	Template class basic_string
lib.bidirectional.iterators	24.1.4	Bidirectional iterators
lib.binary.search	25.3.3.4	binary_search

<code>lib.bind.1st</code>	20.3.6.2	<code>bind1st</code>
<code>lib.bind.2nd</code>	20.3.6.4	<code>bind2nd</code>
<code>lib.binder.1st</code>	20.3.6.1	Template class <code>binder1st</code>
<code>lib.binder.2nd</code>	20.3.6.3	Template class <code>binder2nd</code>
<code>lib.binders</code>	20.3.6	Binders
<code>lib.bitmask.types</code>	17.2.2.1.2	Bitmask types
<code>lib.bitset.cons</code>	23.2.1.1	bitset constructors
<code>lib.bitset.members</code>	23.2.1.2	bitset members
<code>lib.bitset.operators</code>	23.2.1.3	bitset operators
<code>lib.byte.strings</code>	17.2.2.1.3.1	Byte strings
<code>lib.c.files</code>	27.8.2	C Library files
<code>lib.c.limits</code>	18.2.2	C Library
<code>lib.c.locales</code>	22.3	C Library Locales
<code>lib.c.malloc</code>	20.4.6	C Library
<code>lib.c.math</code>	26.5	C Library
<code>lib.c.strings</code>	21.2	Null-terminated sequence utilities
<code>lib.category.collate</code>	22.2.4	The <code>collate</code> category
<code>lib.category.ctype</code>	22.2.1	The <code>ctype</code> category
<code>lib.category.messages</code>	22.2.7	The message retrieval category
<code>lib.category.monetary</code>	22.2.6	The monetary category
<code>lib.category.numeric</code>	22.2.2	The numeric category
<code>lib.category.time</code>	22.2.5	The time category
<code>lib.character.seq</code>	17.2.2.1.3	Character sequences
<code>lib.class.gslice</code>	26.3.5	The <code>gslice</code> class
<code>lib.class.slice</code>	26.3.3	Class <code>slice</code>
<code>lib.classification</code>	22.1.2.1	Character classification
<code>lib.comparisons</code>	20.3.3	Comparisons
<code>lib.complex</code>	26.2.1	Template class <code>complex</code>
<code>lib.complex.member.ops</code>	26.2.4	<code>complex</code> member operators
<code>lib.complex.members</code>	26.2.3	<code>complex</code> member functions
<code>lib.complex.numbers</code>	26.2	Complex numbers
<code>lib.complex.ops</code>	26.2.5	<code>complex</code> non-member operations
<code>lib.complex.special</code>	26.2.2	<code>complex</code> specializations
<code>lib.complex.transcendentals</code>	26.2.7	<code>complex</code> transcendentals
<code>lib.complex.value.ops</code>	26.2.6	<code>complex</code> value operations
<code>lib.compliance</code>	17.3.1.3	Freestanding implementations
<code>lib.conforming</code>	17.3.4	Conforming implementations
<code>lib.cons.slice</code>	26.3.3.1	<code>slice</code> constructors
<code>lib.cons.slice.arr</code>	26.3.4.1	<code>slice_array</code> constructors
<code>lib.constraints</code>	17.3.3	Constraints on programs
<code>lib.construct</code>	20.4.3.3	<code>construct</code>
<code>lib.container.adapters</code>	23.2.4	Container adapters
<code>lib.container.requirements</code>	23.1	Container requirements
<code>lib.containers</code>	23	Containers library
<code>lib.contents</code>	17.3.1.1	Library contents
<code>lib.conventions</code>	17.2.2	Other conventions
<code>lib.conversions</code>	22.1.2.2	Character conversions
<code>lib.date.time</code>	20.5	Date and time
<code>lib.deallocate</code>	20.4.3.2	<code>deallocate</code>
<code>lib.default allocator</code>	20.4.1	The default allocator
<code>lib.definitions</code>	17.1	Definitions
<code>lib.deque</code>	23.2.2	Template class <code>deque</code>
<code>lib.deque.access</code>	23.2.2.5	<code>deque</code> element access
<code>lib.deque.capacity</code>	23.2.2.4	<code>deque</code> capacity
<code>lib.deque.cons</code>	23.2.2.2	<code>deque</code> constructors, copy, and assignment

lib.deque.iterators	23.2.2.3	deque iterator support
lib.deque.modifiers	23.2.2.6	deque modifiers
lib.deque.types	23.2.2.1	deque types
lib.derivation	17.3.4.7	Derived classes
lib.derived.classes	17.3.3.3	Derived classes
lib.description	17.2	Method of description (Informative)
lib.destroy	20.4.3.4	destroy
lib.diagnostics	19	Diagnostics library
lib.distance.type	24.2.5	distance_type
lib.domain.error	19.1.3	Class domain_error
lib.enumerated.types	17.2.2.1.1	Enumerated types
lib.equal.range	25.3.3.3	equal_range
lib.errno	19.3	Error numbers
lib.exception	19.1.1	Class exception
lib.exception.terminate	18.6.2	Abnormal termination
lib.exception.unexpected	18.6.1	Violating <i>exception-specifications</i>
lib.extern.names	17.3.3.1.3	External linkage
lib.facet ctype.char.dtor	22.2.1.3.1	ctype<char> destructor
lib.facet ctype.char.members	22.2.1.3.2	ctype<char> members
lib.facet ctype.char.virtuals	22.2.1.3.3	ctype<char> overridden virtual functions
lib.facet ctype.special	22.2.1.3	ctype specializations
lib.facet.num.get.members	22.2.2.1.1	num_get members
lib.facet.num.get.virtuals	22.2.2.1.2	num_get virtual functions
lib.facet.num.put.members	22.2.2.2.1	num_put members
lib.facet.num.put.virtuals	22.2.2.2.2	num_put virtual functions
lib.facet.numpunct	22.2.3	The numeric punctuation facet
lib.facet.numpunct.members	22.2.3.1.1	numpunct members
lib.facet.numpunct.virtuals	22.2.3.1.2	numpunct virtual functions
lib.facets.examples	22.2.8	Program-defined facets
lib.file.streams	27.8	File-based streams
lib.filebuf	27.8.1.1	Template class basic_filebuf
lib.filebuf.cons	27.8.1.2	basic_filebuf constructors
lib.filebuf.members	27.8.1.3	Member functions
lib.filebuf.virtuals	27.8.1.4	Overridden virtual functions
lib.floatfield.manip	27.4.5.4	floatfield manipulators
lib.fmtflags.manip	27.4.5.1	fmtflags manipulators
lib.fmtflags.state	27.4.3.2	ios_base fmtflags state functions
lib.forward.iterators	24.1.3	Forward iterators
lib.front.insert.iter.cons	24.3.2.4.1	front_insert_iterator constructor
lib.front.insert.iter.op*	24.3.2.4.3	front_insert_iterator::operator*
lib.front.insert.iter.op++	24.3.2.4.4	front_insert_iterator::operator++
lib.front.insert.iter.op=	24.3.2.4.2	front_insert_iterator::operator=
lib.front.insert.iter.ops	24.3.2.4	front_insert_iterator operations
lib.front.insert.iterator	24.3.2.3	Template class front_insert_iterator
lib.front.inserter	24.3.2.4.5	front_inserter
lib.fstreams	27.8.1	File streams
lib.function.objects	20.3	Function objects
lib.function.pointer.adaptors	20.3.7	Adaptors for pointers to functions
lib.functions.within.classes	17.2.2.2	Functions within classes
lib.global.functions	17.3.4.3	Global functions
lib.global.names	17.3.3.1.2	Global names
lib.gslice.access	26.3.5.2	gslice access functions
lib.gslice.array.assign	26.3.6.2	gslice_array assignment
lib.gslice.array.comp.assign	26.3.6.3	gslice_array computed assignment
lib.gslice.array.cons	26.3.6.1	gslice_array constructors

lib.gslice.array.fill	26.3.6.4	gslice_array fill function
lib.gslice.cons	26.3.5.1	gslice constructors
lib.handler.functions	17.3.3.5	Handler functions
lib.headers	17.3.1.2	Headers
lib ifstream	27.8.1.5	Template class basic_ifstream
lib ifstream.cons	27.8.1.6	basic_ifstream constructors
lib ifstream.members	27.8.1.7	Member functions
lib.includes	25.3.5.1	includes
lib.indirect.array.assign	26.3.8.2	indirect_array assignment
lib.indirect.array.comp.assign	26.3.8.3	indirect_array computed assignment
lib.indirect.array.cons	26.3.8.1	indirect_array constructors
lib.indirect.array.fill	26.3.8.4	indirect_array fill function
lib.inner.product	26.4.2	Inner product
lib.input.iterators	24.1.1	Input iterators
lib.input.output	27	Input/output library
lib.input.streams	27.6.1	Input streams
lib.insert.iter.cons	24.3.2.6.1	insert_iterator constructor
lib.insert.iter.op*	24.3.2.6.3	insert_iterator::operator*
lib.insert.iter.op++	24.3.2.6.4	insert_iterator::operator++
lib.insert.iter.op=	24.3.2.6.2	insert_iterator::operator=
lib.insert.iter.ops	24.3.2.6	insert_iterator operations
lib.insert.iterator	24.3.2.5	Template class insert_iterator
lib.insert.iterators	24.3.2	Insert iterators
lib inserter	24.3.2.6.5	inserter
lib.invalid.argument	19.1.4	Class invalid_argument
lib ios	27.4.4	Template class basic_ios
lib ios.base	27.4.3	Class ios_base
lib ios.base.cons	27.4.3.5	ios_base constructors
lib ios.base.locales	27.4.3.3	ios_base locale functions
lib ios.base.storage	27.4.3.4	ios_base storage functions
lib ios.traits	27.4.2	Template struct ios_traits
lib ios.traits.convert	27.4.2.4	ios_traits conversion functions
lib ios.traits.tests	27.4.2.3	ios_traits test functions
lib ios.traits.types	27.4.2.1	ios_traits types
lib ios.traits.values	27.4.2.2	ios_traits value functions
lib ios.types	27.4.3.1	Types
lib ios::Init	27.4.3.1.6	Class ios_base::Init
lib ios::failure	27.4.3.1.1	Class ios_base::failure
lib ios::fmtflags	27.4.3.1.2	Type ios_base::fmtflags
lib ios::iostate	27.4.3.1.3	Type ios_base::iostate
lib ios::openmode	27.4.3.1.4	Type ios_base::openmode
lib ios::seekdir	27.4.3.1.5	Type ios_base::seekdir
lib.iostate.flags	27.4.4.3	basic_ios iostate flags functions
lib.iostream.format	27.6	Formatting and manipulators
lib.iostream.forward	27.2	Forward declarations
lib.iostream.objects	27.3	Standard iostream objects
lib.iostreams.base	27.4	Iostreams base classes
lib.iostreams.char.t	27.1.2.1	Type CHAR_T
lib.iostreams.definitions	27.1.1	Definitions
lib.iostreams.int.t	27.1.2.2	Type INT_T
lib.iostreams.off.t	27.1.2.3	Type OFF_T
lib.iostreams.pos.t	27.1.2.4	Type POS_T
lib.iostreams.requirements	27.1	Iostreams requirements
lib.iostreams.type.reqmts	27.1.2	Type requirements

lib.istream	27.6.1.1	Template class <code>basic_istream</code>	
lib.istream.formatted	27.6.1.2	Formatted input functions	
lib.istream.formatted.reqmts	27.6.1.2.1	Common requirements	
lib.istream.iterator	24.4.1	Template class <code>istream_iterator</code>	
lib.istream.manip	27.6.1.4	Standard <code>basic_istream</code> manipulators	
lib.istream.prefix	27.6.1.1.2	<code>basic_istream</code> prefix and suffix	
lib.istream.unformatted	27.6.1.3	Unformatted input functions	
lib.istream::extractors	27.6.1.2.2	<code>basic_istream::operator>></code>	
lib.istreambuf.iterator	24.4.3	Template class <code>istreambuf_iterator</code>	
lib.istreambuf.iterator.cons	24.4.3.2	<code>istreambuf_iterator</code> constructors	
lib.istreambuf.iterator::equal	24.4.3.5	<code>istreambuf_iterator::equal</code>	
lib.istreambuf.iterator::op!=	24.4.3.8	<code>operator!=</code>	
lib.istreambuf.iterator::op*	24.4.3.3	<code>istreambuf_iterator::operator*</code>	
lib.istreambuf.iterator::op++	24.4.3.4	<code>istreambuf_iterator::operator++</code>	
lib.istreambuf.iterator::op==	24.4.3.7	<code>operator==</code>	
lib.istreambuf.iterator::proxy	24.4.3.1	Template	class
		<code>istreambuf_iterator::proxy</code>	
lib.istringstream	27.7.2	Template class <code>basic_istringstream</code>	
lib.istringstream.cons	27.7.2.1	<code>basic_istringstream</code> constructors	
lib.istringstream.members	27.7.2.2	Member functions	
lib.iterator.category	24.2.3	<code>iterator_category</code>	
lib.iterator.category.i	24.4.3.6	<code>iterator_category</code>	
lib.iterator.operations	24.2.6	Iterator operations	
lib.iterator.primitives	24.2	Iterator primitives	
lib.iterator.requirements	24.1	Iterator requirements	
lib.iterator.tags	24.1.6	Iterator tags	
lib.iterators	24	Iterators library	
lib.language.support	18	Language support library	
lib.length.error	19.1.5	Class <code>length_error</code>	
lib.library	17	Library introduction	
lib.limits	18.2.1	Numeric limits	
lib.list	23.2.3	Template class <code>list</code>	
lib.list.access	23.2.3.5	<code>list</code> element access	
lib.list.capacity	23.2.3.4	<code>list</code> capacity	
lib.list.cons	23.2.3.2	<code>list</code> constructors, copy, and assignment	
lib.list.iterators	23.2.3.3	<code>list</code> iterator support	
lib.list.modifiers	23.2.3.6	<code>list</code> modifiers	
lib.list.ops	23.2.3.7	<code>list</code> operations	
lib.list.types	23.2.3.1	<code>list</code> types	
lib.locale	22.1.1	Class <code>locale</code>	
lib.locale.categories	22.2	Standard <code>locale</code> categories	
lib.locale.category	22.1.1.1.1	Type <code>locale::category</code>	
lib.locale.codecvt	22.2.1.4	Template class <code>codecvt</code>	
lib.locale.codecvt.byname	22.2.1.5	Template class <code>codecvt_byname</code>	
lib.locale.codecvt.members	22.2.1.4.1	<code>codecvt</code> members	
lib.locale.codecvt.virtuals	22.2.1.4.2	<code>codecvt</code> virtual functions	
lib.locale.collate	22.2.4.1	Template class <code>collate</code>	
lib.locale.collate.byname	22.2.4.2	Template class <code>collate_byname</code>	
lib.locale.collate.members	22.2.4.1.1	<code>collate</code> members	
lib.locale.collate.virtuals	22.2.4.1.2	<code>collate</code> virtual functions	
lib.locale.cons	22.1.1.2	<code>locale</code> constructors and destructor	
lib.locale.convenience	22.1.2	Convenience interfaces	
lib.locale ctype	22.2.1.1	Template class <code>ctype</code>	
lib.locale ctype.byname	22.2.1.2	Template class <code>ctype_byname</code>	
lib.locale ctype.members	22.2.1.1.1	<code>ctype</code> members	

lib.locale ctype.virtuals	22.2.1.1.2	ctype virtual functions
lib.locale.facet	22.1.1.1.2	Class locale::facet
lib.locale.id	22.1.1.1.3	Class locale::id
lib.locale.members	22.1.1.3	locale members
lib.locale.messages	22.2.7.1	Template class messages
lib.locale.messages.byname	22.2.7.2	Template class messages_byname
lib.locale.messages.members	22.2.7.1.1	messages members
lib.locale.messages.virtuals	22.2.7.1.2	messages virtual functions
lib.locale.money.get	22.2.6.1	Template class money_get
lib.locale.money.get.members	22.2.6.1.1	money_get members
lib.locale.money.get.virtuals	22.2.6.1.2	money_get virtual functions
lib.locale.money.put	22.2.6.2	Template class money_put
lib.locale.money.put.members	22.2.6.2.1	money_put members
lib.locale.money.put.virtuals	22.2.6.2.2	money_put virtual functions
lib.locale.money.punct	22.2.6.3	Template class moneypunct
lib.locale.money.punct.byname	22.2.6.4	Template class moneypunct_byname
lib.locale.money.punct.members	22.2.6.3.1	moneypunct members
lib.locale.money.punct.virtuals	22.2.6.3.2	moneypunct virtual functions
lib.locale.num.get	22.2.2.1	Template class num_get
lib.locale.num.put	22.2.2.2	Template class num_put
lib.locale.numpunct	22.2.3.1	Template class numpunct
lib.locale.numpunct.byname	22.2.3.2	Template class numpunct_byname
lib.locale.operators	22.1.1.4	locale operators
lib.locale.statics	22.1.1.5	locale static members
lib.locale.time.get	22.2.5.1	Template class time_get
lib.locale.time.get.byname	22.2.5.2	Template class time_get_byname
lib.locale.time.get.members	22.2.5.1.1	time_get members
lib.locale.time.get.virtuals	22.2.5.1.2	time_get virtual functions
lib.locale.time.put	22.2.5.3	Template class time_put
lib.locale.time.put.byname	22.2.5.4	Template class time_put_byname
lib.locale.time.put.members	22.2.5.3.1	time_put members
lib.locale.time.put.virtuals	22.2.5.3.2	time_put virtual functions
lib.locale.types	22.1.1.1	locale types
lib.locales	22.1	Locales
lib.localization	22	Localization library
lib.logic.error	19.1.2	Class logic_error
lib.logical.operations	20.3.4	Logical operations
lib.lower.bound	25.3.3.1	lower_bound
lib.macro.names	17.3.3.1.1	Macro names
lib.make.heap	25.3.6.3	make_heap
lib.map	23.3.1	Template class map
lib.map.access	23.3.1.5	map element access
lib.map.capacity	23.3.1.4	map capacity
lib.map.cons	23.3.1.2	map constructors, copy, and assignment
lib.map.iterators	23.3.1.3	map iterator support
lib.map.modifiers	23.3.1.6	map modifiers
lib.map.observers	23.3.1.7	map observers
lib.map.ops	23.3.1.8	map operations
lib.map.types	23.3.1.1	map types
lib.mask.array.assign	26.3.7.2	mask_array assignment

lib.mask.array.comp.assign	26.3.7.3	mask_array computed assignment
lib.mask.array.cons	26.3.7.1	mask_array constructors
lib.mask.array.fill	26.3.7.4	mask_array fill function
lib.member.functions	17.3.4.4	Member functions
lib.memory	20.4	Memory
lib.memory.primitives	20.4.3	Memory handling primitives
lib.mismatch	25.1.7	Mismatch
lib.multibyte.strings	17.2.2.1.3.2	Multibyte strings
lib.multimap	23.3.2	Template class multimap
lib.multiset	23.3.4	Template class multiset
lib.narrow.stream.objects	27.3.1	Narrow stream objects
lib.negators	20.3.5	Negators
lib.new.delete	18.4.1	Storage allocation and deallocation
lib.new.delete.array	18.4.1.2	Array forms
lib.new.delete.placement	18.4.1.3	Placement forms
lib.new.delete.single	18.4.1.1	Single-object forms
lib.new.handler	18.4.2.2	Type new_handler
lib.numarray	26.3	Numeric arrays
lib.numeric.limits	18.2.1.1	Template class numeric_limits
lib.numeric.limits.members	18.2.1.2	numeric_limits members
lib.numeric.ops	26.4	Generalized numeric operations
lib.numeric.requirements	26.1	Numeric type requirements
lib.numeric.special	18.2.1.4	numeric_limits specializations
lib.numerics	26	Numerics library
lib.objects.within.classes	17.2.2.3	Private members
lib.ofstream	27.8.1.8	Template class basic_ofstream
lib.ofstream.cons	27.8.1.9	basic_ofstream constructors
lib.ofstream.members	27.8.1.10	Member functions
lib.operators	20.2.1	Operators
lib.organization	17.3.1	Library contents and organization
lib ostream	27.6.2.1	Template class basic_ostream
lib ostream.cons	27.6.2.2	basic_ostream constructors
lib ostream.formatted	27.6.2.4	Formatted output functions
lib ostream.formatted.reqmts	27.6.2.4.1	Common requirements
lib ostream.inserters	27.6.2.4.2	basic_ostream::operator<<
lib ostream.iterator	24.4.2	Template class ostream_iterator
lib ostream.manip	27.6.2.6	Standard basic_ostream manipulators
lib ostream.prefix	27.6.2.3	basic_ostream prefix and suffix functions
lib ostream.unformatted	27.6.2.5	Unformatted output functions
lib ostreambuf.iter.cons	24.4.4.1	ostreambuf_iterator constructors
lib ostreambuf.iter.ops	24.4.4.2	ostreambuf_iterator operations
lib ostreambuf.iterator	24.4.4	Template class ostreambuf_iterator
lib ostreambuf.iterator.nonmembers	24.4.4.3	ostreambuf_iterator non-member operations
lib ostreamstringstream	27.7.2.3	Class basic_ostreamstringstream
lib ostreamstringstream.cons	27.7.2.4	basic_ostreamstringstream constructors
lib ostreamstringstream.members	27.7.2.5	Member functions
lib.out.of.range	19.1.6	Class out_of_range
lib.output.iterators	24.1.2	Output iterators
lib.output.streams	27.6.2	Output streams
lib.overflow.error	19.1.9	Class overflow_error
lib.pairs	20.2.2	Pairs
lib.partial.sort	25.3.1.3	partial_sort
lib.partial.sort.copy	25.3.1.4	partial_sort_copy

lib.partial.sum	26.4.3	Partial sum	
lib.pop.heap	25.3.6.2	pop_heap	
lib.predef.iterators	24.3	Predefined iterators	
lib.priority.queue	23.2.4.2	Template class priority_queue	
lib.priqueue.cons	23.2.4.2.1	priority_queue constructors	
lib.priqueue.members	23.2.4.2.2	priority_queue members	
lib.protection.within.classes	17.3.4.6	Protection within classes	
lib.push.heap	25.3.6.1	push_heap	
lib.queue	23.2.4.1	Template class queue	
lib.random.access.iterators	24.1.5	Random access iterators	
lib.range.error	19.1.8	Class range_error	
lib.reentrancy	17.3.4.5	Reentrancy	
lib.replacement.functions	17.3.3.4	Replacement functions	
lib.requirements	17.3	Library-wide requirements	
lib.res.on.arguments	17.3.3.7	Function arguments	
lib.res.on.exception.handling	17.3.4.8	Restrictions on exception handling	
lib.res.on.functions	17.3.3.6	Other functions	
lib.res.on.headers	17.3.4.1	Headers	
lib.res.on.macro.definitions	17.3.4.2	Restrictions on macro definitions	
lib.reserved.names	17.3.3.1	Reserved names	
lib.reverse.bidir.iter	24.3.1.1	Template	class
lib.reverse.bidir.iter.cons	24.3.1.2.1	reverse_bidirectional_iterator reverse_bidirectional_iterator constructor	
lib.reverse.bidir.iter.conv	24.3.1.2.2	Conversion	
lib.reverse.bidir.iter.op++	24.3.1.2.4	operator++	
lib.reverse.bidir.iter.op--	24.3.1.2.5	operator--	
lib.reverse.bidir.iter.op.star	24.3.1.2.3	operator*	
lib.reverse.bidir.iter.op==	24.3.1.2.6	operator==	
lib.reverse.bidir.iter.ops	24.3.1.2	reverse_bidirectional_iterator operations	
lib.reverse.iter.cons	24.3.1.4.1	reverse_iterator constructor	
lib.reverse.iter.conv	24.3.1.4.2	Conversion	
lib.reverse.iter.op++	24.3.1.4.4	operator++	
lib.reverse.iter.op--	24.3.1.4.5	operator--	
lib.reverse.iter.op.star	24.3.1.4.3	operator*	
lib.reverse.iter.op==	24.3.1.4.6	operator==	
lib.reverse.iter.ops	24.3.1.4	reverse_iterator operations	
lib.reverse.iterator	24.3.1.3	Template class reverse_iterator	
lib.reverse.iterators	24.3.1	Reverse iterators	
lib.round.style	18.2.1.3	Type float_round_style	
lib.runtime.error	19.1.7	Class runtime_error	
lib.sequence.reqmts	23.1.1	Sequences	
lib.sequences	23.2	Sequences	
lib.set	23.3.3	Template class set	
lib.set.capacity	23.3.3.4	set capacity	
lib.set.cons	23.3.3.2	set constructors, copy, and assignment	
lib.set.difference	25.3.5.4	set_difference	
lib.set.intersection	25.3.5.3	set_intersection	
lib.set.iterators	23.3.3.3	set iterator support	
lib.set.modifiers	23.3.3.5	set modifiers	
lib.set.new.handler	18.4.2.3	set_new_handler	
lib.set.observers	23.3.3.6	set observers	
lib.set.ops	23.3.3.7	set operations	
lib.set.symmetric.difference	25.3.5.5	set_symmetric_difference	

lib.set.terminate	18.6.2.2	set_terminate	
lib.set.types	23.3.3.1	set types	
lib.set.unexpected	18.6.1.3	set_unexpected	
lib.set.union	25.3.5.2	set_union	
lib.slice.access	26.3.3.2	slice access functions	
lib.slice.arr.assign	26.3.4.2	slice_array assignment	
lib.slice.arr.comp.assign	26.3.4.3	slice_array computed assignment	
lib.slice.arr.fill	26.3.4.4	slice_array fill function	
lib.sort	25.3.1.1	sort	
lib.sort.heap	25.3.6.4	sort_heap	
lib.specialized.algorithms	20.4.4	Specialized algorithms	
lib.stable.sort	25.3.1.2	stable_sort	
lib.stack	23.2.4.3	Template class stack	
lib.std.exceptions	19.1	Exception classes	
lib.std.ios.manip	27.4.5	ios_base manipulators	
lib.std.iterator.tags	24.2.1	Standard iterator tags	
lib.std.manip	27.6.3	Standard manipulators	
lib.storage.iterator	20.4.2	Raw storage iterator	
lib.stream.buffer	27.5	Stream buffers	
lib.stream.iterators	24.4	Stream iterators	
lib.stream.types	27.4.1	Types	
lib.streambuf	27.5.2	Template	class
		basic_streambuf<charT, traits>	
lib.streambuf.buffer	27.5.2.2.2	Buffer management and positioning	
lib.streambuf.cons	27.5.2.1	basic_streambuf constructors	
lib.streambuf.get.area	27.5.2.3.1	Get area access	
lib.streambuf.locales	27.5.2.2.1	Locales	
lib.streambuf.members	27.5.2.2	basic_streambuf public member functions	
lib.streambuf.protected	27.5.2.3	basic_streambuf protected member functions	
lib.streambuf.pub.get	27.5.2.2.3	Get area	
lib.streambuf.pub.pback	27.5.2.2.4	Putback	
lib.streambuf.pub.put	27.5.2.2.5	Put area	
lib.streambuf.put.area	27.5.2.3.2	Put area access	
lib.streambuf.reqts	27.5.1	Stream buffer requirements	
lib.streambuf.virt.buffer	27.5.2.4.2	Buffer management and positioning	
lib.streambuf.virt.get	27.5.2.4.3	Get area	
lib.streambuf.virt.locales	27.5.2.4.1	Locales	
lib.streambuf.virt.pback	27.5.2.4.4	Putback	
lib.streambuf.virt.put	27.5.2.4.5	Put area	
lib.streambuf.virtuals	27.5.2.4	basic_streambuf virtual functions	
lib.string	21.1.2	Class string	
lib.string.access	21.1.1.7	basic_string element access	
lib.string.capacity	21.1.1.6	basic_string capacity	
lib.string.char.traits	21.1.1.1	Template class string_char_traits	
lib.string.char.traits.members	21.1.1.2	string_char_traits members	
lib.string.classes	21.1	String classes	
lib.string.cons	21.1.1.4	basic_string constructors	
lib.string.iterators	21.1.1.5	basic_string iterator support	
lib.string.modifiers	21.1.1.8	basic_string modifiers	
lib.string.nonmembers	21.1.1.10	basic_string non-member functions	
lib.string.ops	21.1.1.9	basic_string string operations	
lib.string.streams	27.7	String-based streams	
lib.string.traits.members	21.1.3	string_char_traits<char> members	

lib.string::append	21.1.1.8.2	basic_string::append
lib.string::assign	21.1.1.8.3	basic_string::assign
lib.string::compare	21.1.1.9.8	basic_string::compare
lib.string::copy	21.1.1.8.7	basic_string::copy
lib.string::find	21.1.1.9.1	basic_string::find
lib.string::find.first.not.of	21.1.1.9.5	basic_string::find_first_not_of
lib.string::find.first.of	21.1.1.9.3	basic_string::find_first_of
lib.string::find.last.not.of	21.1.1.9.6	basic_string::find_last_not_of
lib.string::find.last.of	21.1.1.9.4	basic_string::find_last_of
lib.string::insert	21.1.1.8.4	basic_string::insert
lib.string::op!=	21.1.1.10.3	operator!=
lib.string::op+	21.1.1.10.1	operator+
lib.string::op+=	21.1.1.8.1	basic_string::operator+=
lib.string::op<	21.1.1.10.4	operator<
lib.string::op<=	21.1.1.10.6	operator<=
lib.string::op>	21.1.1.10.5	operator>
lib.string::op>=	21.1.1.10.7	operator>=
lib.string::operator==	21.1.1.10.2	operator==
lib.string::remove	21.1.1.8.5	basic_string::remove
lib.string::replace	21.1.1.8.6	basic_string::replace
lib.string::rfind	21.1.1.9.2	basic_string::rfind
lib.string::substr	21.1.1.9.7	basic_string::substr
lib.string::swap	21.1.1.8.8	basic_string::swap
lib.stringbuf	27.7.1	Template class basic_stringbuf
lib.stringbuf.cons	27.7.1.1	basic_stringbuf constructors
lib.stringbuf.members	27.7.1.2	Member functions
lib.stringbuf.virtuals	27.7.1.3	Overridden virtual functions
lib.strings	21	Strings library
lib.structure	17.2.1	Structure of each subclass
lib.structure.requirements	17.2.1.2	Requirements
lib.structure.see.also	17.2.1.4	C Library
lib.structure.specifications	17.2.1.3	Specifications
lib.structure.summary	17.2.1.1	Summary
lib.support.dynamic	18.4	Dynamic memory management
lib.support.exception	18.6	Exception handling
lib.support.limits	18.2	Implementation properties
lib.support.rtti	18.5	Type identification
lib.support.runtime	18.7	Other runtime support
lib.support.start.term	18.3	Start and termination
lib.support.types	18.1	Types
lib.template.bitset	23.2.1	Template class bitset
lib.template.gslicing.array	26.3.6	Template class gslicing_array
lib.template.indirect.array	26.3.8	Template class indirect_array
lib.template.mask.array	26.3.7	Template class mask_array
lib.template.slice.array	26.3.4	Template class slice_array
lib.template.string	21.1.1	Template class basic_string
lib.template.valarray	26.3.1	Template class valarray
lib.temporary.buffer	20.4.3.5	Temporary buffers
lib.terminate	18.6.2.3	terminate
lib.terminate.handler	18.6.2.1	Type terminate_handler
lib.type.descriptions	17.2.2.1	Type descriptions
lib.type.info	18.5.1	Class type_info
lib.unexpected	18.6.1.4	unexpected
lib.unexpected.handler	18.6.1.2	Type unexpected_handler
lib.uninitialized.copy	20.4.4.1	uninitialized_copy

lib.uninitialized.fill	20.4.4.2	uninitialized_fill
lib.uninitialized.fill.n	20.4.4.3	uninitialized_fill
lib.upper.bound	25.3.3.2	upper_bound
lib.using	17.3.2	Using the library
lib.using.headers	17.3.2.1	Headers
lib.using.linkage	17.3.2.2	Linkage
lib.utilities	20	General utilities library
lib.utility	20.2	Utility components
lib.valarray.access	26.3.1.3	valarray element access
lib.valarray.assign	26.3.1.2	valarray assignment
lib.valarray.binary	26.3.2.1	valarray binary operators
lib.valarray.cassign	26.3.1.6	valarray computed assignment
lib.valarray.comparison	26.3.2.2	valarray comparison operators
lib.valarray.cons	26.3.1.1	valarray constructors
lib.valarray.members	26.3.1.7	valarray member functions
lib.valarray.min.max	26.3.2.3	valarray min and max functions
lib.valarray.nonmembers	26.3.2	valarray non-member operations
lib.valarray.sub	26.3.1.4	valarray subset operations
lib.valarray.transcend	26.3.2.4	valarray transcendentals
lib.valarray.unary	26.3.1.5	valarray unary operators
lib.value.type	24.2.4	value_type
lib.vector	23.2.5	Template class vector
lib.vector.access	23.2.5.5	vector element access
lib.vector.bool	23.2.6	Class vector<bool>
lib.vector.capacity	23.2.5.4	vector capacity
lib.vector.cons	23.2.5.2	vector constructors, copy, and assignment
lib.vector.iterators	23.2.5.3	vector iterator support
lib.vector.modifiers	23.2.5.6	vector modifiers
lib.vector.types	23.2.5.1	vector types
lib.wide.characters	17.2.2.1.3.3	Wide-character sequences
lib.wide.stream.objects	27.3.2	Wide stream objects
lib.wstring	21.1.4	Class wstring
lib.wstring.members	21.1.5	string_char_traits<wchar_t> members
limits	B	Implementation quantities
namespace.alias	7.3.2	Namespace or class alias
namespace.def	7.3.1	Namespace definition
namespace.memdef	7.3.1.4	Namespace member definitions
namespace.qual	7.3.1.1	Explicit qualification
namespace.scope	7.3.1.3	Namespace scope
namespace.udecl	7.3.3	The using declaration
namespace.udir	7.3.4	Using directive
namespace.unnamed	7.3.1.2	Unnamed namespaces
over	13	Overloading
over.ass	13.5.3	Assignment
over.best.ics	13.3.3.1	Implicit conversion sequences
over.binary	13.5.2	Binary operators
over.built	13.6	Built-in operators
over.call	13.5.4	Function call
over.call.func	13.3.1.1.1	Call to named function
over.call.object	13.3.1.1.2	Call to object of class type
over.dcl	13.2	Declaration matching
over.ics.ellipsis	13.3.3.1.3	Ellipsis conversion sequences
over.ics.rank	13.3.3.2	Ranking implicit conversion sequences
over.ics.ref	13.3.3.1.4	Reference binding
over.ics.scs	13.3.3.1.1	Standard conversion sequences

over.ics.user	13.3.3.1.2	User-defined conversion sequences
over.inc	13.5.7	Increment and decrement
over.load	13.1	Overloadable declarations
over.match	13.3	Overload resolution
over.match.best	13.3.3	Best Viable Function
over.match.call	13.3.1.1	Function call syntax
over.match.ctor	13.3.1.4	Initialization by constructor
over.match.funcs	13.3.1	Candidate functions and argument lists
over.match.oper	13.3.1.2	Operators in expressions
over.match.user	13.3.1.3	Initialization by user-defined conversions
over.match.viable	13.3.2	Viable functions
over.oper	13.5	Overloaded operators
over.over	13.4	Address of overloaded function
over.ref	13.5.6	Class member access
over.sub	13.5.5	Subscripting
over.unary	13.5.1	Unary operators
special	12	Special member functions
stmt.ambig	6.8	Ambiguity resolution
stmt.block	6.3	Compound statement or block
stmt.break	6.6.1	The <code>break</code> statement
stmt.cont	6.6.2	The <code>continue</code> statement
stmt.dcl	6.7	Declaration statement
stmt.do	6.5.2	The <code>do</code> statement
stmt.expr	6.2	Expression statement
stmt.for	6.5.3	The <code>for</code> statement
stmt.goto	6.6.4	The <code>goto</code> statement
stmt.if	6.4.1	The <code>if</code> statement
stmt.iter	6.5	Iteration statements
stmt.jump	6.6	Jump statements
stmt.label	6.1	Labeled statement
stmt.return	6.6.3	The <code>return</code> statement
stmt.select	6.4	Selection statements
stmt.stmt	6	Statements
stmt.switch	6.4.2	The <code>switch</code> statement
stmt.while	6.5.1	The <code>while</code> statement
syntax	1.4	Syntax notation
temp	14	Templates
temp.arg	14.8	Template arguments
temp.arg.explicit	14.10.1	Explicit template argument specification
temp.class.order	14.6.2	Partial ordering of class template specializations
temp.class.spec	14.6	Class template specializations
temp.class.spec.match	14.6.1	Matching of class template specializations
temp.deduct	14.10.2	Template argument deduction
temp.dep	14.2.3	Dependent names
temp.encl	14.2.2	Names from the template's enclosing scope
temp.explicit	14.4	Explicit instantiation
temp.fct	14.10	Function templates
temp.friend	14.12	Friends
temp.func.order	14.10.6	Partial ordering of function templates
temp.inject	14.2.4	Non-local names declared within a template
temp.inst	14.3	Template instantiation
temp.linkage	14.3.1	Template linkage
temp.local	14.2.1	Locally declared names
temp.mem.func	14.11	Member function templates
temp.names	14.1	Template names

temp.opref	14.3.3	Instantiation of operator->
temp.over	14.10.3	Overload resolution
temp.over.link	14.10.4	Overloading and linkage
temp.over.spec	14.10.5	Overloading and specialization
temp.param	14.7	Template parameters
temp.point	14.3.2	Point of instantiation
temp.res	14.2	Name resolution
temp.spec	14.5	Template specialization
temp.static	14.13	Static members and variables
temp.type	14.9	Type equivalence