Accredited Standards Committee*
X3, INFORMATION PROCESSING SYSTEMS

Doc No: X3J16 / 96– 0109
WG21 / N0927
Date: 28 May 1996
Project: Programming Language C++

Reply to: Andrew Koenig
AT&T Research
PO Box 636
600 Mountain Avenue
Room 2C– 306
Murray Hill, NJ 07974 USA
ark@research.att.com

Concordance for July 1996 (Stockholm) C++ Working Paper

This document provides a cross-refrerence 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.compliance	Implementation compliance
1.4	intro.defs	Definitions
1.5	syntax	Syntax notation
1.6	intro.memory	The C++ memory model
1.7	intro.object	The C++ object model
1.8	intro.execution	Program execution
2	lex	Lexical conventions
2.1	lex.phases	Phases of translation
2.2	lex.charset	Basic source character set
2.3	lex.trigraph	Trigraph sequences
2.4	lex.pptoken	Preprocessing tokens
2.5	lex.digraph	Alternative tokens
2.6	lex.token	Tokens
2.7	lex.comment	Comments
2.8	lex.header	Header names
2.9	lex.ppnumber	Preprocessing numbers
2.10	lex.name	Identifiers
2.11	lex.key	Keywords
2.12		Operators and punctuators
2.13	lex.literal	Literals
2.13.1	lex.icon	Integer literals
2.13.2	lex.ccon	Character literals
2.13.3	lex.fcon	Floating literals
2.13.4	lex.string	String literals
2.13.5	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.pdecl	Point of declaration
3.3.2	basic.scope.local	Local scope
3.3.3	basic.scope.proto	Function prototype scope
3.3.4		Function scope
3.3.5	basic.scope.namespace	Namespace scope
3.3.6	basic.scope.class	Class scope
3.3.7	basic.scope.hiding	Name hiding
3.4	basic.lookup	Name look up
3.4.1	basic.lookup.unqual	Unqualified name look up
3.4.2	basic.lookup.qual	Qualified name look up
3.4.2.1	class.qual	Class members
3.4.2.2	namespace.qual	Namespace members
3.4.3	basic.lookup.elab	Elaborated type specifiers
3.4.4	basic.lookup.classref	Class member access
3.4.5	basic.lookup.udir	Using directives and namespace aliases
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
3.10	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.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.pseudo	Pseudo destructor call
5.2.5	expr.ref	Class member access
5.2.6	expr.post.incr	Increment and decrement
5.2.7	expr.dynamic.cast	Dynamic cast
5.2.8	expr.typeid	Type identification
5.2.9	expr.static.cast	Static cast
5.2.10	expr.reinterpret.cast	Reinterpret cast
5.2.11	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 `if` statement
6.4.2	stmt.switch	The `switch` statement
6.5	stmt.iter	Iteration statements
6.5.1	stmt.while	The `while` statement
6.5.2	stmt.do	The `do` statement
6.5.3	stmt.for	The `for` statement
6.6	stmt.jump	Jump statements
6.6.1	stmt.break	The `break` statement
6.6.2	stmt.cont	The `continue` statement
6.6.3	stmt.return	The `return` statement
6.6.4	stmt.goto	The `goto` 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 `typedef` specifier
7.1.4	dcl.friend	The `friend` specifier
7.1.5	dcl.type	Type specifiers
7.1.5.1	dcl.type.cv	The *cv-qualifiers*
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.unnamed	Unnamed namespaces
7.3.1.2	namespace.memdef	Namespace member definitions
7.3.2	namespace.alias	Namespace alias
7.3.3	namespace.udecl	The `using` declaration
7.3.4	namespace.udir	Using directive
7.4	dcl.asm	The `asm` 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

13.3.1.2	over.match.oper	Operators in expressions
13.3.1.3	over.match.ctor	Initialization by constructor
13.3.1.4	over.match.copy	Copy-initialization of class by user-defined conversion
13.3.1.5	over.match.conv	Initialization by conversion function
13.3.1.6	over.match.ref	Initialization by conversion function for direct reference binding
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.param	Template parameters
14.2	temp.names	Names of template specializations
14.3	temp.arg	Template arguments
14.4	temp.type	Type equivalence
14.5	temp.decls	Template declarations
14.5.1	temp.class	Class templates
14.5.1.1	temp.mem.func	Member functions of class templates
14.5.1.2	temp.mem.class	Member classes of class templates
14.5.1.3	temp.static	Static data members of class templates
14.5.2	temp.mem	Member templates
14.5.3	temp.friend	Friends
14.5.4	temp.class.spec	Class template partial specializations
14.5.4.1	temp.class.spec.match	Matching of class template partial specializations
14.5.4.2	temp.class.order	Partial ordering of class template specializations
14.5.4.3	temp.class.spec.mfunc	Members of class template specializations
14.5.5	temp.fct	Function templates
14.5.5.1	temp.over.link	Function template overloading
14.5.5.2	temp.func.order	Partial ordering of function templates
14.6	temp.res	Name resolution
14.6.1	temp.local	Locally declared names
14.6.2	temp.dep	Dependent names
14.6.3	temp.nondep	Non-dependent names
14.6.4	temp.point	Point of instantiation
14.6.5	temp.inject	Friend names declared within a class template
14.7	temp.spec	Template specialization
14.7.1	temp.inst	Implicit instantiation
14.7.2	temp.explicit	Explicit instantiation
14.7.3	temp.expl.spec	Explicit specialization

14.8	temp.fct.spec	Function template specializations
14.8.1	temp.arg.explicit	Explicit template argument specification
14.8.2	temp.deduct	Template argument deduction
14.8.3	temp.over	Overload resolution
14.8.4	temp.over.spec	Overloading and template specializations
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
15.5.3	except.uncaught	The `uncaught_exception()` 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
17.3.3.8	lib.res.on.required	Required paragraph
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 `numeric_limits`
18.2.1.2	lib.numeric.limits.members	`numeric_limits` members
18.2.1.3	lib.round.style	Type `float_round_style`
18.2.1.4	lib.numeric.special	`numeric_limits` 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 `bad_alloc`
18.4.2.2	lib.new.handler	Type `new_handler`
18.4.2.3	lib.set.new.handler	`set_new_handler`
18.5	lib.support.rtti	Type identification
18.5.1	lib.type.info	Class `type_info`
18.5.2	lib.bad.cast	Class `bad_cast`
18.5.3	lib.bad.typeid	Class `bad_typeid`
18.6	lib.support.exception	Exception handling
18.6.1	lib.exception	Class `exception`
18.6.2	lib.exception.unexpected	Violating *`exception-specifications`*
18.6.2.1	lib.bad.exception	Class `bad_exception`
18.6.2.2	lib.unexpected.handler	Type `unexpected_handler`
18.6.2.3	lib.set.unexpected	`set_unexpected`
18.6.2.4	lib.unexpected	`unexpected`
18.6.3	lib.exception.terminate	Abnormal termination
18.6.3.1	lib.terminate.handler	Type `terminate_handler`
18.6.3.2	lib.set.terminate	`set_terminate`
18.6.3.3	lib.terminate	`terminate`

18.6.4	lib.uncaught	`uncaught_exception`
18.7	lib.support.runtime	Other runtime support
19	lib.diagnostics	Diagnostics library
19.1	lib.std.exceptions	Exception classes
19.1.1	lib.logic.error	Class `logic_error`
19.1.2	lib.domain.error	Class `domain_error`
19.1.3	lib.invalid.argument	Class `invalid_argument`
19.1.4	lib.length.error	Class `length_error`
19.1.5	lib.out.of.range	Class `out_of_range`
19.1.6	lib.runtime.error	Class `runtime_error`
19.1.7	lib.range.error	Class `range_error`
19.1.8	lib.overflow.error	Class `overflow_error`
19.1.9	lib.underflow.error	Class `underflow_error`
19.2	lib.assertions	Assertions
19.3	lib.errno	Error numbers
20	lib.utilities	General utilities library
20.1	lib.utility.requirements	Requirements
20.1.1	lib.equalitycomparable	Equality comparison
20.1.2	lib.lessthancomparable	Less than comparison
20.1.3	lib.copyconstructible	Copy construction
20.1.4	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
20.3.8	lib.member.pointer.adaptors	Adaptors for pointers to members
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.globals	`allocator` globals
20.4.1.3	lib.allocator.example	Example `allocator`
20.4.2	lib.storage.iterator	Raw storage iterator
20.4.3	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_n`
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.char.traits	Character traits

21.1.1	lib.char.traits.defs	Definitions
21.1.2	lib.char.traits.require	Character traits requirements
21.1.3	lib.char.traits.chartype	char_type
21.1.4	lib.char.traits.typedefs	traits typedefs
21.1.5	lib.char.traits.specializations	`char_traits` specializations
21.1.5.1	lib.char.traits.specializations.char	struct char_traits<char>
21.1.5.2	lib.char.traits.specializations.wchar.t	struct char_traits<wchar_t>
21.2	lib.string.classes	String classes
21.2.1	lib.template.string	Template class `basic_string`
21.2.1.1	lib.basic.string	Template class `basic_string`
21.2.1.2	lib.string.cons	`basic_string` constructors
21.2.1.3	lib.string.iterators	`basic_string` iterator support
21.2.1.4	lib.string.capacity	`basic_string` capacity
21.2.1.5	lib.string.access	`basic_string` element access
21.2.1.6	lib.string.modifiers	`basic_string` modifiers
21.2.1.6.1	lib.string::op+=	`basic_string::operator+=`
21.2.1.6.2	lib.string::append	`basic_string::append`
21.2.1.6.3	lib.string::assign	`basic_string::assign`
21.2.1.6.4	lib.string::insert	`basic_string::insert`
21.2.1.6.5	lib.string::erase	`basic_string::erase`
21.2.1.6.6	lib.string::replace	`basic_string::replace`
21.2.1.6.7	lib.string::copy	`basic_string::copy`
21.2.1.6.8	lib.string::swap	`basic_string::swap`
21.2.1.7	lib.string.ops	`basic_string` string operations
21.2.1.7.1	lib.string::find	`basic_string::find`
21.2.1.7.2	lib.string::rfind	`basic_string::rfind`
21.2.1.7.3	lib.string::find.first.of	`basic_string::find_first_of`
21.2.1.7.4	lib.string::find.last.of	`basic_string::find_last_of`
21.2.1.7.5	lib.string::find.first.not.of	`basic_string::find_first_not_of`
21.2.1.7.6	lib.string::find.last.not.of	`basic_string::find_last_not_of`
21.2.1.7.7	lib.string::substr	`basic_string::substr`
21.2.1.7.8	lib.string::compare	`basic_string::compare`
21.2.1.8	lib.string.nonmembers	`basic_string` non-member functions
21.2.1.8.1	lib.string::op+	`operator+`
21.2.1.8.2	lib.string::operator==	`operator==`
21.2.1.8.3	lib.string::op!=	`operator!=`
21.2.1.8.4	lib.string::op<	`operator<`
21.2.1.8.5	lib.string::op>	`operator>`
21.2.1.8.6	lib.string::op<=	`operator<=`
21.2.1.8.7	lib.string::op>=	`operator>=`
21.2.1.8.8	lib.string.special	`swap`
21.2.1.8.9	lib.string.io	Inserters and extractors
21.2.2	lib.string	Class `string_char_traits<char>`
21.2.3	lib.string.traits.members	`string_char_traits<char>` members
21.2.4	lib.wstring	Class `string_char_traits<wchar_t>`
21.2.5	lib.wstring.members	`string_char_traits<wchar_t>` members
21.3	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.global.templates	locale globals
22.1.3	lib.locale.convenience	Convenience interfaces
22.1.3.1	lib.classification	Character classification
22.1.3.2	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.statics	ctype<char> static members
22.2.1.3.4	lib.facet.ctype.char.virtuals	ctype<char> virtual functions
22.2.1.4	lib.locale.ctype.byname.special	Class ctype_byname<char>
22.2.1.5	lib.locale.codecvt	Template class codecvt
22.2.1.5.1	lib.locale.codecvt.members	codecvt members
22.2.1.5.2	lib.locale.codecvt.virtuals	codecvt virtual functions
22.2.1.6	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.moneypunct	Template class `moneypunct`
22.2.6.3.1	lib.locale.moneypunct.members	`moneypunct` members
22.2.6.3.2	lib.locale.moneypunct.virtuals	`moneypunct` virtual functions
22.2.6.4	lib.locale.moneypunct.byname	Template class `moneypunct_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
23.2.2.7	lib.deque.special	`deque` specialized algorithms
23.2.3	lib.list	Template class `list`
23.2.3.1	lib.list.types	`list` types
23.2.3.2	lib.list.cons	`list` constructors, copy, and assignment
23.2.3.3	lib.list.iterators	`list` iterator support
23.2.3.4	lib.list.capacity	`list` capacity
23.2.3.5	lib.list.access	`list` element access
23.2.3.6	lib.list.modifiers	`list` modifiers
23.2.3.7	lib.list.ops	`list` operations
23.2.3.8	lib.list.special	`list` specialized algorithms
23.2.4	lib.container.adapters	Container adapters
23.2.4.1	lib.queue	Template class `queue`
23.2.4.2	lib.priority.queue	Template class `priority_queue`
23.2.4.2.1	lib.priqueue.cons	`priority_queue` constructors
23.2.4.2.2	lib.priqueue.members	`priority_queue` members
23.2.4.3	lib.stack	Template class `stack`
23.2.5	lib.vector	Template class `vector`
23.2.5.1	lib.vector.types	`vector` types
23.2.5.2	lib.vector.cons	`vector` constructors, copy, and assignment

24.4.1.2.6	lib.reverse.bidir.iter.op--	`operator--`
24.4.1.2.7	lib.reverse.bidir.iter.op==	`operator==`
24.4.1.3	lib.reverse.iterator	Template class `reverse_iterator`
24.4.1.4	lib.reverse.iter.ops	`reverse_iterator` operations
24.4.1.4.1	lib.reverse.iter.cons	`reverse_iterator` constructor
24.4.1.4.2	lib.reverse.iter.conv	Conversion
24.4.1.4.3	lib.reverse.iter.op.star	`operator*`
24.4.1.4.4	lib.reverse.iter.opref	`operator->`
24.4.1.4.5	lib.reverse.iter.op++	`operator++`
24.4.1.4.6	lib.reverse.iter.op--	`operator--`
24.4.1.4.7	lib.reverse.iter.op+	`operator+`
24.4.1.4.8	lib.reverse.iter.op+=	`operator+=`
24.4.1.4.9	lib.reverse.iter.op-	`operator-`
24.4.1.4.10	lib.reverse.iter.op-=	`operator-=`
24.4.1.4.11	lib.reverse.iter.opindex	`operator[]`
24.4.1.4.12	lib.reverse.iter.op==	`operator==`
24.4.1.4.13	lib.reverse.iter.op<	`operator<`
24.4.1.4.14	lib.reverse.iter.opdiff	`operator-`
24.4.1.4.15	lib.reverse.iter.opsum	`operator+=`
24.4.2	lib.insert.iterators	Insert iterators
24.4.2.1	lib.back.insert.iterator	Template class `back_insert_iterator`
24.4.2.2	lib.back.insert.iter.ops	`back_insert_iterator` operations
24.4.2.2.1	lib.back.insert.iter.cons	`back_insert_iterator` constructor
24.4.2.2.2	lib.back.insert.iter.op=	`back_insert_iterator::operator=`
24.4.2.2.3	lib.back.insert.iter.op*	`back_insert_iterator::operator*`
24.4.2.2.4	lib.back.insert.iter.op++	`back_insert_iterator::operator++`
24.4.2.2.5	lib.back.inserter	`back_inserter`
24.4.2.3	lib.front.insert.iterator	Template class `front_insert_iterator`
24.4.2.3.1	lib.front.inserter.decl	`front_inserter`
24.4.2.4	lib.front.insert.iter.ops	`front_insert_iterator` operations
24.4.2.4.1	lib.front.insert.iter.cons	`front_insert_iterator` constructor
24.4.2.4.2	lib.front.insert.iter.op=	`front_insert_iterator::operator=`
24.4.2.4.3	lib.front.insert.iter.op*	`front_insert_iterator::operator*`
24.4.2.4.4	lib.front.insert.iter.op++	`front_insert_iterator::operator++`
24.4.2.4.5	lib.front.inserter	`front_inserter`
24.4.2.5	lib.insert.iterator	Template class `insert_iterator`
24.4.2.5.1	lib.insert.inserter	`insert_inserter`
24.4.2.6	lib.insert.iter.ops	`insert_iterator` operations
24.4.2.6.1	lib.insert.iter.cons	`insert_iterator` constructor
24.4.2.6.2	lib.insert.iter.op=	`insert_iterator::operator=`
24.4.2.6.3	lib.insert.iter.op*	`insert_iterator::operator*`
24.4.2.6.4	lib.insert.iter.op++	`insert_iterator::operator++`
24.4.2.6.5	lib.inserter	`inserter`
24.5	lib.stream.iterators	Stream iterators
24.5.1	lib.istream.iterator	Template class `istream_iterator`
24.5.2	lib.ostream.iterator	Template class `ostream_iterator`
24.5.3	lib.istreambuf.iterator	Template class `istreambuf_iterator`
24.5.3.1	lib.istreambuf.iterator::proxy	Template class `istreambuf_iterator::proxy`
24.5.3.2	lib.istreambuf.iterator.cons	`istreambuf_iterator` constructors
24.5.3.3	lib.istreambuf.iterator::op*	`istreambuf_iterator::operator*`
24.5.3.4	lib.istreambuf.iterator::op++	`istreambuf_iterator::operator++`
24.5.3.5	lib.istreambuf.iterator::equal	`istreambuf_iterator::equal`
24.5.3.6	lib.istreambuf.iterator::op==	`operator==`
24.5.3.7	lib.istreambuf.iterator::op!=	`operator!=`

24.5.4	lib.ostreambuf.iterator	Template class `ostreambuf_iterator`
24.5.4.1	lib.ostreambuf.iter.cons	`ostreambuf_iterator` constructors
24.5.4.2	lib.ostreambuf.iter.ops	`ostreambuf_iterator` 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 `complex`
26.2.2	lib.complex.special	`complex` specializations
26.2.3	lib.complex.members	`complex` member functions
26.2.4	lib.complex.member.ops	`complex` member operators
26.2.5	lib.complex.ops	`complex` non-member operations
26.2.6	lib.complex.value.ops	`complex` value operations
26.2.7	lib.complex.transcendentals	`complex` transcendentals
26.3	lib.numarray	Numeric arrays
26.3.1	lib.template.valarray	Template class `valarray`
26.3.1.1	lib.valarray.cons	`valarray` constructors
26.3.1.2	lib.valarray.assign	`valarray` assignment
26.3.1.3	lib.valarray.access	`valarray` element access
26.3.1.4	lib.valarray.sub	`valarray` subset operations
26.3.1.5	lib.valarray.unary	`valarray` unary operators
26.3.1.6	lib.valarray.cassign	`valarray` computed assignment
26.3.1.7	lib.valarray.members	`valarray` member functions
26.3.2	lib.valarray.nonmembers	`valarray` non-member operations
26.3.2.1	lib.valarray.binary	`valarray` binary operators
26.3.2.2	lib.valarray.comparison	`valarray` comparison operators
26.3.2.3	lib.valarray.min.max	`valarray` min and max functions
26.3.2.4	lib.valarray.transcend	`valarray` transcendentals
26.3.3	lib.class.slice	Class `slice`
26.3.3.1	lib.cons.slice	`slice` constructors
26.3.3.2	lib.slice.access	`slice` access functions
26.3.4	lib.template.slice.array	Template class `slice_array`
26.3.4.1	lib.cons.slice.arr	`slice_array` constructors
26.3.4.2	lib.slice.arr.assign	`slice_array` assignment
26.3.4.3	lib.slice.arr.comp.assign	`slice_array` computed assignment
26.3.4.4	lib.slice.arr.fill	`slice_array` fill function
26.3.5	lib.class.gslice	The `gslice` class
26.3.5.1	lib.gslice.cons	`gslice constructors`
26.3.5.2	lib.gslice.access	`gslice` access functions
26.3.6	lib.template.gslice.array	Template class `gslice_array`
26.3.6.1	lib.gslice.array.cons	`gslice_array` constructors
26.3.6.2	lib.gslice.array.assign	`gslice_array` assignment
26.3.6.3	lib.gslice.array.comp.assign	`gslice_array computed assignment`
26.3.6.4	lib.gslice.array.fill	`gslice_array` fill function
26.3.7	lib.template.mask.array	Template class `mask_array`
26.3.7.1	lib.mask.array.cons	`mask_array` constructors
26.3.7.2	lib.mask.array.assign	`mask_array` assignment
26.3.7.3	lib.mask.array.comp.assign	`mask_array` computed assignment
26.3.7.4	lib.mask.array.fill	`mask_array` fill function
26.3.8	lib.template.indirect.array	Template class `indirect_array`
26.3.8.1	lib.indirect.array.cons	`indirect_array` constructors
26.3.8.2	lib.indirect.array.assign	`indirect_array` assignment
26.3.8.3	lib.indirect.array.comp.assign	`indirect_array` computed assignment
26.3.8.4	lib.indirect.array.fill	`indirect_array` 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 *CHAR_T*
27.1.2.2	lib.iostreams.int.t	Type *INT_T*
27.1.2.3	lib.iostreams.off.t	Type *OFF_T*
27.1.2.4	lib.iostreams.pos.t	Type *POS_T*
27.1.2.5	lib.iostreams.sz.t	Type *SZ_T*
27.1.2.6	lib.iostreams.state.t	Type *STATE_T*
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 `ios_traits`
27.4.3	lib.ios.base	Class `ios_base`
27.4.3.1	lib.ios.types	Types
27.4.3.1.1	lib.ios::failure	Class `ios_base::failure`
27.4.3.1.2	lib.ios::fmtflags	Type `ios_base::fmtflags`
27.4.3.1.3	lib.ios::iostate	Type `ios_base::iostate`
27.4.3.1.4	lib.ios::openmode	Type `ios_base::openmode`
27.4.3.1.5	lib.ios::seekdir	Type `ios_base::seekdir`
27.4.3.1.6	lib.ios::Init	Class `ios_base::Init`
27.4.3.2	lib.fmtflags.state	`ios_base fmtflags` state functions
27.4.3.3	lib.ios.base.locales	`ios_base locale` functions
27.4.3.4	lib.ios.base.storage	`ios_base` storage functions
27.4.3.5	lib.ios.base.cons	`ios_base` constructors/destructors
27.4.4	lib.ios	Template class `basic_ios`
27.4.4.1	lib.basic.ios.cons	`basic_ios` constructors
27.4.4.2	lib.basic.ios.members	Member functions
27.4.4.3	lib.iostate.flags	`basic_ios iostate` flags functions
27.4.5	lib.std.ios.manip	`ios_base` manipulators
27.4.5.1	lib.fmtflags.manip	`fmtflags` manipulators
27.4.5.2	lib.adjustfield.manip	`adjustfield` manipulators
27.4.5.3	lib.basefield.manip	`basefield` manipulators
27.4.5.4	lib.floatfield.manip	`floatfield` manipulators
27.5	lib.stream.buffers	Stream buffers
27.5.1	lib.streambuf.reqts	Stream buffer requirements
27.5.2	lib.streambuf	Template class `basic_streambuf<charT,traits>`
27.5.2.1	lib.streambuf.cons	`basic_streambuf` constructors
27.5.2.2	lib.streambuf.members	`basic_streambuf` public member functions
27.5.2.2.1	lib.streambuf.locales	Locales
27.5.2.2.2	lib.streambuf.buffer	Buffer management and positioning
27.5.2.2.3	lib.streambuf.pub.get	Get area
27.5.2.2.4	lib.streambuf.pub.pback	Putback
27.5.2.2.5	lib.streambuf.pub.put	Put area
27.5.2.3	lib.streambuf.protected	`basic_streambuf` protected member functions
27.5.2.3.1	lib.streambuf.get.area	Get area access

27.5.2.3.2	lib.streambuf.put.area	Put area access
27.5.2.4	lib.streambuf.virtuals	basic_streambuf virtual functions
27.5.2.4.1	lib.streambuf.virt.locales	Locales
27.5.2.4.2	lib.streambuf.virt.buffer	Buffer management and positioning
27.5.2.4.3	lib.streambuf.virt.get	Get area
27.5.2.4.4	lib.streambuf.virt.pback	Putback
27.5.2.4.5	lib.streambuf.virt.put	Put area
27.6	lib.iostream.format	Formatting and manipulators
27.6.1	lib.input.streams	Input streams
27.6.1.1	lib.istream	Template class basic_istream
27.6.1.1.1	lib.istream.cons	basic_istream constructors
27.6.1.1.2	lib.istream::sentry	Class basic_istream::sentry
27.6.1.2	lib.istream.formatted	Formatted input functions
27.6.1.2.1	lib.istream.formatted.reqmts	Common requirements
27.6.1.2.2	lib.istream.formatted.arithmetic	Arithmetic Extractors
27.6.1.2.3	lib.istream::extractors	basic_istream::operator>>
27.6.1.3	lib.istream.unformatted	Unformatted input functions
27.6.1.4	lib.istream.manip	Standard basic_istream manipulators
27.6.1.4.1	lib.iostream.cons	basic_iostream constructors
27.6.2	lib.output.streams	Output streams
27.6.2.1	lib.ostream	Template class basic_ostream
27.6.2.2	lib.ostream.cons	basic_ostream constructors
27.6.2.3	lib.ostream::sentry	Class basic_ostream::sentry
27.6.2.4	lib.ostream.formatted	Formatted output functions
27.6.2.4.1	lib.ostream.formatted.reqmts	Common requirements
27.6.2.4.2	lib.ostream.insertors.arithmetic	Arithmetic Insertors
27.6.2.4.3	lib.ostream.inserters	basic_ostream::operator<<
27.6.2.5	lib.ostream.unformatted	Unformatted output functions
27.6.2.6	lib.ostream.manip	Standard basic_ostream manipulators
27.6.3	lib.std.manip	Standard manipulators
27.7	lib.string.streams	String-based streams
27.7.1	lib.stringbuf	Template class basic_stringbuf
27.7.1.1	lib.stringbuf.cons	basic_stringbuf constructors
27.7.1.2	lib.stringbuf.members	Member functions
27.7.1.3	lib.stringbuf.virtuals	Overridden virtual functions
27.7.2	lib.istringstream	Template class basic_istringstream
27.7.2.1	lib.istringstream.cons	basic_istringstream constructors
27.7.2.2	lib.istringstream.members	Member functions
27.7.2.3	lib.ostringstream	Class basic_ostringstream
27.7.2.4	lib.ostringstream.cons	basic_ostringstream constructors
27.7.2.5	lib.ostringstream.members	Member functions
27.7.3	lib.stringstream	Template class basic_stringstream
27.7.4	lib.stringstream.cons	basic_stringstream constructors
27.7.5	lib.stringstream.members	Member
27.8	lib.file.streams	File-based streams
27.8.1	lib.fstreams	File streams
27.8.1.1	lib.filebuf	Template class basic_filebuf
27.8.1.2	lib.filebuf.cons	basic_filebuf constructors
27.8.1.3	lib.filebuf.members	Member functions
27.8.1.4	lib.filebuf.virtuals	Overridden virtual functions
27.8.1.5	lib.ifstream	Template class basic_ifstream
27.8.1.6	lib.ifstream.cons	basic_ifstream constructors
27.8.1.7	lib.ifstream.members	Member functions

D.2	depr.static	static keyword
D.3	depr.access.dcl	Access declarations
D.4	depr.c.headers	Standard C library headers
D.5	depr.ios.members	Old iostreams members
D.6	depr.str.strstreams	`char*` streams
D.6.1	depr.strstreambuf	Class `strstreambuf`
D.6.1.1	depr.strstreambuf.cons	`strstreambuf` constructors
D.6.1.2	depr.strstreambuf.members	Member functions
D.6.1.3	depr.strstreambuf.virtuals	`strstreambuf` overridden virtual functions
D.6.2	depr.istrstream	Class `istrstream`
D.6.2.1	depr.istrstream.cons	`istrstream` constructors
D.6.2.2	depr.istrstream.members	Member functions
D.6.3	depr.ostrstream	Class `ostrstream`
D.6.3.1	depr.ostrstream.cons	`ostrstream` constructors
D.6.3.2	depr.ostrstream.members	Member functions
D.6.4	depr.strstream	Class `strstream`
D.6.4.1	depr.strstream.cons	`strstream` constructors
E	extendid	Universal-character-names for Identifiers

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.lookup	3.4	Name look up
basic.lookup.classref	3.4.4	Class member access
basic.lookup.elab	3.4.3	Elaborated type specifiers
basic.lookup.qual	3.4.2	Qualified name look up
basic.lookup.udir	3.4.5	Using directives and namespace aliases
basic.lookup.unqual	3.4.1	Unqualified name look up
basic.lval	3.10	Lvalues and rvalues
basic.namespace	7.3	Namespaces
basic.scope	3.3	Declarative regions and scopes
basic.scope.class	3.3.6	Class scope
basic.scope.hiding	3.3.7	Name hiding
basic.scope.local	3.3.2	Local scope
basic.scope.namespace	3.3.5	Namespace scope
basic.scope.pdecl	3.3.1	Point of declaration
basic.scope.proto	3.3.3	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.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	Accessibility of base classes and base class members
class.access.dcl	11.3	Access declarations
class.access.nest	11.8	Nested classes
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	9.6	Bit-fields
class.cdtor	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	9.8	Local class declarations
class.mem	9.2	Class members
class.member.lookup	10.2	Member name lookup
class.mfct	9.3	Member functions
class.mfct.nonstatic	9.3.1	Nonstatic member functions
class.mi	10.1	Multiple base classes
class.name	9.1	Class names
class.nest	9.7	Nested class declarations
class.nested.type	9.9	Nested type names
class.paths	11.7	Multiple access
class.protected	11.5	Protected member access
class.qual	3.4.2.1	Class members
class.static	9.4	Static members
class.static.data	9.4.2	Static data members
class.static.mfct	9.4.1	Static member functions
class.temporary	12.2	Temporary objects
class.this	9.3.2	The `this` pointer
class.union	9.5	Unions
class.virtual	10.3	Virtual functions
conv	4	Standard conversions
conv.array	4.2	Array-to-pointer conversion
conv.bool	4.12	Boolean conversions
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 ## 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 # 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 `asm` 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 `friend` 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 *cv-qualifiers*
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 `typedef` specifier
depr	D	Compatibility features
depr.access.dcl	D.3	Access declarations
depr.c.headers	D.4	Standard C library headers
depr.ios.members	D.5	Old iostreams members
depr.istrstream	D.6.2	Class `istrstream`
depr.istrstream.cons	D.6.2.1	`istrstream` constructors
depr.istrstream.members	D.6.2.2	Member functions
depr.ostrstream	D.6.3	Class `ostrstream`
depr.ostrstream.cons	D.6.3.1	`ostrstream` constructors
depr.ostrstream.members	D.6.3.2	Member functions
depr.post.incr	D.1	Postfix increment operator
depr.static	D.2	static keyword
depr.str.strstreams	D.6	`char*` streams
depr.strstream	D.6.4	Class `strstream`
depr.strstream.cons	D.6.4.1	`strstream` constructors
depr.strstreambuf	D.6.1	Class `strstreambuf`
depr.strstreambuf.cons	D.6.1.1	`strstreambuf` constructors
depr.strstreambuf.members	D.6.1.2	Member functions
depr.strstreambuf.virtuals	D.6.1.3	`strstreambuf` 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 _basic_: 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 _class_: classes
diff.class.nonnested	C.3.5	Nonnested classes
diff.cpp	C.2.9	Clause _cpp_: preprocessing directives
diff.dcl	C.2.5	Clause _dcl.dcl_: declarations

diff.decl	C.2.6	Clause _dcl.decl_: declarators
diff.early	C.1.1	C++ features available in 1985
diff.expr	C.2.3	Clause _expr_: expressions
diff.fct.def	C.3.1	Old style function definitions
diff.header.iso646.h	C.4.2.2	Header `<iso646.h>`
diff.iso	C.2	C++ and ISO C
diff.lex	C.2.1	Clause _lex_: 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 `NULL`
diff.offsetof	C.4.4.1	Macro `offsetof(`*type*, *member-designator*`)`
diff.special	C.2.8	Clause _special_: special member functions
diff.stat	C.2.4	Clause _stmt.stmt_: statements
diff.this	C.3.3	Assignment to `this`
diff.wchar.t	C.4.2.1	Type `wchar_t`
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 `terminate()` function
except.throw	15.1	Throwing an exception
except.uncaught	15.5.3	The `uncaught_exception()` function
except.unexpected	15.5.2	The `unexpected()` 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	5.2.11	Const cast
expr.delete	5.3.5	Delete
expr.dynamic.cast	5.2.7	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	5.2.6	Increment and decrement
expr.pre.incr	5.3.2	Increment and decrement
expr.prim	5.1	Primary expressions
expr.pseudo	5.2.4	Pseudo destructor call
expr.ref	5.2.5	Class member access

expr.reinterpret.cast	5.2.10	Reinterpret cast
expr.rel	5.9	Relational operators
expr.shift	5.8	Shift operators
expr.sizeof	5.3.3	Sizeof
expr.static.cast	5.2.9	Static cast
expr.sub	5.2.1	Subscripting
expr.type.conv	5.2.3	Explicit type conversion (functional notation)
expr.typeid	5.2.8	Type identification
expr.unary	5.3	Unary expressions
expr.unary.op	5.3.1	Unary operators
expr.xor	5.12	Bitwise exclusive OR operator
extendid	E	Universal-character-names for Identifiers
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	1.3	Implementation compliance
intro.defs	1.4	Definitions
intro.execution	1.8	Program execution
intro.memory	1.6	The C++ memory model
intro.object	1.7	The C++ object model
intro.refs	1.2	Normative references
intro.scope	1.1	Scope
lex	2	Lexical conventions
lex.bool	2.13.5	Boolean literals
lex.ccon	2.13.2	Character literals
lex.charset	2.2	Basic source character set
lex.comment	2.7	Comments
lex.digraph	2.5	Alternative tokens
lex.fcon	2.13.3	Floating literals
lex.header	2.8	Header names
lex.icon	2.13.1	Integer literals
lex.key	2.11	Keywords
lex.literal	2.13	Literals
lex.name	2.10	Identifiers
lex.phases	2.1	Phases of translation
lex.ppnumber	2.9	Preprocessing numbers
lex.pptoken	2.4	Preprocessing tokens
lex.string	2.13.4	String literals
lex.token	2.6	Tokens
lex.trigraph	2.3	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.allocator.example	20.4.1.3	Example `allocator`
lib.allocator.globals	20.4.1.2	`allocator` globals
lib.allocator.members	20.4.1.1	`allocator` members
lib.allocator.requirements	20.1.4	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	24.4.2.2.1	`back_insert_iterator` constructor
lib.back.insert.iter.op*	24.4.2.2.3	`back_insert_iterator::operator*`
lib.back.insert.iter.op++	24.4.2.2.4	`back_insert_iterator::operator++`
lib.back.insert.iter.op=	24.4.2.2.2	`back_insert_iterator::operator=`
lib.back.insert.iter.ops	24.4.2.2	`back_insert_iterator` operations
lib.back.insert.iterator	24.4.2.1	Template class `back_insert_iterator`
lib.back.inserter	24.4.2.2.5	`back_inserter`

lib.bad.alloc	18.4.2.1	Class `bad_alloc`
lib.bad.cast	18.5.2	Class `bad_cast`
lib.bad.exception	18.6.2.1	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.iterators	24.3.2	Basic iterators
lib.basic.string	21.2.1.1	Template class `basic_string`
lib.bidirectional.iterators	24.1.4	Bidirectional iterators
lib.binary.search	25.3.3.4	`binary_search`
lib.bind.1st	20.3.6.2	`bind1st`
lib.bind.2nd	20.3.6.4	`bind2nd`
lib.binder.1st	20.3.6.1	Template class `binder1st`
lib.binder.2nd	20.3.6.3	Template class `binder2nd`
lib.binders	20.3.6	Binders
lib.bitmask.types	17.2.2.1.2	Bitmask types
lib.bitset.cons	23.2.1.1	`bitset` constructors
lib.bitset.members	23.2.1.2	`bitset` members
lib.bitset.operators	23.2.1.3	`bitset` operators
lib.byte.strings	17.2.2.1.3.1	Byte strings
lib.c.files	27.8.2	C Library files
lib.c.limits	18.2.2	C Library
lib.c.locales	22.3	C Library Locales
lib.c.malloc	20.4.6	C Library
lib.c.math	26.5	C Library
lib.c.strings	21.3	Null-terminated sequence utilities
lib.category.collate	22.2.4	The collate category
lib.category.ctype	22.2.1	The `ctype` category
lib.category.messages	22.2.7	The message retrieval category
lib.category.monetary	22.2.6	The monetary category
lib.category.numeric	22.2.2	The numeric category
lib.category.time	22.2.5	The time category
lib.char.traits	21.1	Character traits
lib.char.traits.chartype	21.1.3	char_type
lib.char.traits.defs	21.1.1	Definitions
lib.char.traits.require	21.1.2	Character traits requirements
lib.char.traits.specializations	21.1.5	`char_traits` specializations
lib.char.traits.specializations.char	21.1.5.1	struct char_traits<char>
lib.char.traits.specializations.wchar.t	21.1.5.2	struct char_traits<wchar_t>
lib.char.traits.typedefs	21.1.4	traits typedefs
lib.character.seq	17.2.2.1.3	Character sequences
lib.class.gslice	26.3.5	The `gslice` class
lib.class.slice	26.3.3	Class `slice`
lib.classification	22.1.3.1	Character classification
lib.comparisons	20.3.3	Comparisons
lib.complex	26.2.1	Template class `complex`
lib.complex.member.ops	26.2.4	`complex` member operators
lib.complex.members	26.2.3	`complex` member functions
lib.complex.numbers	26.2	Complex numbers
lib.complex.ops	26.2.5	`complex` non-member operations
lib.complex.special	26.2.2	`complex` specializations

lib.complex.transcendentals	26.2.7	`complex` transcendentals
lib.complex.value.ops	26.2.6	`complex` value operations
lib.compliance	17.3.1.3	Freestanding implementations
lib.conforming	17.3.4	Conforming implementations
lib.cons.slice	26.3.3.1	`slice` constructors
lib.cons.slice.arr	26.3.4.1	`slice_array` constructors
lib.constraints	17.3.3	Constraints on programs
lib.container.adapters	23.2.4	Container adapters
lib.container.requirements	23.1	Container requirements
lib.containers	23	Containers library
lib.contents	17.3.1.1	Library contents
lib.conventions	17.2.2	Other conventions
lib.conversions	22.1.3.2	Character conversions
lib.copyconstructible	20.1.3	Copy construction
lib.date.time	20.5	Date and time
lib.default.allocator	20.4.1	The default allocator
lib.definitions	17.1	Definitions
lib.deque	23.2.2	Template class `deque`
lib.deque.access	23.2.2.5	`deque` element access
lib.deque.capacity	23.2.2.4	`deque` capacity
lib.deque.cons	23.2.2.2	`deque` 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.special	23.2.2.7	`deque` specialized algorithms
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.diagnostics	19	Diagnostics library
lib.domain.error	19.1.2	Class `domain_error`
lib.enumerated.types	17.2.2.1.1	Enumerated types
lib.equal.range	25.3.3.3	`equal_range`
lib.equalitycomparable	20.1.1	Equality comparison
lib.errno	19.3	Error numbers
lib.exception	18.6.1	Class `exception`
lib.exception.terminate	18.6.3	Abnormal termination
lib.exception.unexpected	18.6.2	Violating *`exception-specifications`*
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.statics	22.2.1.3.3	`ctype<char>` static members
lib.facet.ctype.char.virtuals	22.2.1.3.4	`ctype<char>` 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	24.4.2.4.1	`front_insert_iterator` constructor
lib.front.insert.iter.op*	24.4.2.4.3	`front_insert_iterator::operator*`
lib.front.insert.iter.op++	24.4.2.4.4	`front_insert_iterator::operator++`
lib.front.insert.iter.op=	24.4.2.4.2	`front_insert_iterator::operator=`
lib.front.insert.iter.ops	24.4.2.4	`front_insert_iterator` operations
lib.front.insert.iterator	24.4.2.3	Template class `front_insert_iterator`
lib.front.inserter	24.4.2.4.5	`front_inserter`
lib.front.inserter.decl	24.4.2.3.1	`front_inserter`
lib.fstream	27.8.1.11	Template class `basic_fstream`
lib.fstream.cons	27.8.1.12	`basic_fstream` constructors
lib.fstream.members	27.8.1.13	Member functions
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.inserter	24.4.2.5.1	`insert_inserter`
lib.insert.iter.cons	24.4.2.6.1	`insert_iterator` constructor
lib.insert.iter.op*	24.4.2.6.3	`insert_iterator::operator*`
lib.insert.iter.op++	24.4.2.6.4	`insert_iterator::operator++`
lib.insert.iter.op=	24.4.2.6.2	`insert_iterator::operator=`
lib.insert.iter.ops	24.4.2.6	`insert_iterator` operations
lib.insert.iterator	24.4.2.5	Template class `insert_iterator`
lib.insert.iterators	24.4.2	Insert iterators
lib.inserter	24.4.2.6.5	`inserter`
lib.invalid.argument	19.1.3	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 / destructors
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.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.cons	27.6.1.4.1	`basic_iostream` constructors
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.state.t	27.1.2.6	Type *STATE_T*
lib.iostreams.sz.t	27.1.2.5	Type *SZ_T*
lib.iostreams.type.reqmts	27.1.2	Type requirements
lib.istream	27.6.1.1	Template class `basic_istream`
lib.istream.cons	27.6.1.1.1	`basic_istream` constructors
lib.istream.formatted	27.6.1.2	Formatted input functions
lib.istream.formatted.arithmetic	27.6.1.2.2	Arithmetic Extractors
lib.istream.formatted.reqmts	27.6.1.2.1	Common requirements
lib.istream.iterator	24.5.1	Template class `istream_iterator`
lib.istream.manip	27.6.1.4	Standard `basic_istream` manipulators
lib.istream.unformatted	27.6.1.3	Unformatted input functions
lib.istream::extractors	27.6.1.2.3	`basic_istream::operator>>`
lib.istream::sentry	27.6.1.1.2	Class `basic_istream::sentry`
lib.istreambuf.iterator	24.5.3	Template class `istreambuf_iterator`
lib.istreambuf.iterator.cons	24.5.3.2	`istreambuf_iterator` constructors
lib.istreambuf.iterator::equal	24.5.3.5	`istreambuf_iterator::equal`
lib.istreambuf.iterator::op!=	24.5.3.7	`operator!=`
lib.istreambuf.iterator::op*	24.5.3.3	`istreambuf_iterator::operator*`
lib.istreambuf.iterator::op++	24.5.3.4	`istreambuf_iterator::operator++`
lib.istreambuf.iterator::op==	24.5.3.6	`operator==`
lib.istreambuf.iterator::proxy	24.5.3.1	Template class `istreambuf_iterator::proxy`
lib.istringstream	27.7.2	Template class `basic_istringstream`
lib.istringstream.cons	27.7.2.1	`basic_istringstream` constructors
lib.istringstream.members	27.7.2.2	Member functions
lib.iterator.operations	24.3.3	Iterator operations
lib.iterator.primitives	24.3	Iterator primitives
lib.iterator.requirements	24.1	Iterator requirements
lib.iterator.synopsis	24.2	Header `<iterator>` synopsis
lib.iterator.tags	24.1.6	Iterator tags
lib.iterators	24	Iterators library

lib.language.support	18	Language support library
lib.length.error	19.1.4	Class `length_error`
lib.lessthancomparable	20.1.2	Less than comparison
lib.library	17	Library introduction
lib.limits	18.2.1	Numeric limits
lib.list	23.2.3	Template class `list`
lib.list.access	23.2.3.5	`list` element access
lib.list.capacity	23.2.3.4	`list` capacity
lib.list.cons	23.2.3.2	`list` constructors, copy, and assignment
lib.list.iterators	23.2.3.3	`list` iterator support
lib.list.modifiers	23.2.3.6	`list` modifiers
lib.list.ops	23.2.3.7	`list` operations
lib.list.special	23.2.3.8	`list` specialized algorithms
lib.list.types	23.2.3.1	`list` types
lib.locale	22.1.1	Class `locale`
lib.locale.categories	22.2	Standard `locale` categories
lib.locale.category	22.1.1.1.1	Type `locale::category`
lib.locale.codecvt	22.2.1.5	Template class `codecvt`
lib.locale.codecvt.byname	22.2.1.6	Template class `codecvt_byname`
lib.locale.codecvt.members	22.2.1.5.1	`codecvt` members
lib.locale.codecvt.virtuals	22.2.1.5.2	`codecvt` virtual functions
lib.locale.collate	22.2.4.1	Template class `collate`
lib.locale.collate.byname	22.2.4.2	Template class `collate_byname`
lib.locale.collate.members	22.2.4.1.1	`collate` members
lib.locale.collate.virtuals	22.2.4.1.2	`collate` virtual functions
lib.locale.cons	22.1.1.2	`locale` constructors and destructor
lib.locale.convenience	22.1.3	Convenience interfaces
lib.locale.ctype	22.2.1.1	Template class `ctype`
lib.locale.ctype.byname	22.2.1.2	Template class `ctype_byname`
lib.locale.ctype.byname.special	22.2.1.4	Class ctype_byname<char>
lib.locale.ctype.members	22.2.1.1.1	`ctype` 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.global.templates	22.1.2	`locale` globals
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.moneypunct	22.2.6.3	Template class `moneypunct`
lib.locale.moneypunct.byname	22.2.6.4	Template class `moneypunct_byname`
lib.locale.moneypunct.members	22.2.6.3.1	`moneypunct` members
lib.locale.moneypunct.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	19.1.1	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.special	23.3.1.9	`map` specialized algorithms
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.member.pointer.adaptors	20.3.8	Adaptors for pointers to members
lib.memory	20.4	Memory
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.multimap.special	23.3.2.1	`multimap` specialized algorithms
lib.multiset	23.3.4	Template class `multiset`
lib.multiset.special	23.3.4.1	`multiset` specialized algorithms
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	27.6.2.4.3	basic_ostream::operator<<
lib.ostream.insertors.arithmetic	27.6.2.4.2	Arithmetic Insertors
lib.ostream.iterator	24.5.2	Template class ostream_iterator
lib.ostream.manip	27.6.2.6	Standard basic_ostream manipulators
lib.ostream.unformatted	27.6.2.5	Unformatted output functions
lib.ostream::sentry	27.6.2.3	Class basic_ostream::sentry
lib.ostreambuf.iter.cons	24.5.4.1	ostreambuf_iterator constructors
lib.ostreambuf.iter.ops	24.5.4.2	ostreambuf_iterator operations
lib.ostreambuf.iterator	24.5.4	Template class ostreambuf_iterator
lib.ostringstream	27.7.2.3	Class basic_ostringstream
lib.ostringstream.cons	27.7.2.4	basic_ostringstream constructors
lib.ostringstream.members	27.7.2.5	Member functions
lib.out.of.range	19.1.5	Class out_of_range
lib.output.iterators	24.1.2	Output iterators
lib.output.streams	27.6.2	Output streams
lib.overflow.error	19.1.8	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	24.4	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	19.1.7	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.res.on.required	17.3.3.8	Required paragraph

lib.reserved.names	17.3.3.1	Reserved names
lib.reverse.bidir.iter	24.4.1.1	Template class reverse_bidirectional_iterator
lib.reverse.bidir.iter.cons	24.4.1.2.1	reverse_bidirectional_iterator constructor
lib.reverse.bidir.iter.conv	24.4.1.2.2	Conversion
lib.reverse.bidir.iter.op++	24.4.1.2.5	operator++
lib.reverse.bidir.iter.op--	24.4.1.2.6	operator--
lib.reverse.bidir.iter.op.star	24.4.1.2.3	operator*
lib.reverse.bidir.iter.op==	24.4.1.2.7	operator==
lib.reverse.bidir.iter.opref	24.4.1.2.4	operator->
lib.reverse.bidir.iter.ops	24.4.1.2	reverse_bidirectional_iterator operations
lib.reverse.iter.cons	24.4.1.4.1	reverse_iterator constructor
lib.reverse.iter.conv	24.4.1.4.2	Conversion
lib.reverse.iter.op+	24.4.1.4.7	operator+
lib.reverse.iter.op++	24.4.1.4.5	operator++
lib.reverse.iter.op+=	24.4.1.4.8	operator+=
lib.reverse.iter.op-	24.4.1.4.9	operator-
lib.reverse.iter.op--	24.4.1.4.6	operator--
lib.reverse.iter.op-=	24.4.1.4.10	operator-=
lib.reverse.iter.op.star	24.4.1.4.3	operator*
lib.reverse.iter.op<	24.4.1.4.13	operator<
lib.reverse.iter.op==	24.4.1.4.12	operator==
lib.reverse.iter.opdiff	24.4.1.4.14	operator-
lib.reverse.iter.opindex	24.4.1.4.11	operator[]
lib.reverse.iter.opref	24.4.1.4.4	operator->
lib.reverse.iter.ops	24.4.1.4	reverse_iterator operations
lib.reverse.iter.opsum	24.4.1.4.15	operator+=
lib.reverse.iterator	24.4.1.3	Template class reverse_iterator
lib.reverse.iterators	24.4.1	Reverse iterators
lib.round.style	18.2.1.3	Type float_round_style
lib.runtime.error	19.1.6	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.special	23.3.3.8	set specialized algorithms
lib.set.symmetric.difference	25.3.5.5	set_symmetric_difference
lib.set.terminate	18.6.3.2	set_terminate
lib.set.types	23.3.3.1	set types
lib.set.unexpected	18.6.2.3	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	24.3.1	Standard iterator tags
lib.std.manip	27.6.3	Standard manipulators
lib.storage.iterator	20.4.2	Raw storage iterator
lib.stream.buffers	27.5	Stream buffers
lib.stream.iterators	24.5	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	21.2.2	Class string_char_traits<char>
lib.string.access	21.2.1.5	basic_string element access
lib.string.capacity	21.2.1.4	basic_string capacity
lib.string.classes	21.2	String classes
lib.string.cons	21.2.1.2	basic_string constructors
lib.string.io	21.2.1.8.9	Inserters and extractors
lib.string.iterators	21.2.1.3	basic_string iterator support
lib.string.modifiers	21.2.1.6	basic_string modifiers
lib.string.nonmembers	21.2.1.8	basic_string non-member functions
lib.string.ops	21.2.1.7	basic_string string operations
lib.string.special	21.2.1.8.8	swap
lib.string.streams	27.7	String-based streams
lib.string.traits.members	21.2.3	string_char_traits<char> members
lib.string::append	21.2.1.6.2	basic_string::append
lib.string::assign	21.2.1.6.3	basic_string::assign
lib.string::compare	21.2.1.7.8	basic_string::compare
lib.string::copy	21.2.1.6.7	basic_string::copy
lib.string::erase	21.2.1.6.5	basic_string::erase
lib.string::find	21.2.1.7.1	basic_string::find
lib.string::find.first.not.of	21.2.1.7.5	basic_string::find_first_not_of
lib.string::find.first.of	21.2.1.7.3	basic_string::find_first_of
lib.string::find.last.not.of	21.2.1.7.6	basic_string::find_last_not_of

lib.string::find.last.of	21.2.1.7.4	basic_string::find_last_of
lib.string::insert	21.2.1.6.4	basic_string::insert
lib.string::op!=	21.2.1.8.3	operator!=
lib.string::op+	21.2.1.8.1	operator+
lib.string::op+=	21.2.1.6.1	basic_string::operator+=
lib.string::op<	21.2.1.8.4	operator<
lib.string::op<=	21.2.1.8.6	operator<=
lib.string::op>	21.2.1.8.5	operator>
lib.string::op>=	21.2.1.8.7	operator>=
lib.string::operator==	21.2.1.8.2	operator==
lib.string::replace	21.2.1.6.6	basic_string::replace
lib.string::rfind	21.2.1.7.2	basic_string::rfind
lib.string::substr	21.2.1.7.7	basic_string::substr
lib.string::swap	21.2.1.6.8	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.stringstream	27.7.3	Template class basic_stringstream
lib.stringstream.cons	27.7.4	basic_stringstream constructors
lib.stringstream.members	27.7.5	Member
lib.structure	17.2.1	Structure of each subclause
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.gslice.array	26.3.6	Template class gslice_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	21.2.1	Template class basic_string
lib.template.valarray	26.3.1	Template class valarray
lib.temporary.buffer	20.4.3	Temporary buffers
lib.terminate	18.6.3.3	terminate
lib.terminate.handler	18.6.3.1	Type terminate_handler
lib.type.descriptions	17.2.2.1	Type descriptions
lib.type.info	18.5.1	Class type_info
lib.uncaught	18.6.4	uncaught_exception
lib.underflow.error	19.1.9	Class underflow_error
lib.unexpected	18.6.2.4	unexpected
lib.unexpected.handler	18.6.2.2	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_n
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.utility.requirements	20.1	Requirements
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.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.special	23.2.5.7	`vector` specialized algorithms
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	21.2.4	Class `string_char_traits<wchar_t>`
lib.wstring.members	21.2.5	`string_char_traits<wchar_t>` members
limits	B	Implementation quantities
namespace.alias	7.3.2	Namespace alias
namespace.def	7.3.1	Namespace definition
namespace.memdef	7.3.1.2	Namespace member definitions
namespace.qual	3.4.2.2	Namespace members
namespace.udecl	7.3.3	The `using` declaration
namespace.udir	7.3.4	Using directive
namespace.unnamed	7.3.1.1	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.conv	13.3.1.5	Initialization by conversion function
over.match.copy	13.3.1.4	Copy-initialization of class by user-defined conversion
over.match.ctor	13.3.1.3	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.ref	13.3.1.6	Initialization by conversion function for direct reference binding
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 `break` statement
stmt.cont	6.6.2	The `continue` statement
stmt.dcl	6.7	Declaration statement
stmt.do	6.5.2	The `do` statement
stmt.expr	6.2	Expression statement
stmt.for	6.5.3	The `for` statement
stmt.goto	6.6.4	The `goto` statement
stmt.if	6.4.1	The `if` 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 `return` statement
stmt.select	6.4	Selection statements
stmt.stmt	6	Statements
stmt.switch	6.4.2	The `switch` statement
stmt.while	6.5.1	The `while` statement
syntax	1.5	Syntax notation
temp	14	Templates
temp.arg	14.3	Template arguments
temp.arg.explicit	14.8.1	Explicit template argument specification
temp.class	14.5.1	Class templates
temp.class.order	14.5.4.2	Partial ordering of class template specializations
temp.class.spec	14.5.4	Class template partial specializations
temp.class.spec.match	14.5.4.1	Matching of class template partial specializations
temp.class.spec.mfunc	14.5.4.3	Members of class template specializations
temp.decls	14.5	Template declarations
temp.deduct	14.8.2	Template argument deduction
temp.dep	14.6.2	Dependent names
temp.expl.spec	14.7.3	Explicit specialization
temp.explicit	14.7.2	Explicit instantiation
temp.fct	14.5.5	Function templates
temp.fct.spec	14.8	Function template specializations
temp.friend	14.5.3	Friends
temp.func.order	14.5.5.2	Partial ordering of function templates
temp.inject	14.6.5	Friend names declared within a class template

temp.inst	14.7.1	Implicit instantiation
temp.local	14.6.1	Locally declared names
temp.mem	14.5.2	Member templates
temp.mem.class	14.5.1.2	Member classes of class templates
temp.mem.func	14.5.1.1	Member functions of class templates
temp.names	14.2	Names of template specializations
temp.nondep	14.6.3	Non-dependent names
temp.over	14.8.3	Overload resolution
temp.over.link	14.5.5.1	Function template overloading
temp.over.spec	14.8.4	Overloading and template specializations
temp.param	14.1	Template parameters
temp.point	14.6.4	Point of instantiation
temp.res	14.6	Name resolution
temp.spec	14.7	Template specialization
temp.static	14.5.1.3	Static data members of class templates
temp.type	14.4	Type equivalence