Accredited Standards Committee*
X3, INFORMATION PROCESSING SYSTEMS

Doc No: X3J16 / 94– 0029
WG21 / N0416
Date: 25 January 1994
Project: Programming Language C++

Reply to: Andrew Koenig
AT&T Bell Laboratories
PO Box 636
600 Mountain Avenue
Room 6D– 416B
Murray Hill, NJ 07974 USA
ark@research.att.com

Concordance for November 1993 (San Jose) 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	Introduction
1.1	overview	Overview
1.2	syntax	Syntax notation
1.3	intro.memory	The C++ memory model
1.4	intro.defs	Definitions of terms
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	Digraph sequences
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
3	basic	Basic concepts
3.1	basic.def	Declarations and definitions
3.2	basic.scope	Scopes
3.3	basic.link	Program and linkage
3.4	basic.start	Start and termination
3.5	basic.stc	Storage classes
3.6	basic.types	Types
3.6.1	basic.fundamental	Fundamental types
3.6.2	basic.derived	Derived types
3.6.3	basic.type.qualifier	CV-qualifiers
3.6.4	basic.type.name	Type names
3.7	basic.lval	Lvalues
4	conv	Standard conversions
4.1	conv.prom	Integral promotions
4.2	conv.integral	Integral conversions
4.3	conv.double	Float and double
4.4	conv.float	Floating and integral
4.5	conv.arith	Arithmetic conversions
4.6	conv.ptr	Pointer conversions
4.7	conv.ref	Reference conversions
4.8	conv.mem	Pointers to members
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
5.3	expr.unary	Unary operators
5.3.1	expr.pre.incr	Increment and decrement
5.3.2	expr.sizeof	Sizeof

5.3.3	expr.new	New
5.3.4	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.template	The `template` specifier
7.1.5	dcl.friend	The `friend` specifier
7.1.6	dcl.type	Type specifiers
7.2	dcl.enum	Enumeration declarations
7.3	dcl.asm	Asm declarations
7.4	dcl.link	Linkage specifications
8	dcl.decl	Declarators
8.1	dcl.name	Type names
8.1.1	dcl.ambig.res	Ambiguity resolution
8.2	dcl.meaning	Meaning of declarators
8.2.1	dcl.ptr	Pointers
8.2.2	dcl.ref	References
8.2.3	dcl.mptr	Pointers to members
8.2.4	dcl.array	Arrays

8.2.5	dcl.fct	Functions
8.2.6	dcl.fct.default	Default parameters
8.3	dcl.fct.def	Function definitions
8.4	dcl.init	Initializers
8.4.1	dcl.init.aggr	Aggregates
8.4.2	dcl.init.string	Character arrays
8.4.3	dcl.init.ref	References
9	class	Classes
9.1	class.name	Class names
9.2	class.mem	Class members
9.2.1	class.scope0	Scope rules for classes
9.3	class.mfct	Member functions
9.3.1	class.this	The `this` pointer
9.3.2	class.inline	Inline member functions
9.4	class.static	Static members
9.5	class.union	Unions
9.6	class.bit	Bit-fields
9.7	class.nest	Nested class declarations
9.8	class.local	Local class declarations
9.9	class.nested.type	Nested type names
10	class.derived	Derived classes
10.1	class.mi	Multiple base classes
10.1.1	class.ambig	Ambiguities
10.2	class.virtual	Virtual functions
10.3	class.abstract	Abstract classes
10.4	class.scope	Summary of scope rules
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.cdtor	Constructors and destructors
12.8	class.copy	Copying class objects
13	over	Overloading
13.1	over.dcl	Declaration matching
13.2	over.match	Argument matching
13.3	over.over	Address of overloaded function
13.4	over.oper	Overloaded operators
13.4.1	over.unary	Unary operators
13.4.2	over.binary	Binary operators
13.4.3	over.ass	Assignment

13.4.4	over.call	Function call
13.4.5	over.sub	Subscripting
13.4.6	over.ref	Class member access
13.4.7	over.inc	Increment and decrement
14	temp	Templates
15	except	Exception handling
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
17.1	lib.sup	Language support
17.1.1	lib.free	Free store `<new>`
17.1.1.1	lib.opnew	`operator new()` and `operator new[]()`
17.1.1.2	lib.opdelete	`operator delete()` and `operator delete[]()`
17.1.1.3	lib.placeopnew	*placement* `operator new()`
17.1.1.4	lib.newhndl	*new-handler* function
17.1.1.5	lib.setnewhndl	`set_new_handler()`
17.1.2	lib.typeinfo	Type identification `<type_info>`
17.1.3	lib.exception	Exceptions
17.1.3.1	lib.excptterm	Abnormal termination
17.1.3.1.1	lib.terminate	`terminate()`
17.1.3.1.2	lib.termfn	*terminate-function*
17.1.3.1.3	lib.setterm	`set_terminate()`
17.1.3.2	lib.violspec	Violating *exception-specification*s
17.1.3.2.1	lib.unexpected	`unexpected()`
17.1.3.2.2	lib.unexpfn	*unexpected-function*
17.1.3.2.3	lib.setunexp	`set_unexpected()`
17.1.3.3	lib.predexcpt	Predefined exceptions
17.1.3.3.1	lib.xmsg	`xmsg` exception
17.1.3.3.2	lib.xalloc	`xalloc` exception
17.1.3.3.3	lib.badcast	`bad_cast` exception
17.2	lib.string	The string class
17.3	lib.io	Input/output
17.4	lib.clib	C library
17.4.1	lib.cintro	Introduction
17.4.1.1	lib.defs	Definitions of terms
17.4.1.2	lib.headers	Standard headers
17.4.1.2.1	lib.reserved	Reserved identifiers
17.4.1.3	lib.errors	Errors `<errno.h>`
17.4.1.4	lib.limits	Limits `<float.h>` and `<limits.h>`
17.4.1.5	lib.stddef	Common definitions `<stddef.h>`
17.4.1.6	lib.libuse	Use of library functions
17.4.2	lib.assert	Diagnostics `<assert.h>`

17.4.3	lib.ctype	Character handling <ctype.h>
17.4.4	lib.locale	Localization <locale.h>
17.4.5	lib.math	Mathematics <math.h>
17.4.6	lib.nonlocal	Nonlocal jumps <setjmp.h>
17.4.6.1	lib.setjmp	Save calling environment
17.4.6.1.1	lib.setjmp.mac	The setjmp macro
17.4.6.2	lib.longjmp	Restore calling environment
17.4.6.2.1	lib.longjmp.func	The longjmp function
17.4.7	lib.signal	Signal handling <signal.h>
17.4.8	lib.stdarg	Variable arguments <stdarg.h>
17.4.9	lib.stdio	Input/output <stdio.h>
17.4.10	lib.stdlib	General utilities <stdlib.h>
17.4.10.1	lib.strconv	NTCS conversion functions
17.4.10.2	lib.rand	Pseudo-random sequence generation functions
17.4.10.3	lib.memmgmt	Memory management functions
17.4.10.4	lib.env	Communication with the environment
17.4.10.4.1	lib.abort	The abort Function
17.4.10.4.2	lib.atexit	The atexit function
17.4.10.4.3	lib.exit	The exit function
17.4.10.4.4	lib.getenv	The getenv function
17.4.10.4.5	lib.system	The system function
17.4.10.5	lib.search	Searching and sorting utilities
17.4.10.6	lib.intarith	Integer arithmetic functions
17.4.10.7	lib.multichar	Multibyte character functions
17.4.10.8	lib.multintcs	Multibyte NTCS functions
17.4.11	lib.ntcshandl	NTCS handling <string.h>
17.4.11.1	lib.ntcs.cnvnt	NTCS function conventions
17.4.11.2	lib.copy	Copying functions
17.4.11.3	lib.concat	Concatenation functions
17.4.11.4	lib.compar	Comparison functions
17.4.11.5	lib.search	Search functions
17.4.11.5.1	lib.memchr	The memchr functions
17.4.11.5.2	lib.strchr	The strchr functions
17.4.11.5.3	lib.strcspn	The strcspn function
17.4.11.5.4	lib.strpbrk	The strpbrk functions
17.4.11.5.5	lib.strrchr	The strrchr function
17.4.11.5.6	lib.strspn	The strspn function
17.4.11.5.7	lib.strstr	The strstr functions
17.4.11.5.8	lib.strtok	The strtok function
17.4.11.6	lib.misc	Miscellaneous functions
17.4.12	lib.time	Date and time <time.h>
17.4.13	lib.cfuture	Future c library directions
17.5	lib.future	Future library directions
18	gram	Appendix A: Grammar summary
18.1	gram.key	Keywords
18.2	gram.expr	Expressions
18.3	gram.decl	Declarators
18.4	gram.class	Class declarations
18.5	gram.stmt	Statements
18.6	gram.cpp	Preprocessor
18.7	gram.temp	Templates
18.8	gram.except	Exception handling
19	diff	Appendix B: Compatibility
19.1	diff.c	Extensions
19.1.1	diff.early	C++ features available in 1985

19.1.2	diff.c++	C++ features added since 1985
19.2	diff.ansi	C++ and ISO C
19.2.1	diff.lex	Chapter 2: Lexical conventions
19.2.2	diff.basic	Chapter 3: Basic concepts
19.2.3	diff.conv	Chapter 4: Standard conversions
19.2.4	diff.expr	Chapter 5: Expressions
19.2.5	diff.stat	Chapter 6: Statements
19.2.6	diff.dcl	Chapter 7: Declarations
19.2.7	diff.decl	Chapter 8: Declarators
19.2.8	diff.class	Chapter 9: Classes
19.2.9	diff.cpp	Chapter 16: Preprocessing directives
19.3	diff.anac	Anachronisms
19.3.1	diff.fct.def	Old style function definitions
19.3.2	diff.base.init	Old style base class initializer
19.3.3	diff.this	Assignment to `this`
19.3.4	diff.bound	Cast of bound pointer
19.3.5	diff.class	Nonnested classes

Listing by symbolic name

basic	3	Basic concepts
basic.def	3.1	Declarations and definitions
basic.derived	3.6.2	Derived types
basic.fundamental	3.6.1	Fundamental types
basic.link	3.3	Program and linkage
basic.lval	3.7	Lvalues
basic.scope	3.2	Scopes
basic.start	3.4	Start and termination
basic.stc	3.5	Storage classes
basic.type.name	3.6.4	Type names
basic.type.qualifier	3.6.3	CV-qualifiers
basic.types	3.6	Types
class	9	Classes
class.abstract	10.3	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.ambig	10.1.1	Ambiguities
class.base.init	12.6.2	Initializing bases and members
class.bit	9.6	Bit-fields
class.cdtor	12.7	Constructors and destructors
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.inline	9.3.2	Inline member functions
class.local	9.8	Local class declarations
class.mem	9.2	Class members
class.mfct	9.3	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.scope	10.4	Summary of scope rules
class.scope0	9.2.1	Scope rules for classes
class.static	9.4	Static members
class.temporary	12.2	Temporary objects
class.this	9.3.1	The `this` pointer
class.union	9.5	Unions
class.virtual	10.2	Virtual functions
conv	4	Standard conversions
conv.arith	4.5	Arithmetic conversions

conv.double	4.3	Float and double
conv.float	4.4	Floating and integral
conv.integral	4.2	Integral conversions
conv.mem	4.8	Pointers to members
conv.prom	4.1	Integral promotions
conv.ptr	4.6	Pointer conversions
conv.ref	4.7	Reference 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	8.1.1	Ambiguity resolution
dcl.array	8.2.4	Arrays
dcl.asm	7.3	Asm declarations
dcl.dcl	7	Declarations
dcl.decl	8	Declarators
dcl.enum	7.2	Enumeration declarations
dcl.fct	8.2.5	Functions
dcl.fct.def	8.3	Function definitions
dcl.fct.default	8.2.6	Default parameters
dcl.fct.spec	7.1.2	Function specifiers
dcl.friend	7.1.5	The `friend` specifier
dcl.init	8.4	Initializers
dcl.init.aggr	8.4.1	Aggregates
dcl.init.ref	8.4.3	References
dcl.init.string	8.4.2	Character arrays
dcl.link	7.4	Linkage specifications
dcl.meaning	8.2	Meaning of declarators
dcl.mptr	8.2.3	Pointers to members
dcl.name	8.1	Type names
dcl.ptr	8.2.1	Pointers
dcl.ref	8.2.2	References
dcl.spec	7.1	Specifiers
dcl.stc	7.1.1	Storage class specifiers
dcl.template	7.1.4	The `template` specifier
dcl.type	7.1.6	Type specifiers
dcl.typedef	7.1.3	The `typedef` specifier
diff	19	Appendix B: Compatibility
diff.anac	19.3	Anachronisms
diff.ansi	19.2	C++ and ISO C
diff.base.init	19.3.2	Old style base class initializer
diff.basic	19.2.2	Chapter 3: Basic concepts
diff.bound	19.3.4	Cast of bound pointer
diff.c	19.1	Extensions
diff.c++	19.1.2	C++ features added since 1985

diff.class	19.2.8	Chapter 9: Classes
diff.class	19.3.5	Nonnested classes
diff.conv	19.2.3	Chapter 4: Standard conversions
diff.cpp	19.2.9	Chapter 16: Preprocessing directives
diff.dcl	19.2.6	Chapter 7: Declarations
diff.decl	19.2.7	Chapter 8: Declarators
diff.early	19.1.1	C++ features available in 1985
diff.expr	19.2.4	Chapter 5: Expressions
diff.fct.def	19.3.1	Old style function definitions
diff.lex	19.2.1	Chapter 2: Lexical conventions
diff.stat	19.2.5	Chapter 6: Statements
diff.this	19.3.3	Assignment to `this`
except	15	Exception handling
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.delete	5.3.4	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	5.3.3	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	5.3.1	Increment and decrement
expr.prim	5.1	Primary expressions
expr.ref	5.2.4	Class member access
expr.rel	5.9	Relational operators
expr.shift	5.8	Shift operators
expr.sizeof	5.3.2	Sizeof
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 operators
expr.xor	5.12	Bitwise exclusive OR operator
gram	18	Appendix A: Grammar summary
gram.class	18.4	Class declarations
gram.cpp	18.6	Preprocessor
gram.decl	18.3	Declarators
gram.except	18.8	Exception handling
gram.expr	18.2	Expressions
gram.key	18.1	Keywords
gram.stmt	18.5	Statements
gram.temp	18.7	Templates
intro	1	Introduction
intro.defs	1.4	Definitions of terms

intro.memory	1.3	The C++ memory model
lex	2	Lexical conventions
lex.ccon	2.9.2	Character literals
lex.comment	2.6	Comments
lex.digraph	2.4	Digraph sequences
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	17	Library
lib.abort	17.4.10.4.1	The `abort` Function
lib.assert	17.4.2	Diagnostics `<assert.h>`
lib.atexit	17.4.10.4.2	The `atexit` function
lib.badcast	17.1.3.3.3	`bad_cast` exception
lib.cfuture	17.4.13	Future c library directions
lib.cintro	17.4.1	Introduction
lib.clib	17.4	C library
lib.compar	17.4.11.4	Comparison functions
lib.concat	17.4.11.3	Concatenation functions
lib.copy	17.4.11.2	Copying functions
lib.ctype	17.4.3	Character handling `<ctype.h>`
lib.defs	17.4.1.1	Definitions of terms
lib.env	17.4.10.4	Communication with the environment
lib.errors	17.4.1.3	Errors `<errno.h>`
lib.exception	17.1.3	Exceptions
lib.excptterm	17.1.3.1	Abnormal termination
lib.exit	17.4.10.4.3	The `exit` function
lib.free	17.1.1	Free store `<new>`
lib.future	17.5	Future library directions
lib.getenv	17.4.10.4.4	The `getenv` function
lib.headers	17.4.1.2	Standard headers
lib.intarith	17.4.10.6	Integer arithmetic functions
lib.io	17.3	Input / output
lib.libuse	17.4.1.6	Use of library functions
lib.limits	17.4.1.4	Limits `<float.h>` and `<limits.h>`
lib.locale	17.4.4	Localization `<locale.h>`
lib.longjmp	17.4.6.2	Restore calling environment
lib.longjmp.func	17.4.6.2.1	The `longjmp` function
lib.math	17.4.5	Mathematics `<math.h>`
lib.memchr	17.4.11.5.1	The `memchr` functions
lib.memmgmt	17.4.10.3	Memory management functions
lib.misc	17.4.11.6	Miscellaneous functions
lib.multichar	17.4.10.7	Multibyte character functions
lib.multintcs	17.4.10.8	Multibyte NTCS functions
lib.newhndl	17.1.1.4	*new-handler* function
lib.nonlocal	17.4.6	Nonlocal jumps `<setjmp.h>`
lib.ntcs.cnvnt	17.4.11.1	NTCS function conventions
lib.ntcshandl	17.4.11	NTCS handling `<string.h>`
lib.opdelete	17.1.1.2	`operator delete()` and `operator`

		delete[]()
lib.opnew	17.1.1.1	operator new() and operator new[]()
lib.placeopnew	17.1.1.3	*placement* operator new()
lib.predexcpt	17.1.3.3	Predefined exceptions
lib.rand	17.4.10.2	Pseudo-random sequence generation functions
lib.reserved	17.4.1.2.1	Reserved identifiers
lib.search	17.4.10.5	Searching and sorting utilities
lib.search	17.4.11.5	Search functions
lib.setjmp	17.4.6.1	Save calling environment
lib.setjmp.mac	17.4.6.1.1	The setjmp macro
lib.setnewhndl	17.1.1.5	set_new_handler()
lib.setterm	17.1.3.1.3	set_terminate()
lib.setunexp	17.1.3.2.3	set_unexpected()
lib.signal	17.4.7	Signal handling <signal.h>
lib.stdarg	17.4.8	Variable arguments <stdarg.h>
lib.stddef	17.4.1.5	Common definitions <stddef.h>
lib.stdio	17.4.9	Input/output <stdio.h>
lib.stdlib	17.4.10	General utilities <stdlib.h>
lib.strchr	17.4.11.5.2	The strchr functions
lib.strconv	17.4.10.1	NTCS conversion functions
lib.strcspn	17.4.11.5.3	The strcspn function
lib.string	17.2	The string class
lib.strpbrk	17.4.11.5.4	The strpbrk functions
lib.strrchr	17.4.11.5.5	The strrchr function
lib.strspn	17.4.11.5.6	The strspn function
lib.strstr	17.4.11.5.7	The strstr functions
lib.strtok	17.4.11.5.8	The strtok function
lib.sup	17.1	Language support
lib.system	17.4.10.4.5	The system function
lib.termfn	17.1.3.1.2	*terminate-function*
lib.terminate	17.1.3.1.1	terminate()
lib.time	17.4.12	Date and time <time.h>
lib.typeinfo	17.1.2	Type identification <type_info>
lib.unexpected	17.1.3.2.1	unexpected()
lib.unexpfn	17.1.3.2.2	*unexpected-function*
lib.violspec	17.1.3.2	Violating *exception-specification*s
lib.xalloc	17.1.3.3.2	xalloc exception
lib.xmsg	17.1.3.3.1	xmsg exception
over	13	Overloading
over.ass	13.4.3	Assignment
over.binary	13.4.2	Binary operators
over.call	13.4.4	Function call
over.dcl	13.1	Declaration matching
over.inc	13.4.7	Increment and decrement
over.match	13.2	Argument matching
over.oper	13.4	Overloaded operators
over.over	13.3	Address of overloaded function
over.ref	13.4.6	Class member access
over.sub	13.4.5	Subscripting
over.unary	13.4.1	Unary operators
overview	1.1	Overview
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	1.2	Syntax notation
temp	14	Templates