

# ISO/IEC TS 19568:2017-03 (E)

## Programming Languages - C++ Extensions for Library Fundamentals

---

Contents	Page
<b>Foreword</b> . . . . .	<b>vii</b>
<b>1 General</b> . . . . .	<b>1</b>
1.1 Scope . . . . .	1
1.2 Normative references . . . . .	1
1.3 Namespaces, headers, and modifications to standard classes . . . . .	1
1.4 Terms and definitions . . . . .	2
1.5 Future plans (Informative) . . . . .	2
1.6 Feature-testing recommendations (Informative) . . . . .	3
<b>2 Modifications to the C++ Standard Library</b> . . . . .	<b>5</b>
2.1 Uses-allocator construction . . . . .	5
<b>3 General utilities library</b> . . . . .	<b>6</b>
3.1 Utility components . . . . .	6
3.1.1 Header <experimental/utility> synopsis . . . . .	6
3.1.2 Class erased_type . . . . .	6
3.2 Tuples . . . . .	6
3.2.1 Header <experimental/tuple> synopsis . . . . .	6
3.2.2 Calling a function with a tuple of arguments . . . . .	7
3.3 Metaprogramming and type traits . . . . .	7
3.3.1 Header <experimental/type_traits> synopsis . . . . .	7
3.3.2 Other type transformations . . . . .	11
3.3.3 Logical operator traits . . . . .	12
3.3.4 Detection idiom . . . . .	13
3.4 Compile-time rational arithmetic . . . . .	14
3.4.1 Header <experimental/ratio> synopsis . . . . .	14
3.5 Time utilities . . . . .	15
3.5.1 Header <experimental/chrono> synopsis . . . . .	15
3.6 System error support . . . . .	15
3.6.1 Header <experimental/system_error> synopsis . . . . .	15
3.7 Class template propagate_const . . . . .	15
3.7.1 Class template propagate_const general . . . . .	15
3.7.2 Header <experimental/propagate_const> synopsis . . . . .	16
3.7.3 propagate_const requirements on T . . . . .	18
3.7.3.1 propagate_const requirements on class type T . . . . .	18
3.7.4 propagate_const constructors . . . . .	19
3.7.5 propagate_const assignment . . . . .	19
3.7.6 propagate_const const observers . . . . .	20
3.7.7 propagate_const non-const observers . . . . .	20
3.7.8 propagate_const modifiers . . . . .	21
3.7.9 propagate_const relational operators . . . . .	21
3.7.10 propagate_const specialized algorithms . . . . .	23
3.7.11 propagate_const underlying pointer access . . . . .	23
3.7.12 propagate_const hash support . . . . .	23
3.7.13 propagate_const comparison function objects . . . . .	23
<b>4 Function objects</b> . . . . .	<b>25</b>
4.1 Header <experimental/functional> synopsis . . . . .	25
4.2 Class template function . . . . .	26
4.2.1 function construct/copy/destroy . . . . .	28
4.2.2 function modifiers . . . . .	29

4.3	Searchers . . . . .	29
4.3.1	Class template default_searcher . . . . .	29
4.3.1.1	default_searcher creation functions . . . . .	30
4.3.2	Class template boyer_moore_searcher . . . . .	30
4.3.2.1	boyer_moore_searcher creation functions . . . . .	31
4.3.3	Class template boyer_moore_horspool_searcher . . . . .	31
4.3.3.1	boyer_moore_horspool_searcher creation functions . . . . .	32
4.4	Function template not_fn . . . . .	33
<b>5</b>	<b>Optional objects . . . . .</b>	<b>34</b>
5.1	In general . . . . .	34
5.2	Header <experimental/optional> synopsis . . . . .	34
5.3	optional for object types . . . . .	35
5.3.1	Constructors . . . . .	37
5.3.2	Destructor . . . . .	39
5.3.3	Assignment . . . . .	40
5.3.4	Swap . . . . .	43
5.3.5	Observers . . . . .	43
5.4	In-place construction . . . . .	44
5.5	No-value state indicator . . . . .	44
5.6	Class bad_optional_access . . . . .	45
5.7	Relational operators . . . . .	45
5.8	Comparison with nullopt . . . . .	45
5.9	Comparison with T . . . . .	46
5.10	Specialized algorithms . . . . .	47
5.11	Hash support . . . . .	47
<b>6</b>	<b>Class any . . . . .</b>	<b>48</b>
6.1	Header <experimental/any> synopsis . . . . .	48
6.2	Class bad_any_cast . . . . .	49
6.3	Class any . . . . .	49
6.3.1	any construct/destroy . . . . .	49
6.3.2	any assignments . . . . .	50
6.3.3	any modifiers . . . . .	51
6.3.4	any observers . . . . .	51
6.4	Non-member functions . . . . .	51
<b>7</b>	<b>string_view . . . . .</b>	<b>54</b>
7.1	Header <experimental/string_view> synopsis . . . . .	54
7.2	Class template basic_string_view . . . . .	55
7.3	basic_string_view constructors and assignment operators . . . . .	57
7.4	basic_string_view iterator support . . . . .	58
7.5	basic_string_view capacity . . . . .	59
7.6	basic_string_view element access . . . . .	59
7.7	basic_string_view modifiers . . . . .	60
7.8	basic_string_view string operations . . . . .	60
7.8.1	Searching basic_string_view . . . . .	62
7.9	basic_string_view non-member comparison functions . . . . .	63
7.10	Inserters and extractors . . . . .	64
7.11	Hash support . . . . .	65
<b>8</b>	<b>Memory . . . . .</b>	<b>66</b>
8.1	Header <experimental/memory> synopsis . . . . .	66
8.2	Shared-ownership pointers . . . . .	69
8.2.1	Class template shared_ptr . . . . .	69
8.2.1.1	shared_ptr constructors . . . . .	72

	8.2.1.2	shared_ptr observers	74
	8.2.1.3	shared_ptr casts	75
	8.2.1.4	shared_ptr hash support	75
	8.2.2	Class template weak_ptr	75
	8.2.2.1	weak_ptr constructors	76
8.3		Type-erased allocator	77
8.4		Header <experimental/memory_resource> synopsis	77
8.5		Class memory_resource	78
	8.5.1	Class memory_resource overview	78
	8.5.2	memory_resource public member functions	79
	8.5.3	memory_resource protected virtual member functions	79
	8.5.4	memory_resource equality	80
8.6		Class template polymorphic_allocator	80
	8.6.1	Class template polymorphic_allocator overview	80
	8.6.2	polymorphic_allocator constructors	81
	8.6.3	polymorphic_allocator member functions	81
	8.6.4	polymorphic_allocator equality	83
8.7		template alias resource_adaptor	83
	8.7.1	resource_adaptor	83
	8.7.2	resource_adaptor_imp constructors	84
	8.7.3	resource_adaptor_imp member functions	84
8.8		Access to program-wide memory_resource objects	85
8.9		Pool resource classes	85
	8.9.1	Classes synchronized_pool_resource and unsynchronized_pool_resource	85
	8.9.2	pool_options data members	87
	8.9.3	pool_resource constructors and destructors	88
	8.9.4	pool_resource members	88
8.10		Class monotonic_buffer_resource	89
	8.10.1	Class monotonic_buffer_resource overview	89
	8.10.2	monotonic_buffer_resource constructor and destructor	90
	8.10.3	monotonic_buffer_resource members	91
8.11		Alias templates using polymorphic memory resources	91
	8.11.1	Header <experimental/string> synopsis	91
	8.11.2	Header <experimental/deque> synopsis	92
	8.11.3	Header <experimental/forward_list> synopsis	92
	8.11.4	Header <experimental/list> synopsis	92
	8.11.5	Header <experimental/vector> synopsis	93
	8.11.6	Header <experimental/map> synopsis	93
	8.11.7	Header <experimental/set> synopsis	94
	8.11.8	Header <experimental/unordered_map> synopsis	94
	8.11.9	Header <experimental/unordered_set> synopsis	95
	8.11.10	Header <experimental/regex> synopsis	95
8.12		Non-owning pointers	96
	8.12.1	Class template observer_ptr overview	96
	8.12.2	observer_ptr constructors	97
	8.12.3	observer_ptr observers	97
	8.12.4	observer_ptr conversions	97
	8.12.5	observer_ptr modifiers	97
	8.12.6	observer_ptr specialized algorithms	98
	8.12.7	observer_ptr hash support	99
<b>9</b>		<b>Containers</b>	<b>100</b>
	9.1	Uniform container erasure	100

9.1.1	Header synopsis	100
9.1.2	Function template erase_if	101
9.1.3	Function template erase	102
9.2	Class template array	102
9.2.1	Header <experimental/array> synopsis	102
9.2.2	Array creation functions	103
<b>10</b>	<b>Iterators library</b>	<b>104</b>
10.1	Header <experimental/iterator> synopsis	104
10.2	Class template ostream_joiner	104
10.2.1	ostream_joiner constructor	105
10.2.2	ostream_joiner operations	105
10.2.3	ostream_joiner creation function	105
<b>11</b>	<b>Futures</b>	<b>106</b>
11.1	Header <experimental/future> synopsis	106
11.2	Class template promise	106
11.3	Class template packaged_task	107
<b>12</b>	<b>Algorithms library</b>	<b>109</b>
12.1	Header <experimental/algorithm> synopsis	109
12.2	Search	109
12.3	Sampling	110
12.4	Shuffle	110
<b>13</b>	<b>Numerics library</b>	<b>111</b>
13.1	Generalized numeric operations	111
13.1.1	Header <experimental/numeric> synopsis	111
13.1.2	Greatest common divisor	111
13.1.3	Least common multiple	111
13.2	Random number generation	112
13.2.1	Header <experimental/random> synopsis	112
13.2.2	Utilities	112
13.2.2.1	Function template randint	112
<b>14</b>	<b>Reflection library</b>	<b>113</b>
14.1	Class source_location	113
14.1.1	Header <experimental/source_location> synopsis	113
14.1.2	source_location creation	114
14.1.3	source_location field access	114