

# ISO/IEC TS 19568:2024-08 (E)

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

---

<b>Contents</b>	<b>Page</b>
<b>Foreword</b> . . . . .	<b>vi</b>
<b>Introduction</b> . . . . .	<b>viii</b>
<b>1 Scope</b> . . . . .	<b>1</b>
<b>2 Normative references</b> . . . . .	<b>2</b>
<b>3 Terms and definitions</b> . . . . .	<b>3</b>
<b>4 General principles</b> . . . . .	<b>4</b>
4.1 Namespaces, headers, and modifications to standard classes . . . . .	4
4.2 Feature-testing recommendations . . . . .	5
<b>5 Modifications to the C++ Standard Library</b> . . . . .	<b>7</b>
5.1 General . . . . .	7
5.2 Exception requirements . . . . .	7
<b>6 General utilities library</b> . . . . .	<b>8</b>
6.1 Constness propagation . . . . .	8
6.1.1 Header <code>&lt;experimental/propagate_const&gt;</code> synopsis . . . . .	8
6.1.2 Class template <code>propagate_const</code> . . . . .	10
6.1.2.1 Overview . . . . .	10
6.1.2.2 General requirements on T . . . . .	12
6.1.2.3 Requirements on class type T . . . . .	12
6.1.2.4 Constructors . . . . .	13
6.1.2.5 Assignment . . . . .	13
6.1.2.6 Const observers . . . . .	14
6.1.2.7 Non-const observers . . . . .	14
6.1.2.8 Modifiers . . . . .	15
6.1.2.9 Relational operators . . . . .	15
6.1.2.10 Specialized algorithms . . . . .	18
6.1.2.11 Underlying pointer access . . . . .	18
6.1.2.12 Hash support . . . . .	18
6.1.2.13 Comparison function objects . . . . .	19
6.2 Scope guard support . . . . .	20
6.2.1 Header <code>&lt;experimental/scope&gt;</code> synopsis . . . . .	20
6.2.2 Class templates <code>scope_exit</code> , <code>scope_fail</code> , and <code>scope_success</code> . . . . .	21
6.2.3 Class template <code>unique_resource</code> . . . . .	24
6.2.3.1 Overview . . . . .	24
6.2.3.2 Constructors . . . . .	26
6.2.3.3 Destructor . . . . .	27
6.2.3.4 Assignment . . . . .	27
6.2.3.5 Other member functions . . . . .	29
6.2.3.6 <code>unique_resource</code> creation . . . . .	30

6.3	Metaprogramming and type traits . . . . .	31
6.3.1	Header <experimental/type_traits> synopsis . . . . .	31
6.3.2	Other type transformations . . . . .	32
6.3.3	Detection idiom . . . . .	34
<b>7</b>	<b>Function objects . . . . .</b>	<b>36</b>
7.1	Header <experimental/functional> synopsis . . . . .	36
7.2	Class template function . . . . .	36
7.2.1	Overview . . . . .	36
7.2.2	Construct/copy/destroy . . . . .	38
7.2.3	Modifiers . . . . .	39
7.2.4	Observers . . . . .	39
<b>8</b>	<b>Memory . . . . .</b>	<b>40</b>
8.1	Header <experimental/memory> synopsis . . . . .	40
8.2	Non-owning (observer) pointers . . . . .	41
8.2.1	Class template observer_ptr overview . . . . .	41
8.2.2	observer_ptr constructors . . . . .	42
8.2.3	observer_ptr observers . . . . .	43
8.2.4	observer_ptr conversions . . . . .	43
8.2.5	observer_ptr modifiers . . . . .	43
8.2.6	observer_ptr specialized algorithms . . . . .	44
8.2.7	observer_ptr hash support . . . . .	45
8.3	Header <experimental/memory_resource> synopsis . . . . .	45
8.4	Alias template resource_adaptor . . . . .	45
8.4.1	resource_adaptor . . . . .	45
8.4.2	resource_adaptor_imp constructors . . . . .	47
8.4.3	resource_adaptor_imp member functions . . . . .	47
<b>9</b>	<b>Iterators library . . . . .</b>	<b>48</b>
9.1	Header <experimental/iterator> synopsis . . . . .	48
9.2	Class template ostream_joiner . . . . .	48
9.2.1	Overview . . . . .	48
9.2.2	Constructor . . . . .	49
9.2.3	Operations . . . . .	50
9.2.4	Creation function . . . . .	50
<b>10</b>	<b>Algorithms library . . . . .</b>	<b>51</b>
10.1	Header <experimental/algorithm> synopsis . . . . .	51
10.2	Sampling . . . . .	51
10.3	Shuffle . . . . .	52
<b>11</b>	<b>Numerics library . . . . .</b>	<b>53</b>
11.1	Random number generation . . . . .	53
11.1.1	Header <experimental/random> synopsis . . . . .	53
11.1.2	Function template randint . . . . .	53