Glossary

maximal fundamental alignment – that of `std::max_align_t`, which, for any given platform, is at least as strict as that of every scalar type. **`alignof`** (193)

mechanism – a term used to characterize a class type capable of instantiating objects and whose purpose it is to provide a service, such as scoped guard, *lock*, *socket*, etc. A mechanism does not attempt to represent a platonic value, such as does `std::complex<double>`, nor even an in-process one (e.g., one whose value incorporates a memory address as a salient attribute). Delegating Ctors (51), Opaque **`enum`**s (663), *Rvalue* References (789)

member – an entity other than a **`friend`** function — such as a data member, member function, user-defined type, or type alias — that is declared to reside within the scope of a class type; see also hidden-friend idiom. **`union`** '11 (1174)

member function – one that is a member of a **`class`**, **`struct`**, or **`union`**; see also free function. *Rvalue* References (793), Variadic Templates (892)

member initializer list – the syntax used in constructors to initialize direct base classes, **`virtual`** base classes, and non**`static`** data members. Note that the initialization order is fixed by the relative order in which base classes and data members are declared (some compilers may warn if the relative orderings differ). Delegating ~~Ctors~~ (46), Braced Init (230), Default Member Init (318)

member operator – a user-defined (overloaded) operator (e.g., copy assignment) that is implemented as a member of a class and, unless implicitly **`static`** (e.g., class-specific operators **`new`** and **`delete`**), has access to the object's **`this`** pointer as an implicit argument.

memory barrier – a form of *synchronization primitive* that is used to enforce an observable modification order for one or more memory locations (a.k.a. a fence), facilitating the coordination of access to data from concurrent execution contexts; see also multithreading contexts. Function **`static`** '11 (80)

memory diffusion – the process by which the size of the working set increases over time, despite constant memory utilization; see also memory fragmentation. **`noexcept`** Operator (628), *Rvalue* References (788)

memory-fence instruction – a machine-level instruction enforcing an observable modification order for one or more memory locations; see also memory barrier. `carries_dependency` (999)

memory fragmentation – the process by which the maximum available contiguous chunk of dynamically allocatable memory decreases over time, despite consistent memory utilization; see also memory diffusion.

memory leak – a type of resource leak involving specifically a memory resource. Note that use of class-specific memory allocators (e.g., implemented in terms of class-specific **`new`** and **`delete`**) can create a *pseudo* memory leak that can be as problematic as a genuine one; see **lakos96**, section 10.3.4, "Class-Specific Memory Management," pp. 698–711, in particular, Figure 10-30, p. 709. Function **`static`** '11 (74)

metafunction – a compile-time operation whose parameters and results are code subject to compilation, such as functions, types, and compile-time constants, rather than runtime values. C++ metafunctions, such as the standard type traits provided in `<type_traits>`, are implemented as class, variable, and alias templates that compute types and/or compile-time constants based on the values of their metaparameters, supplied as template arguments. Note that modern C++ introduced a related feature; see Section 2.1."**`constexpr`** Functions" on page 257. Forwarding References (381), Generalized PODs '11 (469), **`noexcept`** Operator (643), **`constexpr`** Functions '14 (963)