Glossary

associated lifetime extension that would ensue. Range `for` (701), *Rvalue* References (710), Lambda Captures (993), `noexcept` Specifier (1118)

lvalue-to-rvalue conversion – the implicit conversion that occurs when a *prvalue* is needed from an expression whose value category is *glvalue* — effectively, the process of reading the value of an object. Generalized PODs '11 (501), *Rvalue* References (814)

managed allocator – one that keeps track of each active allocation made through it such that it is able to unilaterally release all outstanding memory at destruction or, perhaps, via a single (e.g., `release`) operation (a.k.a. *winking out*). As of C++17, the C++ Standard Library supports this sort of functionality via two concrete classes derived from `std::pmr::memory_resource`: (1) a (special-purpose) monotonic-allocator resource that treats each individual deallocation as a no-op and (2) a (general-purpose) multipool-allocator resource that is substitutable for the global allocator supplied by the C++ run time. `final` (1021)

mandatory RVO – a requirement, as of C++17, for functions returning objects by value (a form of guaranteed copy elision) that, when the value in a unique `return` statement is a *prvalue*, the object, instead of being copied or moved, is constructed in place in its final destination in the calling function's context. *Rvalue* References (807)

mangled name – one created by the compiler and potentially used by the linker to uniquely identify distinct entities having the same name but defined in different contexts, e.g., entities having the same name but declared in multiple scopes or an (overloaded) function having multiple signatures. This feature of the ABI also enables *type-safe* linkage, which helps to avoid mismatches across translation units — e.g., that a function defined to take an `int` cannot be bound to (the use of) a declaration of a function having the same name but taking a `long`. A function template, in addition to its parameters, necessarily incorporates the return type as part of the mangled name. The use of C linkage eliminates such mangling, thereby preventing the overloading of such functions and function templates. Note that a global variable, e.g., a `double`, declared at file or namespace scope might, but is not required to, use a mangled name to identify its type in the ABI; hence, *type-safe* linkage cannot be relied upon to detect such mismatches; see local declarations. `inline namespace` (1056), `noexcept` Specifier (1114)

manifestly constant evaluated – implies, for a given expression, that it is used in a context that requires it be evaluated at compile time, such as an array bound or as an argument for a non-type template parameter; see also constant expression. `constexpr` Functions (258)

mantissa – the part of a floating-point representation that contains the significant digits; note that, when represented in binary, the leading 1 can be omitted. Digit Separators (155)

materialize – the act of temporary materialization — i.e., that of the compiler creating a temporary object in the address space to represent a given *prvalue*. *Rvalue* References (717), Ref-Qualifiers (1163)

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)