Glossary

and *prvalue*. In addition, two compound value categories — *glvalue* (comprising *lvalue* and *xvalue*) and *rvalue* (comprising *xvalue* and *prvalue*) — serve to characterize values that (1) have identity and (2) are expiring, respectively; see Section 2.1.“*Rvalue* References” on page 710. **`decltype`** (25), Forwarding References (377), Lambdas (590), Range **`for`** (680), *Rvalue* References (710), *Generic* Lambdas (972), Lambda Captures (992), **`noexcept`** Specifier (1145), Ref-Qualifiers (1153), **`auto`** Return (1184), **`decltype(auto)`** (1205)

value constructor – one designed to *assemble* (as opposed to *copy* or *move*) an overall value from one or more supplied arguments and that (absent defaulted arguments) is never also a default constructor, copy constructor, or move constructor. Defaulted Functions (37), Generalized PODs ’11 (450), *Rvalue* References (753), User-Defined Literals (836), Variadic Templates (942)

value initialization – a form of initialization, typically invoked by supplying an empty (rather than absent) initializer list, such as `()` or `{}`, that (1) performs zero initialization for scalar types as well as class types having a trivial default constructor, (2) invokes the default constructor for class types having a user-provided default constructor, or (3) performs zero initialization and then invokes the default constructor for all other class types, i.e., those that have a compiler-generated non-trivial default constructor. For an array type, each individual element is value initialized. Value initialization for a type having a deleted or *ambiguous* default constructor is ill formed — even if said initialization would not involve invoking the default constructor. Braced Init (216), **`constexpr`** Functions (273), Generalized PODs ’11 (493)

value initialized – implies, for a given object, that it has undergone value initialization. Braced Init (221), Generalized PODs ’11 (412), *Rvalue* References (764)

value representation – the bits in an object’s footprint that represent its value, excluding, e.g., those used for padding or to represent a virtual-function-table pointer or virtual-base pointer. Generalized PODs ’11 (405)

value semantic (of a type) – implies, for a given type, that it has value semantics. Defaulted Functions (36), Delegating ~~Ctors~~ (48), **`alignof`** (187), Opaque **`enum`**s (663), *Rvalue* References (743)

value-semantic type (VST) – one, specifically a class type, that has value semantics. Forwarding References (386), Generalized PODs ’11 (452), *Rvalue* References (742), Lambda Captures (992), **`friend`** ’11 (1034)

value semantics – the fundamental, language-independent, mathematical principles that must be satisfied by any type that properly represents a platonic value; see **lakos15a**. Importantly, two objects of a value-semantic type do *not* have the same value (as defined by their respective salient attributes) if there exists a sequence of salient operations (a.k.a. a distinguishing sequence) that, when applied to each object separately, mutates the respective objects such that they can be observed not to have (i.e., represent) the same value. Note that a well-written C++ value-semantic type will also be a regular type (see **stepanov09**, section 1.5, “Regular Types,” pp. 6–8) unless its (homogeneous) equality-comparison operator (`==`) would be too computationally complex; if it’s omitted, the type becomes *semiregular* (see **stepanov15**, section 10.3, “Concepts,” pp. 181–184, specifically p. 184); see also **lakos15b**. Also note, as of C++20, the Standard Library supports the concepts `std::regular` and `std::semiregular`. **`noexcept`** Operator (627), *Rvalue* References (811)

variable – a named object having *automatic*, *static*, or *thread* storage duration.

variable template – one — e.g., `template <typename T> T var;` — that can be instantiated to yield a family of like-named variables, each of distinct type, e.g., `var<int>`, `var<double>`; see Section 1.2.“Variable Templates” on page 157. **`constexpr`** Variables (302)