

Glossary

- user-defined literal (UDL)** – a string or numeric literal whose meaning is defined by either the user or the standard library. A UDL suffix at the end of the literal token identifies which UDL operator is used to interpret it and produce a value. [Generalized PODs '11 \(462\)](#), [User-Defined Literals \(837\)](#)
- user-defined type (UDT)** – a (1) **class**, (2) **struct**, (3) **union**, or (4) enumeration type (**enum** or **enum class**; see Section 2.1. “**enum class**” on page 332). [Delegating Ctors \(46\)](#), [alignas \(168\)](#), [Default Member Init \(322\)](#), [Generalized PODs '11 \(462\)](#), [initializer_list \(553\)](#), [noexcept Operator \(622\)](#), [Rvalue References \(742\)](#), [User-Defined Literals \(835\)](#), [final \(1012\)](#), [friend '11 \(1031\)](#)
- user provided** – implies, for a given function, that it is user declared and is not explicitly defaulted or deleted on its first declaration. [Function static '11 \(80\)](#), [Braced Init \(217\)](#), [Generalized PODs '11 \(413\)](#), [Rvalue References \(742\)](#), [noexcept Specifier \(1087\)](#)
- user-provided special member function** – a special member function that is user provided. [Defaulted Functions \(33\)](#)
- using declaration** – one that begins with **using** and introduces an existing declaration into the current scope and is sometimes used colloquially to refer to the declaration of a type alias or alias template; see Section 1.1. “**using Aliases**” on page 133. [constexpr Functions \(268\)](#), [Inheriting Ctors \(535\)](#)
- using directive** – short for **using-namespace** directive. [constexpr Functions \(268\)](#), [User-Defined Literals \(842\)](#), [inline namespace \(1056\)](#)
- using-namespace directive** – one — of the form **using namespace ns** — that makes all names in a nominated namespace *ns* usable in the current scope without namespace qualifiers; see Section 3.1. “**inline namespace**” on page 1055.
- UT** – short for underlying type.
- UTF-8** – a variable-width encoding for Unicode characters that uses one to four 8-bit code units for each code point and is designed to encode the first 128 Unicode code points using a 1-byte representation that is identical to that used by the ASCII character encoding. [User-Defined Literals \(844\)](#)
- UTF-16** – a variable-width encoding for Unicode characters that uses one or two 16-bit code units for each code point. [User-Defined Literals \(844\)](#)
- UTF-32** – a fixed-width encoding for Unicode characters that uses one 32-bit code unit for each code point. [User-Defined Literals \(844\)](#)
- valid but unspecified** – implies, for a given object, that it meets the C++ Standard Library’s minimum requirements for a **moved-from** object; such an object must, in principle, meet all of the requirements of the specific Standard Library template with which it is used, especially that it can be assigned-to (if the template requires assignability), swapped (if the template requires swappability), compared (if the template requires comparison), and destroyed (often required, even if not documented); see also **moved-from state**. [Rvalue References \(715\)](#)
- value** – (1) the platonic value (or else the in-process value) represented by an object such as might affect the result of its associated homogeneous equality-comparison operator or (2) the bit pattern associated with its value representation. [Delegating Ctors \(51\)](#), [noexcept Operator \(625\)](#), [Rvalue References \(741\)](#), [Lambda Captures \(992\)](#), [Ref-Qualifiers \(1162\)](#)
- value category** – a characterization of a compile-time property of a (typically runtime-evaluable) expression. Every expression has one of three disjoint leaf value categories: *lvalue*, *xvalue*,